Adaptive Data Collection for Policy Evaluation, Multi-task Learning and LLM Alignment

by

Subhojyoti Mukherjee

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2025

Date of final oral examination: 02/14/2025

The dissertation is approved by the following members of the Final Oral Committee:

Robert Nowak, Professor, Electrical and Computer Engineering
Josiah Hanna, Assistant Professor, Computer Science
Qiaomin Xie, Assistant Professor, Industrial and Systems Engineering
Kangwook Lee, Assistant Professor, Electrical and Computer Engineering
Branislav Kveton, Principal Scientist, Adobe Research

To all my friends, family, collaborators, mentors, and teachers who helped me reach here.

ACKNOWLEDGMENTS

I am grateful to my thesis advisor Dr. Robert Nowak from the Electrical and Computer Engineering Department at the University of Wisconsin-Madison. His constant guidance, motivation, and perseverance have made my PhD journey enjoyable. One of his greatest influences on me is to teach me how to do research, ask questions, and lead a project from start to end. This became a huge contributing factor to my research for this thesis. I want to thank my co-advisor Josiah Hanna for being a great mentor and working with me on several of my projects. I had a great time working with him on Reinforcement Learning (RL) and learning and ideating new directions in RL. I want to thank my co-advisor, Qiamoin Xie, for grounding me in the theoretical side of my research. I am especially thankful to her for carefully checking all my proofs before submission. Finally, I want to thank my mentor Branislav Kveton from Amazon AWS (and now in Adobe Research). I fondly remember the time we spent at Amazon AWS ideating over new ideas and projects. Most importantly he is always available for feedback, and it is an enjoying experience to conduct research with him.

I am thankful towards a host of my colleagues in the Nowak Lab at Wisconsin Institute of Discovery where I spent a significant time while conducting research at UW-Madison. Thanks, Ardhendu Tripathy, Blake Mason, Julian-Katz Samuels, Greg Canal, Yinglun Zhu, Rahul Parhi, Joseph Shenouda, Liu Yang, Haoyue Bai, Scott Sievert, and Gokcan Tatli for creating such an environment in the lab, conducive to research. I also want to thank my mentors Anusha Lalitha, Aniket Deshmukh, Yifei Ma, Ge Liu, in Amazon AWS who helped me in conducting research during my internship. Because of their guidance and mentorship, I was able to convert the internship to full-time offer. I am thankful to my parents, brother, and sister-in-law for being supportive in my struggles. Last but not least,

I must thank A.P. Vijay Rengarajan who is now working in the industry. He is always there, listening to all my concerns on a host of issues that I faced in my research and personal life. Vijay without you life will not be the same.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

We study the problem of adaptive data collection in Reinforcement Learning (RL). The most challenging aspect of RL is to balance *exploration-exploitation* - the trade-off between finding the most rewarding unexplored action as opposed to sticking with the current best action. In this thesis, we study how to adaptively collect diverse and informative data that balance exploration-exploitation so as to reduce some metric of error. We study how to adaptively collect data to reduce the evaluation error of a policy before its deployment, and how to reduce the prediction error of identifying the best action(s) across all tasks in a multi-task learning setting. In the process, we show how to use data collection by a demonstrator to train a Decision Transformer to learn the optimal algorithm in a multi-task setting. Finally, we extend this idea of adaptive data collection for preference elicitation in Large Language models (LLMs) to align LLMs with human feedback and design prompts for few-shot learning in LLMs.

Part I

Introduction

1 INTRODUCTION AND OVERVIEW

Reinforcement Learning (RL) (Sutton, 1988; Puterman, 2014) has shown great promise in building pro-active agents (Xu et al., 2024a), aligning Large Language Model (LLM) with human feedback (Brown et al., 2020; Rafailov et al., 2023; Liu et al., 2024), autonomous driving (Kiran et al., 2021), robotics (Ibarz et al., 2021; Agarwal et al., 2022), recommender systems (Bottou et al., 2013), and a host of other areas (Fischer, 2018; Yu et al., 2019; Hambly et al., 2021). With the increase in computational power and data availability, researchers have increasingly turned to RL to create agents that can navigate sophisticated challenges through environmental interaction and maximize cumulative rewards in specific tasks.

The RL framework is typically modeled as a Markov Decision Process (MDP) (Puterman, 2014). The MDP is also referred to as an environment and consists of states, actions, transition dynamics regulating movement between states, and reward functions. At every round of interaction, the agent takes action, moves to the next state following the transition dynamics, and observes the reward. The final goal of the agent is to maximize the cumulative reward at the end of some number of interactions. Therefore, the agent needs to plan the sequence of actions to take and states to visit to maximize the cumulative reward. This is called the policy of the agent. We call the *value* of such a policy as the expected cumulative reward that the policy can achieve by the end of all interactions. We also define the *regret* of a policy as the difference between the total expected reward an optimal policy with knowledge of the underlying environment can achieve against the deployed policy. This MDP can model many real-world applications such as recommending items to users (Li et al., 2016) or evaluating such a recommendation system before its deployment (Mukherjee et al., 2022a), training agents for multi-task learning (Du et al., 2023), or aligning LLM with human feedback (Rafailov et al., 2023). In

this thesis, we will be studying several such applications.

The traditional RL approach requires large sample sizes of interactions and long training periods and may not always generalize well to new environments (Zanette and Brunskill, 2019; Agarwal et al., 2019; Crawshaw, 2020). Moreover, it is not always clear apriori what the best actions to take in a task that will enable the agent to maximize the cumulative reward. For example, the agent without any prior knowledge of the task or environment may need to conduct exploration to learn the underlying reward function and transition dynamics. However, conducting too much exploration may lead the agent to select actions that may lead to a decrease in overall cumulative reward. On the other hand, the agent may choose to be myopic and not explore the environment and may get stuck with sub-optimal actions that do not yield the highest cumulative rewards in the long run. This is called the *exploration-exploitation dilemma* and is at the heart of all RL learning problems (Sutton, 1988; Agarwal et al., 2019; Zanette, 2021).

In this thesis, we will study this central question of how to adaptively collect these sequences of interactions to balance the exploration-exploitation of the agent. These sequences of interactions are also referred to as data. While many studies have been conducted on how to balance the exploration-exploitation in RL (Bubeck et al., 2012; Lattimore and Szepesvári, 2020a; Sutton and Barto, 2018; Agarwal et al., 2019; Wang et al., 2022) only a few works have looked into this from the lens of *optimal design* (Jamieson and Jain, 2022; Katz-Samuels et al., 2020; Mukherjee et al., 2023c). Optimal design studies how to collect diverse and informative data (exploration-exploitation) to reduce some *metric of error* within a pre-specified budget (Pukelsheim, 2006; Fedorov, 2010). Many optimal designs such as A, D, T, E, G, V-designs exist in the literature that minimize different metrics of error like the Mean squared error of an estimator, or reducing the variance of an estimator, etc. This metric of error can also

be the estimation error of the value of a policy that needs to be deployed, the error in identifying the optimal arm(s) by a policy that is deployed across multiple tasks, or the estimation error of a reward function before it is used to train an LLM. Consequently, we will study how to collect *diverse and informative* data within a limited budget so that the agent can efficiently learn how good a policy is, how to better generalize across different tasks, or how to align an LLM with human feedback. Therefore, we pose the central question addressed in this thesis as follows:

> *How to adaptively collect diverse and informative data to balance exploration-exploitation and minimize the metric of error?*

Exploration in this context means discovering more diverse data and exploitation means finding examples that minimize the metric of error. For easier exposition to the readers, we divide the thesis into three parts, where each part has one core theme but all of them study the same central question raised above. In the first part, the core theme is data collection for policy evaluation and we study how to use optimal design for data collection for policy to reduce value estimation error of a policy in Markov decision processes (MDPs). In policy evaluation, we are given a *target* policy and asked to estimate the expected cumulative reward it will obtain in an environment formalized as an MDP. We develop theory for optimal design for data collection within the class of tree-structured MDPs and linear bandits where the variances of the rewards depend on the action features. We further extend this line of work to incorporate safety constraints while collecting data for policy evaluation.

In the second part, the core theme is multi-task learning (MTL) and we study optimal design and adaptive data collection for MTL. In MTL setting the goal is to leverage the shared structure across the tasks to perform well in each of the tasks. In this thesis, we focus on the multi-task representation learning (MTRL) setup where the tasks share a common

low-dimensional linear or bilinear representation. We study both linear and bilinear MTRL settings where the goal is to find the best arm(s) within a fixed budget. We develop a double optimal design technique that first selects informative samples to learn the common representation across tasks and then uses informative samples within each task to find the best arm(s) of each task. We show theoretically that this approach leads to a smaller probability of error than existing approaches in the case of fixed budget setting. We further develop the MTL setting to incorporate a new paradigm of data collection to learn an optimal algorithm directly from the collected data without explicitly designing the learning algorithm. This paradigm of learning is called "Learning to learn" (Lee et al., 2023; Friedman et al., 2024). In this setting, we deploy a decision transformer to learn the shared representation across the tasks from data selected by a sub-optimal algorithm. Then we show that the decision transformer can actually leverage this shared representation to learn an optimal algorithm and thereby outperform the demonstrator in various tasks.

In the final part of the thesis, the core theme is LLM alignment and we use the optimal design to collect data for alignment and prompt selection for LLMs. We use the D-optimal design to select examples to learn the human preference model when the underlying ranking feedback follows a Plackett-Luce model. We show empirically that selecting samples using our method leads to minimizing the ranking error in existing LLM datasets like Nectar and Anthropic harmless-helpful dataset. We then use G-optimal design to adaptively design prompts for few-shot learning with LLMs. We experiment with many different tasks in small, medium-sized, and large language models; and show that our proposed algorithms outperform other methods for choosing few-shot examples in the LLM prompt at inference time. In the following section, we briefly discuss each of the chapters covered in the three parts mentioned above.

1.1 Part 2: Adaptive Data Collection for Policy Evaluation in MDPs and Heteroscedastic Linear Bandits

This part of the thesis studies the use of adaptive data collection for policy evaluation in Markov decision processes (MDPs). In policy evaluation, we are given a *target* policy and asked to estimate the expected cumulative reward it will obtain in an environment formalized as an MDP. Previous works in this setting (Carpentier and Munos, 2011, 2012; Carpentier et al., 2015) mainly focused on K-armed stochastic bandits. We extend this line of work in three directions: **1)** Tabular MDPs, **2)** Heteroscedastic linear bandits, and **3)** Tabular MDPs under safety constraints.

Note that we assume that the learning algorithm (agent) does not have access to the underlying problem parameters, including the mean and the variance of the rewards. We call such an algorithm *agnostic algorithm*. In contrast, we call an algorithm that knows the reward variances (but not the reward means) as *oracle algorithm*. We define the loss of an algorithm as the estimation error of the value of the policy. Finally, we define regret as the difference between the loss of the agnostic algorithm against the oracle algorithm.

In this setting, one simple solution is to run the target policy in the environment and estimate its value. This is called the *on-policy method*. However, this is not the optimal methodology because of the noisy rewards. The previous works (Carpentier and Munos, 2011, 2012; Carpentier et al., 2015) have shown that running such on-policy method will result in a regret of $\widetilde{O}(n^{-1})$ where n is the total budget of actions that can be tried and $\widetilde{O}$ hides logarithmic factors.

In fact, running a different policy first to measure the uncertain actions gathering the data, and then evaluating the target policy on them yields a better result. This policy of gathering data is called behavior policy

and this type of learning is called off-policy learning. However, if these behavior policies are generated randomly to gather the dataset to evaluate the target policy they may lead to a lot of excessive sample collection and uninformative samples in the dataset. Instead, we can adaptively select these behavior policies using *Active Learning* (AL) (Settles, 2009; Balcan et al., 2009) to gather the dataset to maximize the accuracy of the value estimate of the target policy using a small number of samples. We briefly discuss the *key contributions* of this part in the following sections.

Chapter 2: Adaptive Data Collection for Policy Evaluation in Tabular MDP (ReVar)

In this chapter, we develop the theory for optimal data collection for policy evaluation within the class of tree-structured MDPs. Note that we are given a target policy and we want to correctly estimate its value (the expected cumulative reward) in an environment where the rewards for each state-action pair can be noisy. Let the learner has n samples for evaluation. We also know that in this setting running an on-policy algorithm will result in a regret of $\widetilde{O}(n^{-1})$. Therefore the key question we ask in this setting is that

> *Can we design an adaptive algorithm for tabular MDP to collect data for policy evaluation that adapts to the variance of each action, and its regret decreases at a rate faster than* $\widetilde{O}(n^{-1})$*?*

We start by first deriving an oracle data collection strategy that uses knowledge of the variance of the reward distributions. We then introduce the **Re**duced **Var**iance Sampling (ReVar) algorithm that approximates the oracle strategy when the reward variances are unknown a priori and bound its sub-optimality compared to the oracle strategy. We show that the regret in tree-structured MDP decreases at a rate of $\widetilde{O}(n^{-3/2})$ which is

achieved through carefully balancing informativeness and diversity of the collected samples by selecting examples with high variance in estimation. Finally, we empirically validate that ReVar leads to policy evaluation with mean squared error comparable to the oracle strategy and significantly lower than simply running the target policy.

Chapter 3: Optimal Design for Data Collection for Policy Evaluation in Linear Heteroscedastic Bandits (SPEED)

In this chapter, we study the problem of optimal data collection for policy evaluation in linear heteroscedastic bandits. Recall that in policy evaluation, we are given a *target* policy and asked to estimate the expected reward it will obtain when executed in a multi-armed bandit environment. Our work is the first work that focuses on such an optimal data collection strategy for policy evaluation involving heteroscedastic reward noise in the linear bandit setting. Let the actions be represented by d-dimensional embeddings, and the learner has n samples for evaluation. The key question we ask in this setting is that

> *Can we design an adaptive algorithm for heteroscedastic linear bandits to collect data for policy evaluation that adapts to the variance of each action, and its regret decreases at a rate faster than* $\widetilde{O}(d^2 n^{-1})$*?*

We first formulate an optimal design for weighted least squares estimates in the heteroscedastic linear bandit setting with the knowledge of noise variances. This design minimizes the mean squared error (MSE) of the estimated value of the target policy and is termed the oracle design. Since the noise variance is typically unknown, we then introduce a novel algorithm, SPEED (**S**tructured **P**olicy **E**valuation **E**xperimental **D**esign), that tracks the oracle design and derive its regret with respect to the oracle design. We show that regret scales as $\widetilde{O}(d^3 n^{-3/2})$ and prove a lower bound

of $\Omega(d^2 n^{-3/2})$. Finally, we evaluate SPEED on a set of policy evaluation tasks and demonstrate that it achieves MSE comparable to an optimal oracle and much lower than simply running the target policy.

Chapter 4: Adaptive Data Collection for Policy Evaluation Under Safety Constraints in Tabular MDP (SaVeR)

In this chapter, we study *safe data collection* for the purpose of policy evaluation in tabular Markov decision processes (MDPs). Again recall that in policy evaluation, we are given a *target* policy and asked to estimate the expected cumulative reward it will obtain. Policy evaluation requires data and we are interested in the question of what *behavior* policy should collect the data for the most accurate evaluation of the target policy. While prior work has considered behavior policy selection, in this paper, we additionally consider a safety constraint on the behavior policy. Namely, we assume there exists a known default policy that incurs a particular expected cost when run and we enforce that the cumulative cost of all behavior policies ran is better than a constant factor of the cost that would be incurred had we always run the default policy. Assume that we have n samples. Then we ask two key questions in this setting:

> 1) *Is there a class of MDPs where it is possible to incur a regret that degrades at a faster rate than* $\widetilde{O}(n^{-1})$*?* while satisfying safety constraints?
>
> 2) *If the answer is yes to (1), can we design an adaptive algorithm (for this class of MDPs) to collect data for policy evaluation that does not violate the safety constraints (in expectation), and its regret degrades at a faster rate than* $\widetilde{O}(n^{-1})$*?*

We first show that there exists a class of intractable MDPs where no

safe oracle algorithm with knowledge about problem parameters can efficiently collect data and satisfy the safety constraints. We then define the tractability condition for an MDP such that a safe oracle algorithm can efficiently collect data and using that we prove the first lower bound for this setting. We then introduce an algorithm SaVeR for this problem that approximates the safe oracle algorithm and bound the finite-sample mean squared error of the algorithm while ensuring it satisfies the safety constraint. Finally, we show in simulations that SaVeR produces low MSE policy evaluation while satisfying the safety constraint.

1.2 Part 3: Adaptive Data Collection in Multi-task Learning

In this part, we study Multi-task learning (MTL) for linear, bilinear, and other structured bandit settings. As discussed before, the traditional RL approach may not always generalize well to new environments (Zanette and Brunskill, 2019; Agarwal et al., 2019; Crawshaw, 2020). To address these challenges, researchers have focused on multi-task learning which allows knowledge to be shared across different tasks, leading to improved learning efficiency, enhanced performance, and better generalization capabilities (Bengio et al., 1990; Schaul and Schmidhuber, 2010; Tripuraneni et al., 2021; Du et al., 2023; Mukherjee et al., 2023b). In MTL multiple tasks are simultaneously learned by a shared model. This type of approach offers advantages such as improved data efficiency, reduced overfitting through shared representations, and fast learning by leveraging side information that is shared across the tasks (Crawshaw, 2020). In the first chapter, we exclusively focus on Multi-task representation learning (MTRL) in pure exploration setting (Audibert et al., 2009, 2010). Previous works in the MTRL setting have exclusively focused on regret minimization whereas we focus on identifying the best set of the arm(s) for each of the tasks

within a fixed budget (fixed budget setting).

In the next chapter, we introduce a paradigm of learning for the MTL setting called "Learning to learn". In all of the previous MTL works (Tripuraneni et al., 2020, 2021; Yang and Tan, 2021; Mukherjee et al., 2024f) the key approach is to first formulate the underlying environment (say linear, bilinear, or MDP), then derive the optimal or near-optimal algorithm with the knowledge of the underlying structure of the problem. Note that the learning agent knows the structure shared across the tasks but does not know the reward means. In this chapter, we study how we can learn the underlying shared structure from demonstrations, and in the process learn an optimal algorithm as well. We briefly discuss the *key contributions* of this part in the following sections.

Chapter 5: Multi-task Representation Learning for Bilinear Bandits for Fixed Confidence Setting (GOBLIN)

In this chapter, we study multi-task representation learning for the problem of pure exploration in bilinear bandits. In bilinear bandits, an action takes the form of a pair of arms from two different entity types and the reward is a bilinear function of the known feature vectors of the arms. In the *multi-task bilinear bandit problem*, we aim to find optimal actions for multiple tasks that share a common low-dimensional linear representation. The objective is to leverage this characteristic to expedite the process of identifying the best pair of arms for all tasks. gFrom the works of Jun et al. (2019); Lu et al. (2021); Kang et al. (2022) we know that the effective dimension is actually $(d_1 + d_2)r$, where d_1, d_2 are the left and right ambient dimensions and r is the rank. Similarly, for the multi-task representation learning given M tasks, the effective dimension should scale with the learned latent features $(k_1 + k_2)r$ where k_1, and k_2 are the left and right latent dimensions. Let Δ be the minimum reward gap. Hence we ask the following question:

Can we design an algorithm for multi-task pure exploration bilinear bandit problem that can learn the latent features and has sample complexity that scales as $\widetilde{O}(M(k_1 + k_2)r/\Delta^2)$*?*

We propose the algorithm GOBLIN that uses an experimental design approach to optimize sample allocations for learning the global representation as well as minimize the number of samples needed to identify the optimal pair of arms in individual tasks. To the best of our knowledge, this is the first study to give sample complexity analysis for pure exploration in bilinear bandits with shared representation. Our results demonstrate that by learning the shared representation across tasks, we achieve significantly improved sample complexity compared to the traditional approach of solving tasks independently.

Chapter 6: Pretraining Decision Transformers with Reward Prediction for In-Context Structured Bandit Learning (PreDeToR)

In this chapter, we study the multi-task structured bandit problem where the goal is to learn a near-optimal algorithm that minimizes cumulative regret. The tasks share a common structure and the algorithm exploits the shared structure to minimize the cumulative regret for an unseen but related test task. We use a transformer as a decision-making algorithm to learn this shared structure so as to generalize to the test task. The prior work of pretrained decision transformers like DPT (Lee et al., 2023) requires access to the optimal action during training which may be hard in several scenarios. With this past work in mind, the goal of this chapter is to answer the question:

Can we learn an in-context bandit learning algorithm in an MTL setting that obtains lower regret than the algorithm used to produce the training data without knowledge of the optimal action in each training task?

We show that our learning algorithm does not need the knowledge of optimal action per task during training but predicts a reward vector for each of the actions using only the observed offline data from the diverse training tasks. Finally, during inference time, it selects action using the reward predictions employing various exploration strategies in-context for an unseen test task. We call this new pre-training methodology as **Pre**-trained **De**cision **T**rans**f**o**r**mer with **R**eward Estimation (PreDeToR).

We show that our model outperforms other SOTA methods like DPT, and Algorithmic Distillation (AD) over a series of experiments on several structured bandit problems (linear, bilinear, latent, non-linear). Interestingly, we show that our algorithm, without the knowledge of the underlying problem structure, can learn a near-optimal policy in-context by leveraging the shared structure across diverse tasks. We further extend the field of pre-trained decision transformers by showing that they can leverage unseen tasks with new actions and still learn the underlying latent structure to derive a near-optimal policy. We validate this over several experiments to show that our proposed solution is very general and has wide applications to potentially emergent online and offline strategies at test time. Finally, we theoretically analyze the performance of our algorithm and obtain generalization bounds in the in-context multi-task learning setting.

1.3 Adaptive Data Collection for Preference Elicitation, Prompt Designing, and Alignment in LLMs

The emergence of Large Language Models (LLMs) with their remarkable capabilities has sparked a new direction in AI agent development (Touvron et al., 2023; Vaswani et al., 2017). Researchers are increasingly using LLMs as the cognitive core of AI agents, expanding their abilities through multimodal perception and tool use (Xi et al., 2023). These LLM-based agents can use reasoning and planning through techniques like Chain-of-Thought (CoT) (Wei et al., 2022) and problem decomposition, while also developing interactive environmental capabilities similar to reactive agents through feedback-based learning and action generation. A key advantage of LLM-based agents stems from their pre-training on vast text corpora, which enables few-shot and zero-shot generalization. This allows them to transfer knowledge between tasks without parameter updates, making them highly adaptable.

As an example of a downstream task that requires fine-tuning, we state the following example. Consider the following proactive agent which takes input the command “Plan a vacation to Italy within my budget". This agent then searches the knowledge base and Internet about important tourist places to visit in Italy, ranks these places based on the budget (reasoning), and then confirms these with the user. It can then go ahead and book flights/cars/hotels at these places, authenticate all of these with the user, and provide the necessary receipts. An illustrative picture is shown in Figure 1.1.

The Large Language Models (LLMs) like BERT, GPT3, dominate the leaderboards for many NLP tasks (Devlin et al., 2018; Yang et al., 2019; Radford et al., 2019; Brown et al., 2020; Suzgun et al., 2022; Srivastava et al., 2022). However, fine-tuning or aligning an LLM on a downstream task

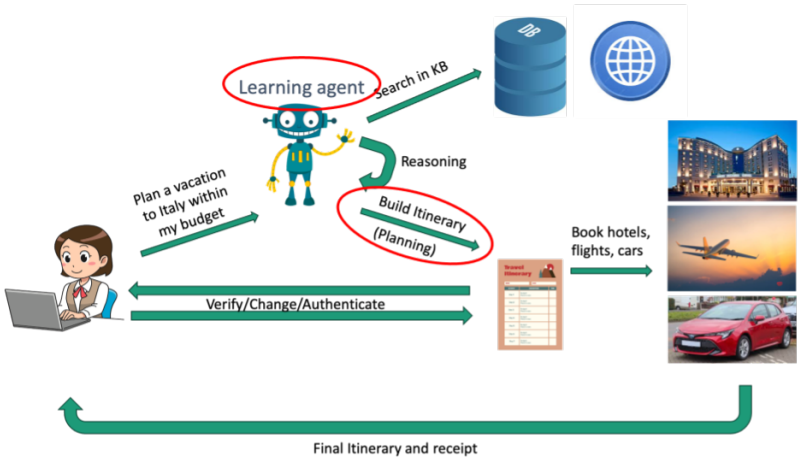


Figure 1.1: A Proactive Agent

requires a lot of informative and diverse labeled data. If these models are not fine-tuned on a large number of examples their performance varies drastically (Dodge et al., 2020). It is also expensive to gather these labeled examples and train these LLMs (Strubell et al., 2020; Dong et al., 2022). Hence, there is a scope for using optimal design to collect informative data to drastically reduce the number of labeled examples needed in the finetuning process or preference learning (Rafailov et al., 2024; Mukherjee et al., 2024d) of these LLMs. Therefore, in this part of the thesis we now focus on how to adaptively select informative examples for preference alignment of LLMs, and adaptively designing prompts for zero-shot learning.

Chapter 7: Optimal Design for Human Preference Elicitation

Learning of preference models from human feedback has been central to recent advances in artificial intelligence. Motivated by the cost of obtaining high-quality human annotations, we study efficient human preference elicitation for learning preference models. So the goal of this chapter is to

answer the question:

> *Given a budget of n samples, can we select informative and diverse examples to learn the preference model?*

We answer affirmatively to this question. The key idea in our work is to generalize optimal designs, a methodology for computing optimal information-gathering policies, to questions with multiple answers, represented as lists of items. The policy is a distribution over lists and we elicit preferences from the list proportionally to its probability. To show the generality of our ideas, we study both absolute and ranking feedback models on items in the list. We design efficient algorithms for both and analyze them. Finally, we demonstrate that our algorithms are practical by evaluating them on existing question-answering problems.

Chapter 8: Optimal Design for Adaptive In-Context Prompt Tuning in Large Language Models

One emergent ability of large language models (LLMs) is that query-specific examples can be included in the prompt at inference time. In this work, we use active learning for adaptive prompt design and call it **A**ctive **I**n-context **P**rompt **D**esign (AIPD). In this chapter, we ask the following question:

> *Given a budget of n samples, can we adaptively design prompts that balance diversity and informativeness for few-shot learning in LLMs?*

We show that this is indeed possible. We design the LLM prompt by adaptively choosing few-shot examples from a training set to optimize performance on a test set. The training examples are initially unlabeled and we obtain the label of the most informative ones, which maximally

reduces uncertainty in the LLM prediction. We propose two algorithms, GO and SAL, which differs in how the few-shot examples are chosen. We analyze these algorithms in linear models: first GO and then use its equivalence with SAL. We experiment with many different tasks in small, medium-sized, and large language models; and show that GO and SAL outperform other methods for choosing few-shot examples in the LLM prompt at inference time.

We briefly summarize the use of adaptive data collection in various chapters in my thesis in Table 1.1.

Chapters	**Adaptive Data Collection**	**Conference**
Chapter 2, Revar: Strengthening policy evaluation via reduced variance sampling (Mukherjee et al., 2022a)	PE-optimal design for Policy evaluation in Tabular MDPs	UAI 2022
Chapter 3, Speed: Experimental design for policy evaluation in linear heteroscedastic bandits (Mukherjee et al., 2024g)	PE-optimal design for Policy evaluation in linear bandits	AISTATS 2024
Chapter 4, SaVeR: Optimal Data Collection Strategy for Safe Policy Evaluation in Tabular MDP (Mukherjee et al., 2024a)	PE-optimal design Under safety constraints	ICML 2024
Chapter 5, Multi-task Representation Learning for Pure Exploration in Bilinear Bandits(Mukherjee et al., 2024f)	E and G-optimal design for representation learning for fixed confidence setting	NeurIPS 2023
Chapter 6, Pretraining Decision Transformers with Reward Prediction for In-Context Multi-task Structured Bandit Learning (Mukherjee et al., 2024b)	Adaptive data collection using Decision Transformer to learn the optimal algorithm	(In submission)
Chapter 7, Optimal Design for Human Preference Elicitation (Mukherjee et al., 2024c)	G-optimal Design for Plackett-Luce Model	NeurIPS 2024
Chapter 8, Optimal Design for Adaptive In-Context Prompt Design in Large Language Models (Mukherjee et al., 2024e)	G-optimal Design for Prompt Design	Technical Report

Table 1.1: Use of Adaptive Data Collection in my Thesis

Part II

Adaptive Data Collection for Policy Evaluation

2 REVAR: STRENGTHENING POLICY EVALUATION VIA REDUCED VARIANCE SAMPLING

In reinforcement learning (RL) applications, there is often a need for policy evaluation to determine (or estimate) the expected return (future cumulative reward) of a given policy. Policy evaluation is also required in other sequential decision-making settings outside of RL. For example, testing an autonomous vehicle stack or ad-serving system can be seen as policy evaluation applications. Accurate and data efficient policy evaluation is critical for safe and trust-worthy deployment of autonomous systems.

This chapter studies data collection for low mean squared error (MSE) policy evaluation in sequential decision-making tasks formalized as Markov decision processes (MDPs). The objective of policy evaluation is to estimate the expected return that will be obtained by running a *target policy* which is a given probabilistic mapping from states to actions.

To evaluate the target policy, we require data from the environment in which it will be deployed. Collecting data requires running a (possibly non-stationary) *behavior* policy to generate state-action-reward trajectories. Our goal is to find a behavior policy that leads to a minimum MSE evaluation of the target policy.

The most natural choice is *on-policy sampling* in which we use the target policy as the behavior policy. However, we show that in some cases this choice is far from optimal (e.g., Figure 2.2 in our empirical analysis) as it fails to actively take actions from which the expected return is uncertain. Instead, an optimal behavior policy should take actions in any given state to reduce uncertainty in the current estimate of the expected return from that state.

This chapter makes the following main contributions. We first derive an optimal "oracle" behavior policy for finite tree-structured MDPs *assuming oracle access to the MDP transition probabilities and variances of the reward*

distributions. Sampling trajectories according to the oracle behavior policy minimizes the MSE of the estimator of the target policy's expected. As a special case (depth 1 tree MDPs), we recover the optimal behavior policy for multi-armed bandits Carpentier et al. (2015).

We then introduce a practical algorithm, **Re**duced **Var**iance Sampling (ReVar), that adaptively learns the optimal behavior policy by observing rewards and adjusting the policy to select actions that reduce the MSE of the estimator. The main idea of ReVar is to plug-in upper-confidence bounds on the reward distribution variances to approximate the oracle behavior policy. We define a notion of policy evaluation regret compared to the oracle behavior policy, and bound the regret of ReVar. The regret converges rapidly to 0 as the number of sampled episodes grows, theoretically guaranteeing that ReVar quickly matches the performance of the oracle policy. Finally, we implement ReVar and show it leads to low MSE policy evaluation in both a tree-structured and a general finite-horizon MDP. Taken together, our contributions provide a theoretical foundation towards optimal data collection for policy evaluation in MDPs.

The remainder of the chapter is organized as follows. In Section 2.1 we describe the preliminaries of our setting, and in Section 2.2 we describe the related works. In Section 2.3 we reformulate our problem in the bandit setting and discuss related bandit works. In Section 2.4 we extend the bandit formulation to the tree MDP. Finally we introduce the more general Directed Acyclic Graph (DAG) MDP in Section 2.5 and discuss some limitations of our sampling behavior. We show numerical experiments in Section 2.6 and conclude in Section 2.7.

2.1 Background

In this section, we introduce notation, define the policy evaluation problem, and discuss the prior literature.

Notation

A finite-horizon Markov Decision Process, $\mathbf{M}$, is the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0, L)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a state transition function, R is the reward distribution (formalized below), $\gamma \in [0, 1)$ is the discount factor, d_0 is the starting state distribution, and L is the maximum episode length. A (stationary) policy, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, is a probability distribution over actions conditioned on a given state. We assume data can only be collected through episodic interaction: an agent begins in state $S_0 \sim d_0$ and then at each step t takes an action $A_t \sim \pi(\cdot|S_t)$ and proceeds to state $S_{t+1} \sim P(\cdot|S_t, A_t)$. Interaction terminates in at most L steps. Each time the agent takes action a_t in state s_t it observes a reward $R_t \sim R(s_t, a_t)$. We assume $R(s, a) = \mathcal{P}(\mu(s, a), \sigma^2(s, a))$, where $\mathcal{P}$ denotes a parametric distribution with mean $\mu(s, a)$ and variance $\sigma^2(s, a)$. The entire interaction produces a trajectory $H \coloneqq \{(S_t, A_t, R_t)\}_{t=1}^{L}$. We assume d_0 is known but P and the reward distributions are unknown. We define the value of a policy as: $v(\pi) \coloneqq \mathbb{E}_\pi[\sum_{t=1}^{L} \gamma^{t-1} R_t]$, where $\mathbb{E}_\pi$ is the expectation w.r.t. trajectories sampled by following π.

We will make use of the fact that the value of a policy can be written as: $v(\pi) = \mathbb{E}[v_0^\pi(S_0)|S_0 \sim d_0]$ where,

$$v_t^\pi(s) \coloneqq \sum_a \pi(a|s)\mu(s, a) + \gamma \sum_{s'} P(s'|s, a) v_{t+1}^\pi(s')$$

for $t \leqslant L$ and $v_t^\pi(s) = 0$ for $t > L$.

Policy Evaluation

We now formally define our objective. We are given a target policy, π, for which we want to estimate $v(\pi)$. To estimate $v(\pi)$ we will generate a set of K trajectories where each trajectory is generated by following some policy. Let $H^k \coloneqq \{s_t^k, a_t^k, R_t^k(s_t^k, a_t^k)\}_{t=1}^{L}$ be the trajectory collected in episode k

and let b^k be the policy ran to produce H^k. The entire set of collected data is given as $\mathcal{D} \coloneqq \{H^k, b^k\}_{k=1}^K$.

Once $\mathcal{D}$ is collected, we estimate $v(\pi)$ with a certainty-equivalence estimate (Sutton, 1988). Suppose $\mathcal{D}$ consists of $n = KL$ state-action transitions. We define the random variable representing the estimated future reward from state s at time-step t as:

$$Y_n(s,t) \coloneqq \sum_a \pi(a|s)\widehat{\mu}(s,a) + \gamma \sum_{s'} \widehat{P}(s'|s,a)Y_n(s',t+1),$$

where $Y_n(s,t+1) \coloneqq 0$ if $t \geqslant L$, $\widehat{\mu}(s,a)$ is an estimate of $\mu(s,a)$ and $\widehat{P}(s'|s,a)$ is an estimate of $P(s'|s,a)$, both computed from $\mathcal{D}$. Finally, the estimate of $v(\pi)$ is computed as $Y_n \coloneqq \sum_s d_0(s)Y_n(s,0)$. In the policy evaluation literature, the certainty-equivalence estimator is also known as the direct method (Jiang and Li, 2016) and, in tabular settings, can be shown to be equivalent to batch temporal-difference estimators (Sutton, 1988; Pavse et al., 2020). Thus, it is representative of two types of policy evaluation estimators that often give strong empirical performance (Voloshin et al., 2019).

Our objective is to determine the sequence of behavior policies that minimize error in estimation of $v(\pi)$. Formally, we seek to minimize mean squared error which is defined as: $\mathbb{E}_{\mathcal{D}}\left[(Y_n - v(\pi))^2\right]$ where the expectation is over the collected data set $\mathcal{D}$.

2.2 Related Work

This chapter builds upon work in the bandit literature for optimal data collection for estimating a weighted sum of the mean reward associated with each arm. Antos et al. (2008) study estimating the mean reward of each arm equally well and show that the optimal solution is to pull each arm proportional to the variance of its reward distribution. Since the

variances are unknown a priori, they introduce an algorithm that pulls arms in proportion to the empirical variance of each reward distribution. Carpentier et al. (2015) extend this work by introducing a weighting on each arm that is equivalent to the target policy action probabilities in our work. They show that the optimal solution is then to pull each arm proportional to the product of the standard deviation of the reward distribution and the arm weighting. Instead of using the empirical standard deviations, they introduce an upper confidence bound on the standard deviation and use it to select actions. Our work is different from these earlier works in that we consider more general tree-structured MDPs of which bandits are a special case.

In RL and MDPs, exploration is widely studied with the objective of finding the optimal policy. Prior work attempts to balance exploration to reduce uncertainty with exploitation to converge to the optimal policy. Common approaches are based on reducing uncertainty (Osband et al., 2016; O'Donoghue et al., 2018) or incentivizing visitation of novel states (Barto, 2013; Pathak et al., 2017; Burda et al., 2018). These works differ from our work in that we focus on evaluating a fixed policy rather than finding the optimal policy. In our problem, the trade-off becomes balancing taking actions to reduce uncertainty with taking actions that the target policy is likely to take.

Our work is similar in spirit to work on adaptive importance sampling (Rubinstein and Kroese, 2013) which aims to lower the variance of Monte Carlo estimators by adapting the data collection distribution. Adaptive importance sampling was used by Hanna et al. (2017a) to lower the variance of policy evaluation in MDPs. It has also been used to lower the variance of policy gradient RL algorithms (Bouchard et al., 2016; Ciosek and Whiteson, 2017). AIS methods attempt to find a single optimal sampling distribution whereas our approach attempts to reduce uncertainty in the estimated mean rewards. In a similar spirit, Talebi and Maillard (2019) adapt the

behavior policy to minimize error in estimating the transition model P.

2.3 Optimal Data Collection in Multi-armed Bandits

Before we address optimal data collection for policy evaluation in MDPs, we first revisit the problem in the bandit setting as addressed by earlier work (Carpentier et al., 2015). The bandit setting provides intuition for how a good data collection strategy should select actions, though it falls short of an entire solution for MDPs.

Observe that the policy value in a bandit problem is defined as $v(\pi) \coloneqq \sum_{a=1}^{A} \pi(a)\mu(a)$ where the bandit consist of a single state s and A actions indexed as $a = 1, 2, \ldots, A$. In this setting, the horizon $L = 1$ so we return to the same state after taking an action a at time t. Hence, we drop the state s from our standard notation.

Suppose we have a budget of n samples to divide between the arms and let $T_n(1), T_n(2), \ldots, T_n(A)$ be the number of samples allocated to actions $1, 2, \ldots, A$ at the end of n rounds. We define the estimate:

$$Y_n \coloneqq \sum_{a=1}^{A} \frac{\pi(a)}{T_n(a)} \sum_{h=1}^{T_n(a)} R_h(a) = \sum_{a=1}^{A} \pi(a)\widehat{\mu}(a). \tag{2.1}$$

where, $R_h(a)$ is the h^{th} reward received after taking action a. Note that, once all actions where $\pi(a) > 0$ have been tried, Y_n is an unbiased estimator of $v(\pi)$ since $\widehat{\mu}(a)$ is an unbiased estimator of $\mu(a)$. Thus, reducing MSE requires allocating the n samples to reduce variance. As shown by Carpentier et al. (2015), the minimal-variance allocation is given by pulling each arm with the proportion $b^\star(a) \propto \pi(a)\sigma(a)$. Though this result was previously shown, we prove it for completeness in Proposition 1 in Section A.1. Intuitively, there is more uncertainty about the mean reward for

actions with higher variance reward distributions. Selecting these actions more often is needed to offset higher variance. The optimal proportion also takes π into account as a high variance mean reward estimate for one action can be acceptable if π would rarely take that action.

Note that sampling according to eq. (A.1) introduces unnecessary variance compared to deterministically selecting actions to match the optimal proportion. Since the variances are typically unknown, a number of works in the bandit community propose different approaches to estimate the variances for both basic bandits and several related extensions (Antos et al., 2008; Carpentier and Munos, 2011, 2012; Carpentier et al., 2015; Neufeld et al., 2014). Finally, note that incorporating variance aware techniques has been studied in multi-armed bandits (Audibert et al., 2009; Mukherjee et al., 2018). However, these works tend to focus on regret minimization, whereas we focus on MSE reduction. However, none of these works address the fundamental challenge that MDPs bring – action selection must account for both immediate variance reduction in the current state as well as variance reduction in future states visited. In the next section, we begin to address this challenge by deriving minimal-variance action proportions for tree-structured MDPs.

2.4 Optimal Data Collection in Tree MDPs

In this section, we derive the optimal action proportions for tree-structured MDPs assuming the variances of the reward distributions are known, introduce an algorithm that approximates the optimal allocation when the variances are unknown, and bound the finite-sample MSE of this algorithm. Tree MDPs are a straightforward extension of the multi-armed bandit model to capture the fact that the optimal allocation for each action in a given state must consider the future states that could arise from taking that action.

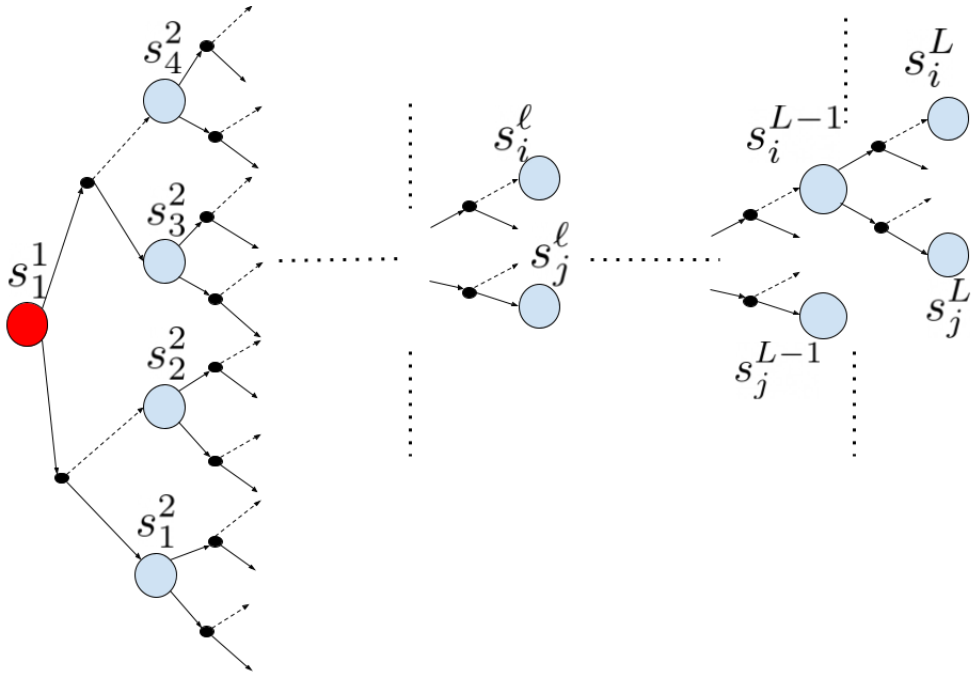


Figure 2.1: An L-depth tree with 2 actions at each state.

We first define a discrete tree MDP as follows:

Definition 2.1. (***Tree MDP***) *An MDP is a discrete tree MDP* $\mathbf{T} \subset \mathbf{M}$ (*see Figure 2.1*) *if the following holds:*

(***1***) *There are* L *levels indexed by* ℓ *where* $\ell = 1, 2, \ldots, L$.

(***2***) *Every state is represented as* s_i^ℓ *where* ℓ *is the level of the state* s *indexed by* i.

(***3***) *The transition probabilities are such that one can only transition from a state in level* ℓ *to one in level* $\ell + 1$ *and each non-initial state can only be reached through one other state and only one action in that state. Formally,* $\forall s'$, $P(s'|s, a) \neq 0$ *for only one state-action pair* s, a *and if* s' *is in level* $\ell + 1$ *then* s *is in level* ℓ. *Finally,* $P(s_j^{L+1}|s_i^L, a) = 0, \forall a$.

(***4***) *For simplicity, we assume that there is a single starting state* s_1^1 (*called the root*). *It is easy to extend our results to multiple starting states with a starting state distribution,* d_0, *by assuming that there is only one action available in the*

root that leads to each possible start state, s*, with probability* $d_0(s)$*. The leaf states are denoted as* s_i^L*.*

(5) *The interaction stops after* L *steps in state* s_i^L *after taking an action* a *and observing the reward* $R_L(s_i^L, a)$*.*

Note that, because we assume a single initial state, s_1^1, we have that estimating $v(\pi)$ is equivalent to estimating $v(s_1^1)$. A similar Tree MDP model has been previously used in theoretical analysis by Jiang and Li (2016); our model is slightly more general as we consider per-step stochastic rewards whereas Jiang and Li (2016) only consider deterministic rewards at the end of trajectories.

Oracle Data Collection

We first consider an oracle data collection strategy which knows the variance of all reward distributions and knows the state transition probabilities. After observing n state-action-reward tuples, the oracle computes the following estimate of $v^\pi(s_1^1)$ (or equivalently $v(\pi)$):

$$\begin{aligned} Y_n(s_1^1) &\coloneqq \sum_{a=1}^{A} \pi(a|s_1^1)\left(\frac{1}{T_n(s_1^1, a)} \sum_{h=1}^{T_n(s_1^1,a)} R_h(s_1^1, a) + \gamma \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_1^1, a) Y_n(s_j^2)\right) \\ &= \sum_{a=1}^{A} \pi(a|s_1^1)\left(\widehat{\mu}(s_1^1, a) + \gamma \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_1^1, a) Y_n(s_j^2)\right) \end{aligned} \tag{2.2}$$

where $T_n(s, a)$ denotes the number of times that the oracle took action a in state s. Note that in Section 2.1 we define $Y_n(s, t)$ but now we use $Y_n(s)$ as timestep is implicit in the layer of the tree. Also (2.2) differs from the estimator defined in Section 2.1 as it uses the true transition probabilities, P, instead of their empirical estimate, $\widehat{P}$. The MSE of Y_n is:

$$\mathbb{E}_{\mathcal{D}}[\left(Y_n(s_1^1) - v^\pi(s_1^1)\right)^2] = \mathrm{Var}(Y_n(s_1^1)) + \mathrm{bias}^2(Y_n(s_1^1)). \tag{2.3}$$

The bias of this estimator becomes zero once all (s, a)-pairs with $\pi(a|s) > 0$ have been visited a single time, thus we focus on reducing $\text{Var}(Y_n(s_1^1))$. Before defining the oracle data collection strategy, we first state an assumption on $\mathcal{D}$.

Assumption 1. *The data $\mathcal{D}$ collected over n state-action-reward samples has at least one observation of each state-action pair, (s, a), for which $\pi(a|s) > 0$.*

Assumption 1 ensures that Y_n is an unbiased estimator of $v(\pi)$ so that reducing MSE is equivalent to reducing variance. Before stating our main result, we provide intuition with a lemma that gives the optimal proportion for each action in a 2-depth tree.

Lemma 2.2. *Let* **T** *be a 2-depth stochastic tree MDP as defined in Theorem 2.1 (see Figure A.1 in Section A.2). Let $Y_n(s_1^1)$ be the estimated return of the starting state s_1^1 after observing n state-action-reward samples. Note that $v^\pi(s_1^1)$ is the expectation of $Y_n(s_1^1)$ under Assumption 1. Let $\mathcal{D}$ be the observed data over n state-action-reward samples. Minimal MSE, $\mathbb{E}_\mathcal{D}[(Y_n(s_1^1) - v^\pi(s_1^1))^2]$, is obtained by taking actions in each state in the following proportions:*

$$b^*(a|s_j^2) \propto \pi(a|s_j^2)\sigma(s_j^2, a)$$

$$b^*(a|s_1^1) \propto \sqrt{\pi^2(a|s_1^1)\left[\sigma^2(s_1^1, a) + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1, a)B^2(s_j^2)\right]},$$

where, $B(s_j^2) = \sum_a \pi(a|s_j^2)\sigma(s_j^2, a)$.

Proof (Overview): We decompose the MSE into its variance and bias terms and show that Y_n is unbiased under Assumption 1. Next note that the reward in the next state is conditionally independent of the reward in the current state given the current state and action. Hence we can write the variance in terms of the variance of the estimate in the initial state and the variance of the estimate in the final layer. We then rewrite the total

samples of a state-action pair i.e $T_n(s_i^\ell, a)$ in terms of the proportion of the number of times the action was sampled in the state i.e $b(a|s_i^\ell)$. To do so, we take into account the tree structure to derive the expected proportion of times that action a is taken in each state in layer 2 as follows:

$$b(a|s_i^2) = \frac{T_n(s_i^2, a)}{\sum_{a'} T_n(s_i^2, a')} \overset{(a)}{=} \frac{T_n(s_i^2, a)/n}{P(s_i^2|s_1^1, a)T_n(s_1^1, a)/n}$$

where in (a) the action a is used to transition to state s_j^2 from s_1^1 and so $\sum_a T_n(s_i^2, a) = P(s_i^2|s_1^1, a)T_n(s_1^1, a)$. We next substitute the $b(a|s_i^\ell)$ for each state-action pair into the variance expression and determine the b values that minimize the expression subject to $\forall s, \sum_a b(a|s) = 1$ and $\forall s, b(a|s) > 0$. The full proof is given in Section A.2. ■

Note that the optimal proportion in the leaf states, $b^*(a|s_j^2)$, is the same as in Carpentier and Munos (2011) (see Proposition 1) as terminal states can be treated as bandits in which actions do not affect subsequent states. The key difference is in the root state, s_1^1, where the optimal action proportion, $b^*(a|s_1^1)$ depends on the expected leaf state normalization factor $B(s_j^2)$ where s_j^2 is a state sampled from $P(\cdot|s_1^1, a)$. The normalization factor, $B(s_i^2)$, captures the total contribution of state s_i^2 to the variance of Y_n and thus actions in the root state must be chosen to 1) reduce variance in the immediate reward estimate and to 2) get to states that contribute more to the variance of the estimate. We explore the implications of the oracle action proportions in Theorem 2.2 with the following two examples.

Example 2.3. (***Child Variance matters***) *Consider a* 2*-depth,* 2*-action tree MDP* $\mathbf{T}$ *with deterministic* P*, i.e.,* $P(s_2^2|s_1^1, 2) = P(s_1^2|s_1^1, 1) = 1$ *and* $\gamma = 1$ *(see Figure A.2 (Left) in Section A.3). Suppose the target policy is the uniform distribution in all states so that* $\forall(s, a), \pi(a|s) = \frac{1}{2}$*. The reward distribution variances are given by* $\sigma^2(s_1^1, 1) = 400$*,* $\sigma^2(s_1^1, 2) = 600$*,* $\sigma^2(s_1^2, 1) = 400$*,* $\sigma^2(s_1^2, 2) = 400$*,* $\sigma^2(s_2^2, 1) = 4$*, and* $\sigma^2(s_2^2, 2) = 4$*. So the right sub-tree at* s_1^1 *has higher variance (larger* B*-value) than the left sub-tree. Following the sampling*

rule in Theorem 2.2 we can show that $b^*(1|s_1^1) > b^*(2|s_1^1)$ *(the full calculation is given in Section A.3). Hence the right sub-tree with higher variance will have a higher proportion of pulls which allows the oracle to get to the high variance* s_1^2*. Observe that treating* s_1^1 *as a bandit leads to choosing action 2 more often as* $\sigma^2(s_1^1, 2) > \sigma^2(s_1^1, 1)$*. However, taking action 2 leads to state* s_2^2 *which contributes much less to the total variance. Thus, this example highlights the need to consider the variance of subsequent states.*

Example 2.4. **(*Transition Model matters*)** *Consider a* 2*-depth,* 2*-action tree MDP* **T** *in which we have* $P(s_1^2|s_1^1, 1) = p$, $P(s_1^2|s_1^1, 1) = 1 - p$, $P(s_3^2|s_1^1, 2) = p$, *and* $P(s_4^2|s_1^1, 2) = 1 - p$*. This example is shown in Figure A.2 (Right) in Section A.3. Following the result of Theorem 2.2 if* $p \gg (1 - p)$ *it can be shown that the variances of the states* s_1^2 *and* s_3^2 *have greater importance in calculating the optimal sampling proportions of* s_1^1*. The calculation is shown in Section A.4. Thus, less likely future states have less importance for computing the optimal sampling proportion in a given state.*

Having developed intuition for minimal-variance action selection in a 2-depth tree MDP, we now give our main result that extends Theorem 2.2 to an L-depth tree.

Theorem 1. *Assume the underlying MDP is an* L*-depth tree MDP as defined in Theorem 2.1. Let the estimated return of the starting state* s_1^1 *after* n *state-action-reward samples be defined as* $Y_n(s_1^1)$*. Note that the* $v^\pi(s_1^1)$ *is the expectation of* $Y_n(s_1^1)$ *under Assumption 1. Let* $\mathcal{D}$ *be the observed data over* n *state-action-reward samples. To minimize MSE* $\mathbb{E}_{\mathcal{D}}[(Y_n(s_1^1)) - \mu(Y_n(s_1^1)))^2]$ *the optimal sampling proportions for any arbitrary state is given by:*

$$b^*(a|s_i^\ell) \propto \sqrt{\pi^2(a|s_i^\ell)\left[\sigma^2(s_i^\ell, a) + \gamma^2 \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) B^2(s_j^{\ell+1})\right]},$$

where, $B(s_j^\ell)$ *is the normalization factor defined as follows:*

$$B(s_i^\ell) = \sum_a \sqrt{\pi^2(a|s_i^\ell)\left(\sigma^2(s_i^\ell, a) + \gamma^2 \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) B^2(s_j^{\ell+1})\right)} \tag{2.4}$$

Proof (Overview): We prove Theorem 1 by induction. Theorem 2.2 proves the base case of estimating the sampling proportion for level $L-1$ and L. Then, for the induction step, we assume that all the sampling proportions from level L till some arbitrary level $\ell+1$ can be subsequently built up using dynamic programming starting from level L. For states in level L to the states in level $\ell+1$ we can compute $b^*(a|s_i^{\ell+1})$ by repeatedly applying Theorem 2.2. Then we show that at the level ℓ we get a similar recursive sampling proportion as stated in the theorem statement. The proof is given in Section A.5. ■

MSE of the Oracle

In this subsection, we derive the MSE that the oracle will incur when matching the action proportions given by Theorem 1. The oracle is run for K episodes where each episode consist of L length trajectory of visiting state-action pairs. So the total budget is $n = KL$. At the end of the K-th episode the MSE of the oracle is estimated which is shown in Proposition 2. Before stating the proposition we introduce additional notation which we will use throughout the remainder of the chapter. Let

$$T_t^k(s, a) = \sum_{i=0}^{k-1} \mathbb{I}\left\{\left(s_t^i, a_t^i\right) = (s, a)\right\}, \forall t, s, a \tag{2.5}$$

denote the total number of times that (s, a) has been observed in $\mathcal{D}$ (across all trajectories) up to time t in episode k and $\mathbb{I}\{\cdot\}$ is the indicator function.

Similarly let

$$T_t^k(s,a,s') = \sum_{i=0}^{k-1} \mathbb{I}\left\{\left(s_t^i, a_t^i, s_{t+1}^i\right) = (s,a,s')\right\}, \forall t, s, a, s' \tag{2.6}$$

denote the number of times action a is taken in s to transition to s'. Finally we define the state sample $T_t^k(s) = \sum_a T_t^k(s,a)$ as the total number of times any state is visited and an action is taken in that state.

Proposition 2. *Let there be an oracle which knows the state-action variances and transition probabilities of the* L*-depth tree MDP* $\mathbf{T}$*. Let the oracle take actions in the proportions given by Theorem 1. Let* $\mathcal{D}$ *be the observed data over* n *state-action-reward samples such that* $n = KL$*. Then the oracle suffers an MSE of*

$$\mathcal{L}_n^* = \sum_{\ell=1}^{L} \left[\frac{B^2(s_i^\ell)}{T_L^{*,K}(s_i^\ell)} + \gamma^2 \sum_a \pi^2(a|s_i^\ell) \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) \frac{B^2(s_j^{\ell+1})}{T_L^{*,K}(s_j^{\ell+1})} \right]. \tag{2.7}$$

where, $T_L^{*,K}(s_i^\ell)$ *denotes the optimal state samples of the oracle at the end of episode* K*.*

The proof is given in Section A.6. From Proposition 2 we see that the MSE of the oracle goes to 0 as the number of episodes $K \to \infty$, and $T_L^{*,K}(s_i^\ell) \to \infty$ simultaneously for all $s_i^\ell \in \mathcal{S}$. Observe that if for every state s the total state counts $T_L^{*,K}(s) = cn$ for some constant $c > 0$ then the loss of the oracle goes to 0 at the rate $O(1/n)$.

Reduced Variance Sampling

The oracle data collection strategy provides intuition for optimal data collection for minimal-variance policy evaluation, however, it is *not* a practical strategy itself as it requires σ and P to be known. We now introduce a practical data collection algorithm – **Re**duced **Var**iance Sampling (ReVar) – that is agnostic to σ and P. Our algorithm follows the proportions given by

Theorem 1 with the true reward variances replaced with an upper confidence bound and the true transition probabilities replaced with empirical frequencies. Formally, we define the desired proportion for action a in state s_i^ℓ after t steps as $\widehat{b}_{t+1}^k(a|s_i^\ell) \propto$

$$\sqrt{\pi^2(a|s_i^\ell)\left[\widehat{\sigma^u}_t^{(2),k}(s_i^\ell,a)+\gamma^2\sum_{s_j^{\ell+1}}\widehat{P}_t^k(s_j^{\ell+1}|s_i^\ell,a)\widehat{B}_t^{(2),k)}(s_j^{\ell+1})\right]}, \tag{2.8}$$

The upper confidence bound on the variance $\sigma^2(s_i^\ell,a)$, denoted by $\widehat{\sigma^u}_{t-1}^{(2),k}(s_i^\ell,a) = (\widehat{\sigma^u}_t^k(s_i^\ell,a))^2$, is defined as:

$$\widehat{\sigma^u}_t^k(s_i^\ell,a) \coloneqq \widehat{\sigma}_t^k(s_i^\ell,a)+2c\sqrt{\frac{\log(SAn(n+1)/\delta)}{T_t^k(s_i^\ell,a)}} \tag{2.9}$$

where, $\widehat{\sigma}_t^k(s_i^\ell,a)$ is the plug-in estimate of the standard deviation $\sigma(s_i^\ell,a)$, $c>0$ is a constant depending on the boundedness of the rewards to be made explicit later, and $n = KL$ is the total budget of samples. Using an upper confidence bound on the reward standard deviations captures our uncertainty about $\sigma(s_i^\ell,a)$ needed to compute the true optimal proportions. The state transition model is estimated as:

$$\widehat{P}_t^k(s_j^{\ell+1}|s_i^\ell,a) = \frac{T_t^k(s_i^\ell,a,s_j^{\ell+1})}{T_t^k(s_i^\ell,a)} \tag{2.10}$$

where, $T_t^k(s_i^\ell,a,s_j^{\ell+1})$ is defined in (2.6). Further in (2.8), $\widehat{B}_t^k(s_j^{\ell+1})$ is the plug-in estimate of $B(s_j^{\ell+1})$. Observe that for all of these plug-in estimates we use all the past history till time t in episode k to estimate these statistics.

Eq. (2.8) allows us to estimate the optimal proportion for all actions in any state. To match these proportions, rather than sampling from

$\widehat{b}_{t+1}^{k}(a|s_i^{\ell})$, ReVar takes action I_{t+1}^{k} at time $t+1$ in episode k according to:

$$I_{t+1}^{k} = \arg\max_{a} \left\{ \frac{\widehat{b}_{t}^{k}(a|s_i^{\ell})}{T_t^{k}(s_i^{\ell}, a)} \right\}. \tag{2.11}$$

This action selection rule ensures that the ratio $\widehat{b}_{t}^{k}(a|s_i^{\ell})/T_t(s_i^{L}, a) \approx 1$. It is a deterministic action selection rule and thus avoids variance due to simply sampling from the estimated optimal proportions. Note that in the terminal states, s_i^{L}, the sampling rule becomes

$$I_{t+1}^{k} = \arg\max_{a} \left\{ \frac{\pi(a|s_i^{L})\widehat{\sigma^{u}}_{t}^{k}(s_i^{L}, a)}{T_t^{k}(s_i^{L}, a)} \right\}$$

which matches the bandit sampling rule of Carpentier and Munos (2011, 2012).

We give pseudocode for ReVar in Algorithm 1. The algorithm proceeds in episodes. In each episode we generate a trajectory from the starting state s_1^1 (root) to one of the terminal state s_j^{L} (leaf). At episode k and time-step t in some arbitrary state s_i^{ℓ} the next action I_{t+1} is chosen based on (2.11). The trajectory generated is added to the dataset $\mathcal{D}$. At the end of the episode we update the model parameters, i.e. we estimate the $\widehat{\sigma}_t^{k}(s_i^{\ell}, a)$, and $\widehat{P}_t^{k}(s_i^{\ell+1}|s_j^{\ell}, a)$ for each state-action pair. Finally, we update $\widehat{b}_1^{k+1}(a|s_{\ell}^{i})$ for the next episode using eq. (2.9).

Regret Analysis

We now theoretically analyze ReVar by bounding its regret with respect to the oracle behavior policy. We analyze ReVar under the assumption that P is known and so we are only concerned with obtaining accurate estimates of the reward means and variances. This assumption is only made for the regret analysis and is *not* a fundamental requirement of ReVar. Though somewhat restrictive, the case of known state transitions is still interesting

Algorithm 1 Reduced **Var**iance Sampling (ReVar)

Input: Number of trajectories to collect, K.
Output: Dataset $\mathcal{D}$.
Initialize $\mathcal{D} = \emptyset$, $\widehat{b}_1^0(a|s_i^\ell)$ uniform over all actions in each state.
for $k \in 0, 1, \ldots, K$ **do**
    Generate trajectory $H^k := \{S_t, I_t, R(I_t)\}_{t=1}^L$ by selecting I_t according to (2.11).
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(H^k, \widehat{b}_L^k)\}$
    Update model parameters and estimate $\widehat{b}_1^{k+1}(a|s_i^\ell)$ for each (s_i^ℓ, a).
    Update $\widehat{b}_1^{k+1}(a|s_i^\ell)$ from level L to 1 following (2.8).
Return Dataset $\mathcal{D}$ to evaluate policy π.

as it arises in practice when state transitions are deterministic or we can estimate P much easier than we can estimate the reward means.

We first define the notion of regret of an algorithm compared to the oracle MSE $\mathcal{L}_n^*$ in (2.7) as follows:

$$\mathcal{R}_n = \mathcal{L}_n - \mathcal{L}_n^*$$

where, n is the total budget, and $\mathcal{L}_n$ is the MSE at the end of episode K following the sampling rule in (2.8). We make the following assumption that rewards are bounded:

Assumption 2. *The reward from any state-action pair has bounded range, i.e.,* $R_t(s, a) \in [-\eta, \eta]$ *almost surely at every time-step* t *for some fixed* $\eta > 0$.

Note that this is a common assumption in the RL literature (Munos, 2005; Agarwal et al., 2019). The reward can also be multi-modal as long as it is bounded. Then the regret of ReVar over a L-depth deterministic tree is given by the following theorem.

Theorem 2. *Let the total budget be* $n = KL$ *and* $n \geqslant 4SA$. *Then the total regret in a deterministic* L*-depth* **T** *at the end of* K*-th episode when taking actions*

according to (2.8) *is given by*

$$\mathcal{R}_n \leqslant \widetilde{O}\left(\frac{B^2_{s_1^1}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b^{*,3/2}_{min}(s_1^1)} + \gamma\sum_{\ell=2}^{L}\max_{s_j^\ell,a}\pi(a|s_1^1)P(s_j^\ell|s_1^1,a)\frac{B^2_{s_j^\ell}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b^{*,3/2}_{min}(s_j^\ell)}\right)$$

where, the $\widetilde{O}$ *hides other lower order terms and* $B_{s_i^\ell}$ *is defined in* (2.4) *and* $b^*_{min}(s) = \min_a b^*(a|s)$.

Note that if $L = 1$, $|\mathcal{S}| = 1$, we recover the bandit setting and our regret bound matches the bound in Carpentier and Munos (2011). Note that MSE using data generated by any policy decays at a rate no faster than $O(n^{-1})$, the parametric rate. The key feature of ReVar is that it converges to the oracle policy. This means that asymptotically, the MSE based on ReVar will match that of the oracle. Theorem 2 shows that the regret scales like $O(n^{-3/2})$ if we have the $b^*_{min}(s)$ over all states $s \in \mathcal{S}$ as some reasonable constant $O(1)$. In contrast, suppose we sample trajectories from a suboptimal policy, i.e., a policy that produces an MSE worse than that of the oracle for every n. This MSE gap never diminishes, so the regret cannot decrease at a rate faster than the oracle rate of $O(n^{-1})$. Finally, note that the regret bound in Theorem 2 is a problem dependent bound as it involves the parameter $b^*_{min}(s)$.

Proof (Overview): We decompose the proof into several steps. We define the good event ξ_δ based on the state-action-reward samples $\mathcal{D}$ that holds for all episode k and time t such that $|\widehat{\sigma}^k_t(s,a) - \sigma(s,a)| \leqslant \epsilon$ for some $\epsilon > 0$ with probability $1-\delta$ made explicit in Theorem A.6 . Now observe that MSE of ReVar is

$$\begin{aligned}\mathcal{L}_n &= \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)) - v^\pi(s_1^1))\right)^2\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)) - v^\pi(s_1^1))\right)^2 \mathbb{I}\{\xi_\delta\}\right] + \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)) - v^\pi(s_1^1))\right)^2 \mathbb{I}\left\{\xi_\delta^C\right\}\right]\end{aligned} \tag{2.12}$$

Note that here we are considering a known transition function P.The first term in (2.12) can be bounded using

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)) - v^{\pi}(s_1^1)\right)\right)^2 \mathbb{I}\{\xi_\delta\}\right] &= \mathrm{Var}[Y_n(s_1^1)]\mathbb{E}[T_n^k(s_1^1)] \\
&\leqslant \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_n^{(2),k}(s_1^1,a)}\right]\mathbb{E}[T_n^k(s_1^1,a)] \\
&+ \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} P^2(s_j^2|s_1^1,a) \\
&\quad \cdot \sum_{a'} \pi^2(a'|s_j^2)\left[\frac{\sigma^2(s_j^2,a')}{\underline{T}_n^{(2),k}(s_j^2,a')}\right]\mathbb{E}[T_n^k(s_j^2,a')]
\end{aligned}$$

where, $\underline{T}^{(2),k}(s_1^1,a)$ is a lower bound to $T^{(2),k}(s_1^1,a)$ made explicit in Theorem A.8, and $\underline{T}^{(2),k}(s_j^2,a)$ is a lower bound to $T^{(2),k}(s_1^1,a)$ made explicit in Theorem A.7. We can combine these two lower bounds and give an upper bound to MSE in a two depth $\mathbf{T}$ which is shown Theorem A.9. Finally, for the L depth stochastic tree we can repeatedly apply Theorem A.9 to bound the first term. For the second term we set the $\delta = n^{-2}$ and use the boundedness assumption in Assumption 2 to get the final bound. The proof is given in Section A.8. ■

2.5 Optimal Data Collection Beyond Trees

The tree-MDP model considered above allows us to develop a foundation for minimal-variance data collection in decision problems where actions at one state affect subsequent states. One limitation of this model is that, for any non-initial state, s_i^ℓ, there is only a single state-action path that could have been taken to reach it. In a more general finite-horizon MDP, there could be many different paths to reach the same non-initial state. Unfortunately, the existence of multiple paths to a state introduces cyclical dependencies between states that complicate derivation of the minimal-

variance data collection strategy and regret analysis. In this section, we elucidate this difficulty by considering the class of directed acyclic graph (DAG) MDPs.

In this section we first define a DAG $\mathcal{G} \subset \mathbf{M}$. An illustrative figure of a 3-depth 2-action $\mathcal{G}$ is in Figure A.3 of Section A.9 .

Definition 2.5. (***DAG MDP***) *A DAG MDP follows the same definition as the tree MDP in Theorem 2.1 except* $P(s'|s,a)$ *can be non-zero for any* s *in layer* ℓ, s' *in layer* $\ell+1$, *and any* a, *i.e., one can now reach* s' *through multiple previous state-action pairs.*

Proposition 3. *Let* $\mathcal{G}$ *be a* 3*-depth,* A*-action DAG defined in Theorem 2.5. The minimal-MSE sampling proportions* $b^*(a|s_1^1), b^*(a|s_j^2)$ *depend on themselves such that* $b(a|s_1^1) \propto f(1/b(a|s_1^1))$ *and* $b(a|s_j^2) \propto f(1/b(a|s_j^2))$ *where* $f(\cdot)$ *is a function that hides other dependencies on variances of* s *and its children.*

The proof technique follows the approach of Theorem 2.2 but takes into account the multiple paths leading to the same state. The possibility of multiple paths results in the cyclical dependency of the sampling proportions in level 1 and 2. Note that in $\mathbf{T}$ there is a single path to each state and this cyclical dependency does not arise. The full proof is given in Section A.9. Because of this cyclical dependency it is difficult to estimate the optimal sampling proportions in $\mathcal{G}$. However, we can approximate the optimal sampling proportion that ignores the multiple path problem in $\mathcal{G}$ by using the tree formulation in the following way: At every time t during a trajectory τ^k call the Algorithm 12 in Section A.10 to estimate $B_0(s)$ where $B_{t'}(s) \in \mathbb{R}^{L\times|\mathcal{S}|}$ stores the expected standard deviation of the state s at iteration t'. After L such iteration we use the value $B_0(s)$ to estimate $b(a|s)$ as follows:

$$b^*(a|s) \propto \sqrt{\pi^2(a|s)\left[\sigma^2(s,a)+\gamma^2\sum_{s'}P(s'|s,a)B_0^2(s)\right]}.$$

Note that for a terminal state s we have the transition probability $P(s'|s,a) = 0$ and then the $b(a|s) = \pi(a|s)\sigma(s,a)$. This iterative procedure follows from the tree formulation in Theorem 1 and is necessary in $\mathcal{G}$ to take into account the multiple paths to a particular state. Also observe that in Algorithm 12 we use value-iteration for the episodic setting (Sutton and Barto, 2018) to estimate the the optimal sampling proportion iteratively.

2.6 Empirical Study

We next verify our theoretical findings with simulated policy evaluation tasks in both a tree MDP and a non-tree GridWorld domain. Our experiments are designed to answer the following questions: 1) can ReVar produce policy value estimates with MSE comparable to the oracle solution? and 2) does our novel algorithm lower MSE relative to on-policy sampling of actions? Full implementation details are given in Section A.10.

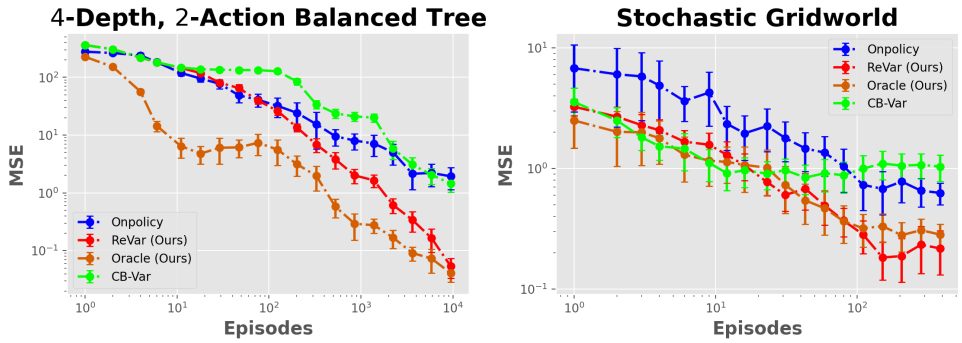


Figure 2.2: (Left) Deterministic 4-depth Tree. (Right) Stochastic gridworld. The vertical axis gives MSE and the horizontal axis is the number of episodes collected. Axes use a log-scale and confidence bars show one standard error.

Experiment 1 (Tree): In this setting we have a 4-depth 2-action deterministic tree MDP $\mathbf{T}$ consisting of 15 states. Each state has a low variance arm with $\sigma^2(s,1) = 0.01$ and high target probability $\pi(1|s) = 0.95$

and a high variance arm with $\sigma^2(s,1) = 20.0$ and low target probability $\pi(2|s) = 0.05$. Hence, the Onpolicy sampling which samples according to π will sample the second (high variance) arm less and suffer a high MSE. The CB-Var policy is a bandit policy that uses an empirical Bernstein Inequality (Maurer and Pontil, 2009) to sample an action without looking ahead and suffers high MSE. The Oracle has access to the model and variances and performs the best. ReVar lowers MSE comparable to Onpolicy and CB-Var and eventually matches the oracle's MSE.

Experiment 2 (Gridworld): In this setting we have a 4×4 stochastic gridworld consisting of 16 grid cells. Considering the current episode timestep as part of the state, this MDP is a DAG MDP in which there are multiple path to a single state. There is a single starting location at the top-left corner and a single terminal state at the bottom-right corner. Let $\mathbf{L}, \mathbf{R}, \mathbf{D}, \mathbf{U}$ denote the left, right, down and up actions in every state. Then in each state the right and down actions have low variance arms with $\sigma^2(s,\mathbf{R}) = \sigma^2(s,\mathbf{D}) = 0.01$ and high target policy probability $\pi(\mathbf{R}|s) = \pi(\mathbf{D}|s) = 0.45$. The left and top actions have high variance arms with $\sigma^2(s,\mathbf{L}) = \sigma^2(s,\mathbf{U}) = 0.01$ and low target policy probability $\pi(\mathbf{L}|s) = \pi(\mathbf{U}|s) = 0.05$. Hence, Onpolicy which goes right and down with high probability (to reach the terminal state) will sample the low variance arms more and suffer a high MSE. Similar to above, CB-Var fails to look ahead when selecting actions and thus suffers from high MSE. ReVar lowers MSE compared to Onpolicy and CB-Var and actually matches and then reduces MSE compared to the Oracle. We point out that the DAG structure of the Gridworld violates the tree-structure under which Oracle and ReVar were derived. Nevertheless, both methods lower MSE compared to Onpolicy.

2.7 Conclusion And Future Works

This chapter has studied the question of how to take actions for minimal-variance policy evaluation of a fixed target policy. We developed a theoretical foundation for data collection in policy evaluation by deriving an oracle data collection policy for the class of finite, tree-structured MDPs. We then introduced a practical algorithm, ReVar, that approximates the oracle strategy by computing an upper confidence bound on the variance of the future cumulative reward at each state and using this bound in place of the true variances in the oracle strategy. We bound the finite-sample regret (excess MSE) of our algorithm relative to the oracle strategy. We also present an empirical study where we show that ReVar decreases the MSE of policy evaluation relative to several baseline data collection strategies including on-policy sampling. In the future, we would like to extend our derivation of optimal data collection strategies and regret analysis of ReVar to a more general class of MDPs, in particular, relaxing the tree structure and also considering infinite-horizon MDPs. Finally, real world problems often require function approximation to deal with large state and action spaces. This setting raises new theoretical and implementation challenges for ReVar where we intend to incorporate experimental design approaches (Pukelsheim, 2006; Mason et al., 2021; Mukherjee et al., 2022b). Another interesting direction is to incorporate structure in the reward distribution of arms (Gupta et al., 2021, 2020b). Addressing these challenges is an interesting direction for future work.

3 SPEED: OPTIMAL DESIGN FOR POLICY EVALUATION IN LINEAR HETEROSCEDASTIC BANDITS

Bandit policy optimization has been applied in various applications such as web marketing (Bottou et al., 2013), web search (Li et al., 2011), and healthcare recommendations (Zhou et al., 2017). In practice, before widely deploying a learned policy, it is often necessary to have an accurate estimation of its performance (i.e., expected reward). To this effect, *policy evaluation* is often a critical step as it allows practitioners to determine if a learned policy truly represents improved task performance. While off-policy evaluation has been extensively studied as a potential solution (Dudík et al., 2014; Li et al., 2015; Swaminathan et al., 2017; Wang et al., 2017; Su et al., 2020; Kallus et al., 2021; Cai et al., 2021), in practice, some amount of online evaluation is often required before widescale deployment. For instance, in web-marketing it is common to run an A/B test with a subset of users before a potential new policy is deployed for all users (Kohavi and Longbotham, 2017). When online policy evaluation is required, we desire methods that provide an accurate estimate of policy performance with a minimal amount of data collected. The default choice for online policy evaluation is to simply run the target policy and average the resulting rewards. However, this approach is sub-optimal when the action space is large or different actions have reward distributions with different variances.

In this paper, we formulate a new experimental design for allocating action samples so as to obtain minimal mean squared error (MSE) for policy evaluation. Specifically, we consider optimal policy evaluation under the following linear heteroscedastic bandit model.

Let $\mathcal{A}$ be the set of *actions* and each $a \in \mathcal{A}$ is associated with a feature vector $\mathbf{x}(a) \in \mathbb{R}^d$ and $|\mathcal{A}| = A$. The reward distribution for each action a has mean $\theta_*^\top \mathbf{x}(a)$, for some $\theta_* \in \mathbb{R}^d$. Often the variance of the reward

distribution is assumed to be the same for all actions, but in this paper, we depart from this assumption. We consider the setting that the variance is governed by a quadratic function of the form $\mathbf{x}(a)^\top \mathbf{\Sigma}_* \mathbf{x}(a)$, for some symmetric positive definite matrix $\mathbf{\Sigma}_* \in \mathbb{R}^{d \times d}$. This assumption allows us to capture problems in which both the mean reward and the variance may depend on the action taken, but both vary smoothly in $\mathbf{x}(a)$.

We briefly contrast our studied setting with other work. In policy evaluation, the common metric of algorithm performance is regret with respect to the mean squared error of an oracle algorithm that has knowledge of the variances of different reward distributions (i.e., knows Σ^*). There has been an increasing focus on studying data collection for policy evaluation in bandit settings (Zhu and Kveton, 2021, 2022a; Wan et al., 2022) and there has been some theoretical progress (Chaudhuri et al., 2017; Fontaine et al., 2021). Several works (Antos et al., 2008; Carpentier and Munos, 2012; Carpentier et al., 2015; Fontaine et al., 2021) have shown that in the classical bandit setting a regret of $\widetilde{O}(An^{-3/2})$ is possible where n is the total budget of actions that can be tried and $\widetilde{O}$ hides logarithmic factors. These works have also shown that simply running the target policy to take actions results in a slower decrease of regret at the rate of $\widetilde{O}(An^{-1})$. Note that collecting data through running the target policy is called on-policy sampling. The work of Zhu and Kveton (2022a); Wan et al. (2022) studies the same setting under safety constraints and provides finite error bounds. However, none of the above works provides a finite-time regret guarantee for data collection for policy evaluation in the heteroscedastic linear bandit setting.

The closest works to ours (Antos et al., 2008; Carpentier and Munos, 2012; Carpentier et al., 2015; Fontaine et al., 2021) either consider unstructured settings or consider the classical bandit setting. As many real-world bandit applications have $d \ll A$, a natural question arises as to how to build an algorithm for policy evaluation in the heteroscedastic linear

bandit setting with unknown θ_* and $\mathbf{\Sigma}_*$ that can leverage the structure. Further, we want the regret of such an algorithm to decrease at a rate faster than $\widetilde{O}(n^{-1})$ (the on-policy regret rate) and to scale with the dimension d instead of actions as $A \gg d$. Note that the regret should scale at least by d^2 because the learner needs to probe in d^2 dimensions to estimate $\mathbf{\Sigma}_* \in \mathbb{R}^{d\times d}$ (Wainwright, 2019). Thus, the goal of our work is to answer the question:

> *Can we design an algorithm to collect data for policy evaluation that adapts to the variance of each action, and its regret decreases at a rate faster than* $\widetilde{O}(d^2 n^{-1})$*?*

In this paper, we answer this question affirmatively. We make the following novel contributions to the growing literature on online policy evaluation:

1. We are the first to formulate the policy evaluation problem for heteroscedastic linear bandit setting where the variance of each action $a \in \mathcal{A}$ depends on a lower dimensional co-variance matrix parameter $\mathbf{\Sigma}_* \in \mathbb{R}^{d\times d}$ such that variance $\sigma^2(a) = \mathbf{x}(a)^\top \mathbf{\Sigma}_* \mathbf{x}(a)$. This is a more general heteroscedastic linear bandit setting than studied in Chaudhuri et al. (2017); Kirschner and Krause (2018); Fontaine et al. (2021), and different than the time-dependent variance model of Zhang et al. (2021); Zhao et al. (2022).

2. We characterize the MSE in this setting and show that the optimal design, denoted as **P**olicy **E**valuation (PE) Optimal design that minimizes the MSE is different than A-, D-, E-, G-optimality (Pukelsheim, 2006). We establish several key properties of this novel PE-Optimal design and discuss how we can solve for the design efficiently.

3. Finally, we propose the agnostic algorithm, SPEED, that does not know the underlying covariance matrix $\mathbf{\Sigma}_*$. SPEED tracks the oracle design and we analyze its MSE. We then bound the regret of SPEED compared to

an oracle strategy that follows the optimal design with the knowledge of $\mathbf{\Sigma}_*$. We show that the regret scales as $O(\frac{d^3 \log(n)}{n^{3/2}})$ which is an improvement over the regret for the stochastic non-structured bandit setting which scales as $O(\frac{A \log(n)}{n^{3/2}})$ (Carpentier and Munos, 2011, 2012; Carpentier et al., 2015; Fontaine et al., 2021). Hence, we answer positively to our main query. We also prove the first lower bound for this setting that scales as $\Omega(\frac{d^2 \log(n)}{n^{3/2}})$. Finally, we conduct experiments on synthetic and real-life data sets and show that SPEED lowers the MSE of policy evaluation compared to baseline methods. We discuss more related works and motivations in Section 3.2.

3.1 Preliminaries

We study the linear bandit setting where the expected reward for each action is assumed to be a linear function (Mason et al., 2021; Jamieson and Jain, 2022). We define $[m] \coloneqq [1, 2, \ldots, m]$. We denote the action space as $\mathcal{A}$ and $|\mathcal{A}| = A$. Actions are indexed by $a \in [A]$, and each action a is associated with a feature vector $\mathbf{x}(a) \in \mathbb{R}^d$ with dimension $d \ll A$. Denote by $\triangle(\mathcal{A})$ the probability simplex over the action space $\mathcal{A}$ and a policy $\pi \in \triangle(\mathcal{A})$ as a mapping $\pi : \mathcal{A} \to [0, 1]$ such that $\sum_a \pi(a) = 1$.

Data collection is performed over n rounds of action selection. Specifically, at each round $t \in [n]$, the selected action a_t yields a reward: $r_t = \mathbf{x}(a_t)^\top \theta_* + \eta_t$, where $\theta_* \in \mathbb{R}^d$ is the *unknown* reward parameter, and η_t is zero-mean noise with variance $\sigma^2(a_t)$ and we further assume that η_t is κ^2-subgaussian. We assume that for each action $a \in \mathcal{A}$ the variance $\sigma^2(a)$ has a lower-dimensional structure such that $\sigma^2(a) = \mathbf{x}(a)^\top \mathbf{\Sigma}_* \mathbf{x}(a)$ where $\mathbf{\Sigma}_* \in \mathbb{R}^{d \times d}$ is an *unknown* variance parameter. Observe that the variance depends on the action features, which is called the heteroscedastic noise model (Greene, 2002; Chaudhuri et al., 2017) which differs from the unknown time-dependent variance model of Zhang et al. (2021); Zhao

et al. (2022). Moreover, Fontaine et al. (2021) do not consider structure in variances and Chaudhuri et al. (2017) only consider a special case of our setting where $\mathbf{\Sigma}_*$ is a rank-1 matrix. We also assume that the norms of the features are bounded such that $H_L^2 \leqslant \|\mathbf{x}(a)\|^2 \leqslant H_U^2$ for all $a \in \mathcal{A}$. In our heteroscedastic linear bandit setting selecting any action gives information about θ_* and also gives information about the noise covariance matrix $\mathbf{\Sigma}_*$.

The value of a policy π is defined as $v(\pi) \coloneqq \mathbb{E}[R_t]$ where the expectation is taken over $a_t \sim \pi, R_t \sim \mathbf{x}(a_t)^\top \theta_* + \eta_t$. In the policy evaluation problem, we are given a fixed, target policy π and asked to estimate $v(\pi)$. Estimating $v(\pi)$ requires a dataset of actions and their associated rewards, $\mathcal{D} \coloneqq \{(a_1, r_1, ..., a_n, r_n)\}$, which is collected by executing some policy. We refer to the policy that collects $\mathcal{D}$ as the *behavior policy*, denoted by $\mathbf{b} \in \triangle(\mathcal{A})$. We then define the value estimate of a policy π as Y_n, where n is the sample budget. The exact nature of the value estimate for the linear bandit setting will be made clear in Section 3.4. Our goal is to choose a behavior policy that minimizes the mean squared error (MSE) defined as $\mathbb{E}_{\mathcal{D}}[(Y_n - v(\pi))^2]$, where the expectation is over the collected data set $\mathcal{D}$.

We now state an assumption on the boundedness on the variance of each action $a \in [A]$. Let the singular value decomposition of $\mathbf{\Sigma}_*$ be $\overline{U}\mathbf{D}\mathbf{P}^\top$ with orthogonal matrices $\overline{U}, \mathbf{P}^\top$ and $\mathbf{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ where $\{\lambda_i\}$ are singular values. It follows that $\sigma_{\min}^2 \leqslant \sigma^2(a) \leqslant \sigma_{\max}^2$ where $\sigma_{\min}^2 = \min_i |\lambda_i| H_L^2$ and $\sigma_{\max}^2 = \max_i |\lambda_i| H_U^2$ (see Theorem B.4).

Assumption 3. *We assume that $\mathbf{\Sigma}_*$ has its minimum and maximum eigenvalues bounded such that for every action $a \in [A]$ the following holds $\sigma_{\min}^2 \leqslant \sigma^2(a) \leqslant \sigma_{\max}^2$.*

3.2 Related Work

Our work is most closely related to existing work on data collection for policy evaluation. Perhaps the most natural choice of behavior policy is

to simply run the target policy, i.e., on-policy data collection (Sutton and Barto, 2018). The works in adaptive Monte Carlo for bandits (Oosterhuis and de Rijke, 2020; Tucker and Joachims, 2022a) and MDPs (Hanna et al., 2017b; Ciosek and Whiteson, 2017; Bouchard et al., 2016; Zhong et al., 2022a; Corrado and Hanna, 2023) have shown how to lower the variance of Monte Carlo estimation through the choice of behavior policy. In contrast to these works, we consider estimating $v(\pi)$ by estimating the reward distributions rather than using Monte Carlo estimation. Such *certainty-equivalence* estimators take advantage of the setting's structure and are thus typically of lower variance than Monte Carlo estimators (Sutton and Barto, 2018). The work of Wan et al. (2022) studies a different estimator for reducing the variance of the importance sampling in constrained MDP setting whereas we study certainty equivalence estimator. Another set of work has studied sample allocation for stratified Monte Carlo estimators – a problem that is formally equivalent to behavior policy selection for policy evaluation in the bandit setting with linearly independent arms (Antos et al., 2008; Carpentier et al., 2015). This line of work was recently extended to tabular, tree-structured MDPs by Mukherjee et al. (2022a). In contrast, we consider the structured linear bandit setting which incorporates generalization across actions. Li et al. (2024b) use A-optimal design to find an optimal behavior policy for the doubly robust estimator. Their focus is different though as they consider tabular MDPs rather than linear heteroscedastic bandits.

Our work is closely related to optimal experimental design and active learning literature. We formulate determining the optimal behavior policy in the bandit setting as an optimal design problem. In contrast to prior work, we introduce a new type of optimality that is tailored to the policy evaluation problem. We are also, to the best of our knowledge, the first to consider both heteroscedastic noise and weighted least squares estimators in formulating our design. The heteroscedastic noise model and weighted

least squares estimator have been considered by Chaudhuri et al. (2017) in the active learning literature and in linear bandit setting by Kirschner and Krause (2018) using information directed sampling. In contrast to these works (and the active learning setting in general), we aim to minimize the weighted error $\sum_{a \in \mathbf{A}} \pi(a)\mathbf{x}(a)^\top(\theta^* - \widehat{\theta})^2$ whereas in the active learning setting the goal is to minimize $\|\theta^* - \widehat{\theta}\|^2$ which results in A-optimal design (Fontaine et al., 2021; Pukelsheim, 2006). Moreover the regret bounds in Fontaine et al. (2021) holds for $d = |\mathcal{A}|$. Riquelme et al. (2017) extends the results of Carpentier and Munos (2011) to a different linear regression setting than ours but under the homoscedastic noise model.

Data collection for policy evaluation is also related to the problem of exploration for policy learning in MDPs or best-arm identification in bandits. In those contexts, the aim of exploration is to find the optimal policy and the exploration-exploitation trade-off describes the tension between reducing uncertainty and focusing on known promising actions. In bandits, the exploration-exploitation trade-off is often navigated under the "**O**ptimism in the **F**ace of **U**ncertainty" principle using techniques such as UCB (Lai and Robbins, 1985; Auer et al., 2002; Abbasi-Yadkori et al., 2011) or Thompson Sampling (Thompson, 1933; Agrawal and Goyal, 2012). In contrast to the standard exploration problem, we focus on evaluating a fixed policy. Instead of balancing exploration and exploitation, a behavior policy for policy evaluation should take actions that reduce uncertainty about $v(\pi)$ with emphasis on actions that have high probability under π. Also, note that heteroscedastic bandits have been studied from the perspective of policy improvement (Kirschner and Krause, 2018; Zhao et al., 2022) however, in this paper we focus on optimal data collection for policy evaluation.

We note that heteroscedasticity is also studied for the policy improvement setup (Kirschner and Krause, 2018; Zhou and Gu, 2022; Zhou et al., 2021; Zhang et al., 2021; Zhao et al., 2022). In these prior works the re-

ward variances are time-dependent as opposed to the quadratic structure studied in this paper. Note that policy improvement requires a different approach than policy evaluation. These works build tight confidence sets around the unknown model parameter θ_* by employing weighted ridge regression involving an estimated upper bound to the time-dependent variances. However, in our setting, the variances of each action share the unknown low dimensional co-variance matrix $\mathbf{\Sigma}_*$. Hence we deviate from these approaches and employ an alternating OLS-WLS estimation to learn the underlying parameter $\mathbf{\Sigma}_*$.

3.3 Optimal Design for Policy Evaluation

In this section, we first discuss why following the target policy to take actions can lead to a poor estimation of the value of the policy. This discussion motivates how a different behavior policy can produce more accurate estimates of the target policy's value. After this motivation, we derive an expression for policy evaluation error in terms of the behavior sampling proportion $\mathbf{b} \in \triangle(\mathcal{A})$, target policy π, and action features $\mathbf{x}(a) \in \mathbb{R}^d$. We call the minimizer of this expression the "optimal design" (Pukelsheim, 2006) as it minimizes the mean squared error for policy evaluation. We then analyze the error incurred by an oracle that can compute and follow the optimal behavior policy through knowledge of problem-dependent parameters.

Motivating Example: Consider the linear bandit environment where $d = 2$ and $A = 100$ actions. Let one action be along the x-axis, one action along the y-axis, and 98 actions along the direction of $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. Assume θ_* is in the direction of x-axis (so action 1 is the optimal action). A similar canonical linear bandit setting has been studied by Fiez et al. (2019); Katz-Samuels et al. (2020). Consider a target policy π such that $\pi(1) = 0.9$ and it distributes 0.1 probability equally on the remaining actions. In

this case, just running the target policy π for n rounds leads to sampling uninformative actions for identifying θ_*. In fact, in our experiments, we show that the estimate $v(\pi)$ will be inaccurate compared to running the optimal behavior policy (called Oracle policy; see Figure 3.1 top-left).

Now suppose we divide the budget of n samples across the actions, and let $T_n(1), T_n(2), \ldots, T_n(A)$ be the number of samples allocated to actions $1, 2, \ldots, A$ at the end of n rounds. After observing n samples, let the *weighted* least square estimate (WLS) be:

$$\widehat{\theta}_n \coloneqq \arg\min_{\theta} \sum_{t=1}^{n} \tfrac{1}{\sigma^2(a_t)} (r_t - \mathbf{x}(a_t)^\top \theta)^2 \tag{3.1}$$

where a_t is the action sampled at round t and $\sigma^2(a_t)$ is the variance of action a_t. Also note that this is an unbiased estimator of θ_* (see Theorem B.6). In a linear bandit, we can define the value estimate of a *target policy* as $Y_n \coloneqq \sum_a \mathbf{w}(a)^\top \widehat{\theta}_n$, where $\mathbf{w}(a) \coloneqq \pi(a)\mathbf{x}(a)$ is the expected feature for each action $a \in \mathcal{A}$ under the target policy, and $\widehat{\theta}_n$ is an unbiased estimate of θ_* computed with n samples in $\mathcal{D}$. As $\widehat{\theta}_n$ is an unbiased estimate, we have that $\mathbb{E}_{\mathcal{D}}[Y_n] = \sum_{a=1}^{A} \mathbf{w}(a)^\top \theta_* = v(\pi)$. Since we have an unbiased estimator of $v(\pi)$, minimizing the MSE is equivalent to minimizing the variance, $\min \mathbb{E}_{\mathcal{D}}\left[(Y_n - \mathbb{E}[Y_n])^2 \right] = \min \mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_n - \theta_*) \right)^2 \right]$, where the minimization is with respect to the data distribution $\mathcal{D}$, which is determined by the behavior policy. In general, the behavior policy that minimizes the MSE may be different from the target policy. To identify this optimal behavior policy, following the optimal design literature (Pukelsheim, 2006; Fedorov, 2013) we define the design or information matrix $\mathbf{A}_{\mathbf{b},\Sigma_*} \in \mathbb{R}^{d \times d}$ w.r.t. each $\mathbf{b} \in \Delta(\mathcal{A})$ as

$$\mathbf{A}_{\mathbf{b},\Sigma_*} = \sum_{a \in \mathcal{A}} \mathbf{b}(a) \left(\tfrac{\mathbf{x}(a)}{\sigma(a)} \right) \left(\tfrac{\mathbf{x}(a)}{\sigma(a)} \right)^\top = \sum_{a \in \mathcal{A}} \mathbf{b}(a) \widetilde{\mathbf{x}}(a) \widetilde{\mathbf{x}}(a)^\top \tag{3.2}$$

where $\widetilde{\mathbf{x}}(a) = \mathbf{x}(a)/\sigma(a)$. Observe that our design matrix in (3.2) captures

the information about the action features $\mathbf{x}(a)$, and variance $\sigma^2(a)$ and weights them by the sampling proportion $\mathbf{b}(a)$. Then in the following proposition, we exactly characterize the MSE with respect to the design matrix $\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}$, target policy π and action features $\mathbf{x}$. Moving forward, we will use the term *loss* interchangeably with MSE.

Proposition 1. *Let $\widehat{\theta}_n$ be the Weighted Least Square (WLS) estimate (3.1) of θ_* after observing n samples and define $\mathbf{w}(a) = \pi(a)\mathbf{x}(a)$. Define the design matrix as $\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}$ (see (3.2)). Then the loss is given by*

$$\mathbb{E}_{\mathcal{D}}\Big[\Big(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\Big)^2\Big] = \underbrace{\frac{1}{n}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*} \mathbf{w}(a')}_{:=\mathcal{L}_n(\pi,\mathbf{b},\boldsymbol{\Sigma}_*)}.$$

Proof (Overview) The key idea is to show that the linear model yields for each action $a \in [A]$, $\widetilde{Y}_n(a) = \widetilde{\mathbf{x}}_n(a)^\top \theta^\star + \widetilde{\eta}_n(a)$ where we define

$$\widetilde{Y}_n(a) = \sum_{i=1}^{T_n(a)} \frac{R_i(a)}{\sigma(a)\sqrt{T_n(a)}}, \widetilde{\mathbf{x}}_n(a) = \frac{\sqrt{T_n(a)}\mathbf{x}(a)}{\sigma(a)},$$
$$\widetilde{\eta}_n(a) = \sum_{i=1}^{T_n(a)} \frac{\eta_i(a)}{\sigma(a)\sqrt{T_n(a)}},$$

with $R_i(a)$ being the reward observed for action a taken for the i-th time, $\eta_i(a)$ being the corresponding noise, and $T_n(a)$ is the number of samples of action a. Next, observe that using the independent noise assumption, we have that $\mathbb{E}[\widetilde{\eta}_n(a)] = 0$ and $\mathrm{Var}[\widetilde{\eta}_n(a)] = 1$. Let $\mathbf{X} = \big(\widetilde{\mathbf{x}}_n(1)^\top, \cdots, \widetilde{\mathbf{x}}_n(A)^\top\big)^\top \in \mathbb{R}^{A\times d}$ be the induced feature matrix of the policy and $\mathbf{Y} = [\widetilde{Y}_n(1), \widetilde{Y}_n(2), \ldots, \widetilde{Y}_n(A)]^\top$. The above weighted least squares (WLS) problem has an optimal unbiased estimator $\widehat{\theta}_n = \big(\mathbf{X}^\top\mathbf{X}\big)^{-1}\mathbf{X}^\top\mathbf{Y}$ (Fontaine et al., 2021). Substituting the definition of $\widehat{\theta}_n$ yields the desired expression of the loss as stated in the proposition. The detailed proof is given in Chapter B. ■

Observe that the loss in our setting depends on the inverse of the design matrix denoted by $\mathbf{A}_{\mathbf{b},\mathbf{\Sigma}_*}^{-1}$, the target policy, as well as features of action pairs $(a, a') \in \mathcal{A} \times \mathcal{A}$. Hence, minimizing the loss is equivalent to minimizing the quantity $1/n(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\mathbf{b},\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a'))$. As this design is different than a number of existing notions of optimality such as D-, E-, T-, or G-optimality (Pukelsheim, 2006; Fedorov, 2013; Jamieson and Jain, 2022), we call this the *PE-Optimal design*. None of these previously proposed designs capture the objective of minimal MSE for policy evaluation. For example, G-optimality (as studied by (Katz-Samuels et al., 2020; Mason et al., 2021; Katz-Samuels et al., 2021; Mukherjee et al., 2023c, 2024f)) minimizes the worst-case error of $\max_{\mathbf{x}(a)} \mathbb{E}_{\mathcal{D}}[(\mathbf{x}(a)^\top(\widehat{\theta}_n - \theta_*))^2]$ by minimizing the quantity $\max_{\mathbf{x}(a)} \mathbf{x}(a)^\top \mathbf{A}_{\mathbf{b}}^{-1} \mathbf{x}(a)$ for homoscedastic noise. The E-optimal design minimizes $\max_{\|\mathbf{u}\| \leqslant 1} \mathbb{E}_{\mathcal{D}}[(\mathbf{u}^\top(\widehat{\theta}_n - \theta_*))^2]$ by minimizing the minimum eigenvalue of the inverse of design matrix (Mukherjee et al., 2022b) and the A-optimal design minimizes $\mathbb{E}_{\mathcal{D}}[(\widehat{\theta}_n - \theta_*)^2]$ by minimizing the trace of the inverse of design matrix (Fontaine et al., 2021).

We now state a few more notations for ease of exposition. Using Proposition 1 we define the optimal behavior policy when the matrix $\mathbf{\Sigma}_*$ is known as:

$$\mathbf{b}_* \coloneqq \arg\min_{\mathbf{b}} \mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*), \tag{3.3}$$

where the loss $\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*)$ is defined in Proposition 1. We define the optimal loss (with knowledge of $\mathbf{\Sigma}_*$) as:

$$\mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) = \min_{\mathbf{b}} \mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*). \tag{3.4}$$

Computation of the optimal design $\mathbf{b}_*$

In this section, we digress a bit to discuss the computational aspect of $\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*)$. Since PE-Optimal design is a new type of design, the natural question to ask is *how to optimize this loss function w.r.t.* $\mathbf{b}$*?* We show in

Proposition 2 that the loss $\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*)$ for any arbitrary design proportion $\mathbf{b} \in \triangle(\mathcal{A})$ is strictly convex with respect to the proportion $\mathbf{b}$. The proposition and its proof are given in Chapter B. Next in Proposition 3 we show that the gradient of the loss function is bounded. Due to space constraints, both propositions and their proofs are given in Chapter B and Chapter B respectively. We first state an assumption that the minimum eigenvalue satisfies $\lambda_{\min}\left(\sum_{a=1}^{A} \mathbf{w}(a)\mathbf{w}(a)^\top\right) > 0$, which is required for proving Proposition 3.

Assumption 4. (***Distribution of*** π) *We assume that the set of actions* a *such that* $\pi(a)>0$, *spans* $\mathbb{R}^d$ *and* $\mathbb{R}^{d\times d}$.

Note that this is a realistic and not a restrictive assumption, since if the target policy never takes an action that is needed to cover some dimension then we can avoid identifying θ_* in that dimension. Using Proposition 2, 3 we can effectively solve the PE-Optimal design with gradient descent approaches (Lacoste-Julien and Jaggi, 2013; Berthet and Perchet, 2017). We capture this convergence guarantee with the assumption of the existence of an approximation oracle.

Assumption 5. (***Approximation Oracle***) *We assume access to an approximation oracle. Given a convex loss function* $\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*)$ *with minimizer* $\mathbf{b}_*$, *the approximation oracle returns a proportion* $\widehat{b}_* = \arg\min_{\mathbf{b}} \mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*)$ *such that* $|\mathcal{L}_n(\pi, \widehat{b}_*, \mathbf{\Sigma}_*) - \mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)| \leqslant \epsilon$.

Therefore from Proposition 2, and 3 and using Assumption 4, and 5 we can get a computationally efficient solution to $\min_{\mathbf{b}\in\Delta(\mathcal{A})} \mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*)$.

3.4 Loss of the Oracle

Recall from Chapter 3, that our final goal is to control the regret (excess loss) of an agnostic algorithm that does not know $\mathbf{\Sigma}_*$, with respect to

an oracle that already knows $\mathbf{\Sigma}_*$. Towards this goal, in this section, we develop our theory for optimal data collection by considering an oracle for the heteroscedastic linear bandit setting. Specifically, we consider an oracle that has knowledge of $\mathbf{\Sigma}_*$ but does not know θ_*. With this knowledge, it can solve Equation (3.3) (Assumption 5) to determine the PE-Optimal design, $\mathbf{b}_*$, that minimizes the loss. The oracle takes actions in proportion $\mathbf{b}_*$ for n samples and then computes the WLS estimate $\widehat{\theta}_n$ using $\mathbf{\Sigma}_*$. The following proposition then bounds the loss of the oracle after n samples.

Proposition 5. (***Oracle Loss***) *Let the oracle sample each action a for $\lceil n\mathbf{b}_*(a)\rceil$ times, where $\mathbf{b}_*$ is the solution to (3.3). Define $\lambda_1(\mathbf{V})$ as the maximum eigenvalue of $\mathbf{V} := \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$. Then the loss satisfies $\mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \leqslant O_{\kappa^2, H_U^2}{}^{1}\left(\frac{d\lambda_1(\mathbf{V})\log n}{n}\right) + O_{\kappa^2,H_U^2}\left(\frac{1}{n}\right)$.*

Proof (Overview) Note that the oracle knows the $\mathbf{\Sigma}_*$ and uses $\widehat{\theta}_n$ in (3.1) to estimate θ_*. We use Theorem B.5 to show that $\mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \leqslant \lambda_1(\mathbf{V})d$ where $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$. The proof follows by showing that $(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*))^2$ is a sub-exponential variable. Then using sub-exponential concentration inequality in Theorem B.2 (Chapter B) and setting $\delta = O(1/n^2)$ we can bound the expected loss with high probability. The full proof is given in Section B.1. ■

Connection to prior work: Prior work has considered a similar oracle for the basic stochastic bandit setting, which is a special case of our setting with $\mathbf{x}(a)$ being a one-hot vector in $\mathbb{R}^A$. In this case, we can see that $\mathbf{b}_* = \arg\min_{\mathbf{b}} \sum_a \frac{\pi^2(a)\sigma^2(a)}{\lceil \mathbf{b}(a)n\rceil}$. This captures the optimal number of times the actions should be pulled weighted by the target policy and their variance. Solving for $\mathbf{b}_*$, we obtain $\mathbf{b}_*(a) \propto \pi^2(a)\sigma^2(a)$. This solution matches the optimal sampling proportion given by Antos et al. (2008); Carpentier and Munos (2011, 2012); Carpentier et al. (2015) for this special case.

[1] Here $O_{\kappa^2,H_U^2}()$ hides the sub-Gaussian factor κ^2 and upper bound H_U^2 on feature norm

The loss in prior work decays at the rate of $\widetilde{O}^2(An^{-1})$ whereas the loss in Proposition 5 decreases at the rate of $\widetilde{O}(dn^{-1})$. Also note the loss in Proposition 5 scales with d instead of d^2 as the oracle knows the $\mathbf{\Sigma}_*$ and does not need to explore d^2 directions to estimate $\mathbf{\Sigma}_*$. So we obtain an equivalence between the PE-Optimal design and the solution from prior work in the basic bandit setting while considering a more general setting.

3.5 SPEED and Regret Analysis When Variance is Unknown

In this section, we first introduce an agnostic algorithm called SPEED for data collection that does not know $\mathbf{\Sigma}_*$, and then analyze its regret. Here, regret refers to the excess loss relative to the oracle that knows $\mathbf{\Sigma}_*$.

Details of Algorithm SPEED

In practice, $\mathbf{\Sigma}_*$ is unknown and so the oracle behavior policy cannot be directly computed. Instead, we first conduct a small amount of exploration to estimate $\mathbf{\Sigma}_*$ and then use the estimate in place of $\mathbf{\Sigma}_*$ in (3.2). Specifically, we define the forced exploration phase as the first Γ rounds in which the algorithm conducts exploration to estimate $\mathbf{\Sigma}_*$. To ensure adequate exploration, we first apply Principal Component Analysis (PCA) on the feature matrix $\mathbf{X}$ and choose the most significant d directions (directions having the highest variance). Then we choose one random action for each of these d significant directions and sample these actions uniform randomly for Γ rounds. Since the algorithm explores first and then uses the estimate to compute the PE-Optimal design, it can be viewed as an explore-then-commit algorithm (Rusmevichientong and Tsitsiklis, 2010; Lattimore and Szepesvári, 2020a). As we consider a structured setting

[2]Here $\widetilde{O}$ hides logarithmic and problem dependent factors like $\sigma^2_{\min}, \kappa^2, H^2_U$.

we call this algorithm **S**tructured **P**olicy **E**valuation **E**xperimental **D**esign (SPEED). After $\Gamma = \sqrt{n}$ rounds, SPEED estimates the covariance matrix $\widehat{\mathbf{\Sigma}}_\Gamma$ as follows:

$$\widehat{\mathbf{\Sigma}}_\Gamma = \min_{\mathbf{S} \in \mathbb{R}^{d \times d}} \sum_{t=1}^{\Gamma} \left[\langle \mathbf{x}(a_t)\mathbf{x}(a_t)^\top, \mathbf{S} \rangle - (r_t - \mathbf{x}(a_t)^\top \widehat{\theta}_\Gamma)^2 \right]^2 \tag{3.5}$$

where $\widehat{\theta}_\Gamma$ is the ordinary least square (OLS) estimate of θ_* using the data from the first Γ rounds. Note that the OLS estimate is given by $\widehat{\theta}_\Gamma = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$, where $\mathbf{X} = \left(\mathbf{x}_1^\top, \cdots, \mathbf{x}_\Gamma^\top\right)^\top$ and $\mathbf{Y} = [r_1, \ldots, r_\Gamma]^\top$. A covariance estimation technique similar to (3.5) has been considered for the active regression setting though only for the case when $\mathbf{\Sigma}_*$ has rank 1 (Chaudhuri et al., 2017). The estimate of the covariance matrix $\widehat{\mathbf{\Sigma}}_\Gamma$ is then fed to the oracle optimizer (Assumption 5) to compute the sampling proportion $\widehat{b}$. Actions are chosen according to $\widehat{b}$ for the remaining $n - \Gamma$ rounds and then the WLS estimate $\widehat{\theta}_{n-\Gamma}$ is computed using $\widehat{\mathbf{\Sigma}}_\Gamma$ as the covariance matrix parameter (Equation (3.1)). Finally, SPEED outputs the dataset $\mathcal{D}$ to estimate the value of target policy π and $\widehat{\theta}_{n-\Gamma}$. Full pseudocode is given in Algorithm 2.

Algorithm 2 Structured Policy Evaluation Experimental Design (SPEED)

1: **Input:** Action set $\mathcal{A}$, target policy π, budget n.
2: Conduct forced exploration for $\Gamma = \sqrt{n}$ rounds and estimate $\widehat{\mathbf{\Sigma}}_\Gamma$ using (3.5).
3: Let $\widehat{b} \in \triangle(\mathcal{A})$ be the minimizer of $\mathcal{L}_n(\pi, \mathbf{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$.
4: Pull each action a exactly $T_n(a) = \left\lfloor \widehat{b}(a)(n - \Gamma) \right\rfloor$ times, and let $\mathcal{H}(a) \coloneqq \{a, R_i(a)\}_{i=1}^{T_n(a)}$ be the corresponding data. Set $\mathcal{D} \leftarrow \cup_a \mathcal{H}(a)$.
5: Construct the weighted least squares estimator $\widehat{\theta}_{n-\Gamma}$ using only the observations $\mathcal{D}$ from step 4.
6: **Output:** $\mathcal{D}$ and $\widehat{\theta}_{n-\Gamma}$.

Regret Analysis of SPEED

In this section, we first state our regret definition and then analyze the regret of the agnostic algorithm SPEED. As an agnostic algorithm, SPEED does not know the true covariance matrix $\mathbf{\Sigma}_*$ and must estimate the covariance matrix $\widehat{\mathbf{\Sigma}}_\Gamma$ after conducting exploration for Γ rounds. We define the loss of an algorithm after exploring for Γ rounds as the MSE of the resulting value estimate as follows:

$$\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) := \mathbb{E}_{\mathcal{D}}\Big[\Big(\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_{n-\Gamma} - \theta_*)\Big)^2\Big], \tag{3.6}$$

where $\widehat{\theta}_{n-\Gamma}$ is the WLS estimate of θ_* calculated from data of last $n-\Gamma$ rounds. We now define the regret for the agnostic algorithm with the estimated behavior policy $\widehat{b}$ as

$$\mathcal{R}_n = \overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*). \tag{3.7}$$

where $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$ is the loss of the agnostic algorithm and $\mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)$ is the oracle loss defined in (3.4). We now state the main theorem for the regret of SPEED.

Theorem 1. (***Regret of Algorithm 2, informal***) *Running Algorithm 2 with budget* $n \geqslant O_{\kappa^2, H_U^2}\big(\frac{d^4 \sigma_{max}^4 \log^2(A/\delta)}{\sigma_{min}^4}\big)$*, the resulting regret satisfies*

$$\mathcal{R}_n = O_{\kappa^2, H_U^2}\left(\frac{d^3 \sigma_{max}^2 \log(n)}{\sigma_{min}^2 n^{3/2}}\right).$$

Discussion of Regret: Theorem 1 states that the regret of Algorithm 2 scales as $O_{\kappa^2, H_U^2}(d^3 \sigma_{max}^2 \log(n)/n^{3/2})$ where d is the dimension of θ_*. Note that our regret bound depends on the underlying feature dimension d instead of actions A, and scales as $\widetilde{O}(d^3 n^{-3/2})$ which gives a positive answer to the main question of whether such a result is possible. In

comparison to earlier work, when $d^3 < A$, we have a tighter bound than that given by Carpentier and Munos (2011). Furthermore, the results of Carpentier and Munos (2011, 2012); Carpentier et al. (2015) are for the standard multi-armed bandit setting and cannot be easily extended to incorporate structure in the linear bandit setting. Our new bound also improves upon the A-optimal design method given by Fontaine et al. (2021), as their regret depends on the number of actions A and scales as $O(\frac{A \log n}{n^{3/2}})$.

Proof (Overview) of Theorem 1: We now outline the key steps for proving Theorem 1.

Step 1 (Regret Decomposition): We first decompose the regret $\mathcal{R}_n = \overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}^*_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)$. Recall that $\mathbf{b}_* \in \triangle(\mathcal{A})$ is the optimal design in (3.3) and $\widehat{b} \in \triangle(\mathcal{A})$ is the design followed by SPEED. However, we cannot directly go after the loss $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$ as it does not admit a simple structure like $\mathcal{L}^*_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)$. Rather we establish an upper bound on the loss $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$, given by $\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)$ (defined formally in (3.9)). Consequently, we can decompose the regret $\mathcal{R}_n$ into three parts as follows:

$$\begin{aligned}\mathcal{R}_n \overset{(a)}{\leqslant} & \underbrace{\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)}_{\textbf{Approximation error}} \\ & + \underbrace{\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)}_{\textbf{Comparing two different loss}} \\ & + \underbrace{\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}^*_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)}_{\textbf{Estimation error of } \mathbf{\Sigma}_*}. \end{aligned} \tag{3.8}$$

where (a) follows as we show that

$$\begin{aligned}\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) &= \mathbb{E}\left[\Big(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_{n-\Gamma} - \theta_*)\Big)^2\right] \\ &\leqslant \tfrac{1}{n-\Gamma}\left(1 + \tfrac{2Cd^2\log(A/\delta)}{\sigma_{\min}^2\Gamma}\right)\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a') \\ &\coloneqq \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma), \end{aligned} \tag{3.9}$$

where $C > 0$ is a constant. Note that the inequality above is shown in Proposition 6 which we discuss in depth in step 2. Finally note that $\widehat{b}_*$ is the empirical PE-Optimal design returned by the approximator after it is supplied with $\widehat{\mathbf{\Sigma}}_\Gamma$.

Step 2 (Bounding the loss $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$): In this step we discuss how to upper bound the agnostic loss $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$ with $\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)$ as defined in (3.9).

We first state a concentration lemma that is key to proving this upper bound. This lemma is novel for our proof because we estimate the underlying covariance matrix $\mathbf{\Sigma}_*$ using OLS estimator for Γ rounds. We then use the estimation $\widehat{\mathbf{\Sigma}}_\Gamma$ in the WLS estimator. For our lemma, we first define the variance concentration good event under Γ rounds of forced exploration as:

$$\xi_\delta^{var}(\Gamma) \coloneqq \left\{\forall a, \left|\mathbf{x}(a)^\top(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\mathbf{x}(a)\right| < \frac{2Cd^2\sigma_{\max}^2\log(A/\delta)}{\Gamma}\right\} \tag{3.10}$$

Lemma 3.1. (***OLS-WLS Concentration Lemma***) *After Γ samples of exploration, we can show that $\mathbb{P}(\xi_\delta^{var}(\Gamma)) \geqslant 1 - 8\delta$, where $C > 0$ is a constant.*

Proof (Overview) of Theorem 3.1: Note that we construct an initial estimate $\widehat{\theta}_\Gamma$ of θ_* using OLS estimate based on the first Γ rounds of data $\{a_t, r_t\}_{t=1}^\Gamma$. Let the feature of a_t be $\mathbf{x}_t$ and the squared residual $y_t \coloneqq (\mathbf{x}_t^\top\widehat{\theta}_\Gamma - r_t)^2$. Recall that SPEED estimates $\mathbf{\Sigma}_*$ via $\min_{\mathbf{S}\in\mathbb{R}^{d\times d}}\sum_{t=1}^\Gamma(\langle \mathbf{x}_t\mathbf{x}_t^\top, \mathbf{S}\rangle -$

$y_t)^2$. Let $\zeta_\Gamma \coloneqq \widehat{\theta}_\Gamma - \theta_*$. Then we can show that $y_t = \mathbf{x}_t^\top \mathbf{\Sigma}_* \mathbf{x}_t + \epsilon_t$ and the noise ϵ_t can be bounded by

$$\epsilon_t = \underbrace{\eta_t^2 - \mathbb{E}[\eta_t^2]}_{\textbf{Part A}} + \underbrace{2\eta_t \mathbf{x}_t^\top \zeta_\Gamma}_{\textbf{Part B}} + \underbrace{\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2}_{\textbf{Part C}}.$$

For the part A, observe that η_t^2 is a sub-exponential random variable as $\eta_t \sim \mathcal{SG}(0, \mathbf{x}_t^\top \mathbf{\Sigma}_* \mathbf{x}_t)$. Hence we can use sub-exponential concentration inequality from Theorem B.2 (Chapter B) to bound it. For part C first recall that $\zeta_\Gamma \coloneqq \widehat{\theta}_\Gamma - \theta_*$ and we use Theorem B.3 (Chapter B) to bound it. Finally, for part B, we can decompose $2\eta_t \mathbf{x}_t^\top \zeta_\Gamma \leqslant 2\eta_t^2 + \frac{1}{2}(\mathbf{x}_t^\top \zeta_\Gamma)^2$. Then using the same technique for parts A and C we bound the total deviation for part B. Combining the three parts gives the desired concentration inequality. The proof is in Section B.1. ■

Theorem 3.1 directly leads to Theorem B.11 (Section B.1) which shows that for $n \geqslant 16C^2 d^4 \log^2(A/\delta)/\sigma_{\min}^4$ we have that $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) \leqslant \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$. Compared to earlier work, Fontaine et al. (2021) does not require this approach as the variances of each action lack a common structure. Similarly, this approach differs from the time-dependent variance model of Zhang et al. (2021); Zhao et al. (2022).

Step 3 (Bounding the approximation error and comparing two different losses): For the approximation error in (3.8) we need access to an optimization oracle that gives ϵ approximation error (Assumption 5). Then setting $\epsilon = \frac{1}{\sqrt{n}}$ we have that the estimation error is upper bounded by $n^{-3/2}$. For comparing the two different losses in (3.8), we use their definition of to bound it as $O_{\kappa^2, H_u^2}(\frac{d^2 \log(A/\delta)}{n^{3/2}})$ as shown in (B.22) in Section B.1.

Step 4 (Bounding Estimation Error): Now observe that the third quantity in (3.8) (estimation error of $\mathbf{\Sigma}_*$) contains $\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)$ that depends on the design matrix $\mathbf{A}^{-1}_{\mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma}$ which in turn depends on the estimation of $\widehat{\mathbf{\Sigma}}_\Gamma$. Similarly $\mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)$ in the third quantity depends on the design matrix

$\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}$ which in turn depends on the true $\boldsymbol{\Sigma}_*$. Hence, we now bound the concentration of the loss under $\mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma}$ against the design matrix $\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}$ in the following lemma.

Lemma 3.2. (***Concentration of the design matrix***) *Let $\widehat{\boldsymbol{\Sigma}}_\Gamma$ be the empirical estimate of $\boldsymbol{\Sigma}_*$, and $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$. For any arbitrary proportion $\mathbf{b}$, with probability at least $(1-\delta)$, we have the following:*

$$\left| \sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma} - \mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}) \mathbf{w}(a') \right| \leqslant \frac{2CB^* d^3 \sigma^2_{\max} \log(A/\delta)}{\Gamma},$$

where B^* is a problem-dependent quantity and $C > 0$ is a universal constant.

Proof (Overview) of Theorem 3.2: We can upper bound $|\sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma} - \mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}) \mathbf{w}(a')| \leqslant \|\mathbf{u}\| \underbrace{\left\| \mathbf{A}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*} - \mathbf{A}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma} \right\|}_{\Delta} \|\mathbf{v}\|$ where, $\|\mathbf{u}\| = \|\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*} \mathbf{w}\|$ and $\|\mathbf{v}\| = \|\mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma} \mathbf{w}\|$. First, observe that $\|\mathbf{u}\|$ is a problem-dependent quantity. Then to bound Δ we use the Theorem 3.1 on the concentration of $\widehat{\sigma}^2_\Gamma(a)$. Finally to bound $\|\mathbf{v}\|$ we need to bound $\widehat{\sigma}^2_\Gamma(a) \leqslant \sigma^2(a) + \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\Gamma}$ where $\widehat{\sigma}^2_\Gamma(a)$ is the empirical variance of $\sigma^2(a)$. Combining everything yields the desired result. The proof is in Section B.1 ■

One of our key technical contributions in Theorem 3.2 is to show that the difference between the two losses $\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\boldsymbol{\Sigma}}_\Gamma)$, and $\mathcal{L}_n(\pi, \mathbf{b}_*, \boldsymbol{\Sigma}_*)$ scales with d^3 instead of the number of actions A. In contrast to prior work, a similar loss concentration in Fontaine et al. (2021) scales with A. Now using Theorem 3.2, setting the exploration factor $\Gamma = \sqrt{n}$, and $\delta = \frac{1}{n}$ we can show that the estimation error is upper bounded by $\frac{B^* C d^3 \sigma^2_{\max} \log(n)}{\sigma^2_{\min} n^{3/2}} + \frac{d^2}{n^2} \mathbf{Tr}(\sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top)$. Combining steps 1 – 4 we have the regret of SPEED as $O_{\kappa^2, H^2_U}(\frac{B^* d^3 \sigma^2_{\max} \log(n)}{\sigma^2_{\min} n^{3/2}})$. The full proof of Theorem 1 is in Section B.1. ■.

Lower Bound

Theorem 1 upper bounds the regret of our agnostic algorithm SPEED compared to an oracle algorithm with knowledge of $\mathbf{\Sigma}_*$. To quantify the tightness of our upper bound, we now turn to the question of whether we can lower bound the regret for any behavior policy learning algorithm. For our final theoretical result, we consider a slightly different notion of regret: $\mathcal{R}'_n \coloneqq \mathcal{L}_n(\pi, \widehat{b}, \mathbf{\Sigma}_*) - \mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)$. This notion of regret captures how sub-optimal the estimated $\widehat{b}$ is compared to $\mathbf{b}_*$, *without* additional error incurred by using an estimate of $\mathbf{\Sigma}_*$ in the WLS estimator. We conjecture that $\mathcal{R}'_n$ is indeed a lower bound to $\mathcal{R}_n$ as we have established in Proposition 1 that the minimum variance estimator is the WLS estimator using $\mathbf{\Sigma}_*$. Intuitively, $\mathcal{L}_n(\pi, \widehat{b}, \mathbf{\Sigma}_*)$ is a lower bound to $\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma)$ as estimation error will likely increase when using $\widehat{\mathbf{\Sigma}}_\Gamma$ in place of $\mathbf{\Sigma}_*$ in the WLS estimator. We leave proving that $\mathcal{R}'_n$ is a lower bound to $\mathcal{R}_n$ to future work.

Theorem 2.(*Lower Bound*) *Let* $|\mathbf{\Theta}|{=}2^d$, $\theta_* \in \mathbf{\Theta}$. *Then any arbitrary* δ*-PAC policy following the design* $\mathbf{b} \in \triangle(\mathcal{A})$ *satisfies* $\mathcal{R}'_n{=}\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*){-}\mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \geqslant \Omega\left(\frac{d^2 \lambda_d(\mathbf{V}) \log(n)}{n^{3/2}}\right)$ *for the environment specified in* (B.26).

Proof (Overview:) The proof follows the change of measure argument (Lattimore and Szepesvári, 2020a) and we follow the proof technique of Huang et al. (2017); Mukherjee et al. (2022b). We reduce the policy evaluation problem to the hypothesis testing setting and state a worst-case environment as in (B.26). We then show that the regret of any δ-PAC algorithm against an oracle in this environment must scale as $\Omega(\log n / n^{3/2})$. The proof is given in Section B.2. ■

From the above result, the upper bound of SPEED regret $\mathcal{R}_n$ matches the lower bound of regret $\mathcal{R}'_n$ in n but suffers an additional factor of d.

3.6 Experiments

We now conduct numerical experiments to show that SPEED decreases MSE faster than other baselines. These experiments complement our theoretical analysis as they do not have the conditions on budget n required in Theorem 1. Thus, our experimental analysis will show that the theoretically motivated SPEED algorithm still provides benefit even outside of the sample regime considered in theory. As baselines, we compare against Onpolicy, Oracle , A-Optimal (Fontaine et al., 2021), and G-Optimal (Wan et al., 2022). The Onpolicy algorithm simply runs the target policy to collect data, whereas the Oracle (as discussed in Section 3.3) samples according to the optimal $\mathbf{b}_*$. Of existing optimal design methods, A-Optimal , and G-Optimal are the closest in relation to our work. We experiment with A-Optimal design because this criterion minimizes the average variance of the estimates of the regression coefficients and is most closely aligned with our goal. The work of Wan et al. (2022) considers data collection under safety constraints using Inverse Propensity Weighting. In our unconstrained policy evaluation setting their approach boils down to just G-optimal design. Further experimental details are in Section B.3.

Unit Ball: We perform this experiment on a set of 5 actions that are arranged in a unit ball in $\mathbb{R}^2$ to show that SPEED allocates proportion to the most informative action (weighted by their variance). Figure 3.1 (Top Left) shows that SPEED reduces the MSE faster than Onpolicy, G-Optimal , and A-Optimal . We also include Oracle in this setting to show how quickly SPEED converges to it. However, for settings based on real-life data, we do not have such oracles.

Movielens Dataset: Consider a startup that wants to recommend movies to users based on their ratings. They have access to a target policy and want to evaluate it on a limited informative dataset before deploying it for full public use. We use real-world Movielens 1M dataset (Lam and Herlocker, 2016) datasets for this experiment. We apply low-rank

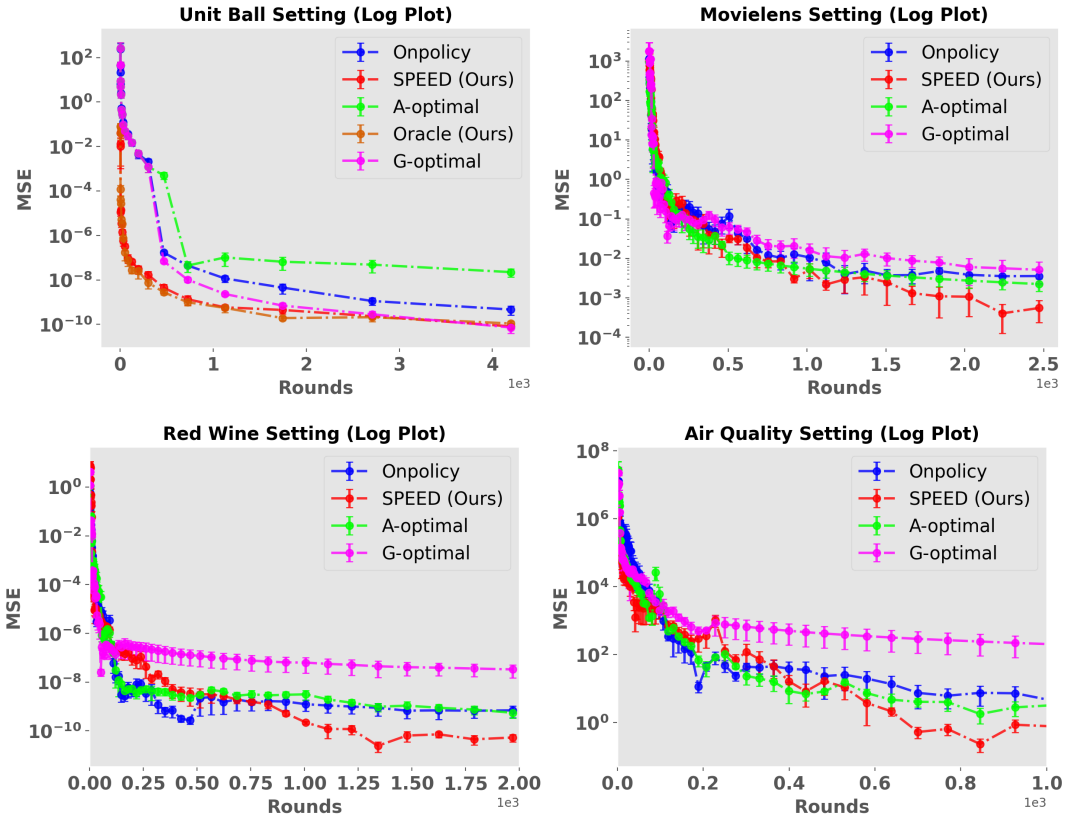


Figure 3.1: (Top-left) MSE plot for the Unit ball. (Top-right) MSE plot for the Movielens dataset. (Bottom-left) MSE plot for Red Wine Quality dataset. (Bottom-right) MSE plot for Air Quality dataset. The vertical axis gives MSE and the horizontal axis is the number of rounds. The vertical axis is log-scaled and confidence bars show one standard error.

factorization to the rating matrix to obtain 5-dimensional representations of users and movies. We then fit a weighted least square estimate of θ_* and Σ_*. We generate the reward using this θ_* and Σ_*. Then we use SPEED and other baselines to generate the small informative dataset to evaluate the target policy and this experiment is shown in Figure 3.1 (Top Right). SPEED initially conducts forced exploration to estimate θ_*, Σ_* and incurs slightly higher MSE but the MSE decreases faster than other baselines as the number of rounds increases.

Red Wine Quality: Consider an online wine company that wants to recommend wines to users and wants to evaluate a target policy before full deployment. We perform this experiment on real-world dataset *Red Wine Quality* from UCI datasets (Cortez et al., 2009). The dataset consists of 1600 samples (actions) of red wine with each sample a having feature $\mathbf{x}(a) \in \mathbb{R}^{11}$ and their ratings. We fit a weighted least square estimate to the original dataset and get an estimate of θ_* and $\mathbf{\Sigma}_*$. Then we use SPEED to generate the informative dataset to evaluate the target policy. Figure B.1 (Bottom-left) shows SPEED outperforming other baselines as horizon increases.

Air Quality: We now consider a setting where a government agency wants to record air quality and notify the public. However, it wants to evaluate a target policy on a limited informative dataset before full deployment. We perform this experiment on real-world dataset *Air-Quality* from UCI datasets (De Vito et al., 2008). The dataset consists of 1500 samples (actions) with each sample a having feature $\mathbf{x}(a) \in \mathbb{R}^6$ and their air quality value. Similar to red wine dataset we estimate of θ_* and $\mathbf{\Sigma}_*$. Then we use SPEED and other baselines (which do not know θ_* and $\mathbf{\Sigma}_*$) to generate the informative dataset to evaluate the target policy and this experiment is shown in Figure 3.1 (Bottom-right). Observe that SPEED's MSE decreases faster than other baselines as the number of rounds increases.

3.7 Conclusions and Future Directions

We proposed SPEED for optimal data collection for policy evaluation in linear bandits with heteroscedastic reward noise. We formulated a novel optimal design problem, PE-Optimal design, for which the optimal behavior policy is the solution that will produce minimal MSE policy evaluation when using a weighted least square estimate of the hidden reward parameters θ_* and $\mathbf{\Sigma}_*$. We showed the regret of SPEED degrades at

the rate of $\widetilde{O}(d^3 n^{-3/2})$ and matches the lower bound of $\widetilde{O}(d^2 n^{-3/2})$ except a factor of d. In contrast the Onpolicy suffers a regret of $\widetilde{O}(n^{-1})$ (Carpentier et al., 2015). We showed empirically that our design outperforms other optimal designs. In future work, we intend to extend the result to a more general class of hard problems such as collecting data to minimize the MSE of multiple target policies.

4 SAVER: OPTIMAL DATA COLLECTION STRATEGY FOR SAFE POLICY EVALUATION IN TABULAR MDPS

4.1 Introduction

Reinforcement learning has emerged as a powerful tool for decision-making in a wide range of applications, from robotics (Ibarz et al., 2021; Agarwal et al., 2022) and game-playing (Szita, 2012) to autonomous driving (Kiran et al., 2021), web-marketing (Bottou et al., 2013), healthcare (Fischer, 2018; Yu et al., 2019) and finance (Hambly et al., 2021). However, in these applications, it is often necessary to first evaluate the decision-making policy before its long-term deployment in the real world. In fact, policy evaluation is a critical step in reinforcement learning, as it allows us to assess the quality of a learned policy and to check whether it can truly achieve the desired goal for the target task. One potential solution to this issue is off-policy evaluation (OPE) (Dudík et al., 2014; Li et al., 2015; Swaminathan et al., 2017; Wang et al., 2017; Su et al., 2020; Kallus et al., 2021; Cai et al., 2021). However, for OPE estimators there is no control over how the static dataset is generated, which could result in low accuracy estimates.

Hence, a natural idea is to actively gather the dataset using an adaptive behavior policy and thus increase accuracy in the evaluation of the target policy's value. In many real-world settings, the behavior policy itself must satisfy some side constraints (specific to the industry) (Wu et al., 2016) or safety constraints (Wan et al., 2022) while collecting the dataset. For instance, in web marketing, it is common to run an A/B test with safety constraints over a subset of all users before a potential new policy is deployed for all users (Kohavi and Longbotham, 2017; Tucker and Joachims, 2022b). While testing autonomous vehicles it is quite natural to incorporate safety constraints in the behavior policy. So it is of great

practical importance to ensure that our data collection rule is safe (Zhu and Kveton, 2022b).

In this paper, we consider the question of optimal data collection for policy evaluation under safety constraints in the tabular reinforcement learning (RL) setting. Consider the following scenario that could arise in web marketing. Suppose we have a policy learned from offline data that has never been run in a real application. Moreover, we want this learned policy to be at least as good as a baseline policy that is already deployed in the application (Wu et al., 2016; Zhu and Kveton, 2021, 2022b). Off-policy evaluation often has high variance, so engineers may want to have some controlled deployment where the learned policy only makes decisions for some users before letting the policy make decisions for all users. We are motivated by how to make this controlled deployment as data-efficient and safe as possible. By *safe*, we mean that we want the expected return seen during data collection to remain close to the expected return under the baseline policy. A similar motivation can be found in Tucker and Joachims (2022b). In this paper, we focus on finding a behavior policy that produces a minimal variance estimate while remaining safe. We can state this formally as follows: We are given a target policy, π, for which we want to estimate its value denoted by $V^{\pi}(s_1)$, where s_1 is an arbitrary start state. To estimate $V^{\pi}(s_1)$ we will generate a set of K episodes where each episodic interaction ends after L time steps. We denote the total available budget of samples as $n = KL$. Each episode is generated by following some behavior policy and collect the dataset $\mathcal{D}$. Let $Y^{\pi}_{n}(s_1)$ be the estimate of $V^{\pi}(s_1)$ computed from $\mathcal{D}$. Then our objective is to determine a sequence of behavior policies that minimizes error in the estimation of $V^{\pi}(s_1)$ defined as $\mathbb{E}_{\mathcal{D}}[(Y^{\pi}_{n}(s_1) - V^{\pi}(s_1))^2]$ subject to a *safety constraint* on the cost-value of the behavior policies (to be defined later) that must hold with high probability.

There is a growing body of literature studying this important problem

of data collection for policy evaluation in both constrained and unconstrained setups. The work of Antos et al. (2008); Carpentier and Munos (2011, 2012); Carpentier et al. (2015); Fontaine et al. (2021); Mukherjee et al. (2022a, 2024g) studies this problem in the bandit setting without any constraints under the finite sample regime. A common metric of performance that these works consider is the difference between the loss of the agnostic algorithm that does not know problem-dependent parameters, and the oracle loss (which has access to problem-dependent parameters). This metric is termed *regret* and these works show that in the bandit setting the regret of the agnostic algorithm scales as $\widetilde{O}(n^{-3/2})$ where $\widetilde{O}(\cdot)$ hides log factors. One might be tempted to just run the target policy π, build $\mathcal{D}$ and then estimate $Y_n^{\pi}(s_1)$. This is called *on-policy* data collection. However, these works show that the on-policy regret degrades at a much slower rate of $\widetilde{O}(n^{-1})$ compared to active agnostic algorithms. Hence, a natural question arises, can we achieve similar performance for policy evaluation in the MDP setup under a finite sample regime even when we must conform to safety constraints? Thus, the goal of our work is to answer the following questions:

> 1) *Is there a class of MDPs where it is possible to incur a regret that degrades at a faster rate than* $\widetilde{O}(n^{-1})$*?* while satisfying safety constraints?
>
> 2) *If the answer is yes to* (1)*, can we design an adaptive algorithm* (*for this class of MDPs*) *to collect data for policy evaluation that does not violate the safety constraints* (*in expectation*)*, and its regret degrades at a faster rate than* $\widetilde{O}(n^{-1})$*?*

In this paper, we answer these questions affirmatively. Regarding the first question, we state the tractability condition on the class of MDPs which enables the optimal behavior policy to gather data for policy evaluation

without violating the safety constraint and suffer a regret of $\widetilde{O}(n^{-3/2})$. This condition leads to the first lower bound for this setting.

We also note that safe data collection for policy evaluation has also been studied in the bandit setting in Zhu and Kveton (2021, 2022b). However, we are the first to provide finite-time regret guarantees when per-step constraints must be maintained in expectation. We also show that in the bandit setup, our method empirically outperforms the adaptive importance sampling based algorithms in these works. Our formulation is also related to constrained MDPs though we specify that the constraint must be satisfied *throughout learning* and not just by the final policy (Efroni et al., 2020; Vaswani et al., 2022). We discuss further related works in Section 4.3.

Our main contributions are as follows:

(1) We formulate the problem of safe data collection for policy evaluation. We introduce the safety constraint such that at the end of n trajectories, the cumulative cost is above a constant factor of the baseline cost. To our knowledge, this is the first work to study this setting under such a safety constraint in the MDP setup with the goal of minimizing the estimate of the MSE of the target policy's expected reward.

(2) We then show that even in the special case of finite tree-structured MDPs the safe data collection for policy evaluation can be intractable. Then we come up with a condition on MDPs that enables any behavior policy to collect data without violating safety constraints. We also provide the first regret lower bound for the bandit and Tree MDP setting and show that it scales with $\Omega(n^{-3/2})$.

(3) We then consider an oracle strategy that knows the reward variances (problem-dependent parameter) of the reward distributions and derives its sampling strategy. We then introduce the agnostic algorithm **Sa**fe **V**arianc**e** **R**eduction (SaVeR) that does not know the problem-dependent parameters and show that its regret scales as $\widetilde{O}(n^{-3/2})$. We evaluate its performance against other baseline approaches and show that SaVeR reduces

MSE faster while satisfying the safety constraint.

4.2 Preliminaries

We consider the standard finite-horizon Markov Decision process, $\mathbf{M}$, with both a reward and constraint function. Formally, $\mathbf{M}$, is a tuple $(\mathcal{S}, \mathcal{A}, P, R, C, \gamma, d_0, L)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a state transition function, R is the reward function (formalized below), C is the constraint function (formalized below), $\gamma \in [0, 1)$ is the discount factor, d_0 is the starting state distribution, and L is the maximum episode length. A (stationary) policy, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, is a probability distribution over actions conditioned on a given state. We assume data can only be collected through episodic interaction: an agent begins in state $s_1 \sim d_0$ and then at each step t takes an action $a_t \sim \pi(\cdot|s_t)$ and proceeds to state $s_{t+1} \sim P(\cdot|s_t, a_t)$.

When the agent takes an action, a, in state, s, it receives both a reward $R \sim R(s, a)$ and a constraint value $C \sim C(s, a)$. We assume the transition model P is known but the reward distributions and constraint values are unknown. We define the reward value of a policy as: $V^\pi(s_1) \coloneqq \mathbb{E}_\pi[\sum_{t=1}^{n} \gamma^{t-1} R_t]$, where $\mathbb{E}_\pi$ is the expectation w.r.t. trajectories sampled by following π from the initial state s_1. We define a constraint-value of π similarly: $V_c^\pi(s_1) \coloneqq \mathbb{E}_\pi[\sum_{t=1}^{n} \gamma^{t-1} C_t]$. For simplicity, let the initial state distribution has probability mass on a single state s_1.

Our goal is to efficiently estimate $V^\pi(s_1)$ for a given policy π and this estimation requires data from the environment MDP. Past work has approached this problem by designing a sequence of *behavior policies* which are ran to produce informative data for evaluating π. However, in practical applications, it is often infeasible to simply run *any* behavior policy as doing so may violate domain constraints. We formalize this constraint by first assuming the existence of a safe *baseline policy*, π_0 that provides an ac-

ceptable constraint-value $V_c^{\pi_0}(s_1)$. Our objective is to determine a sequence of behavior policies, $\{\mathbf{b}_1, .., \mathbf{b}_K\}$, that will produce a set of K episodes that lead to the most accurate estimate of $V^\pi(s_1)$ subject to the constraint that the cumulative expected constraint-value $V_c^{\mathbf{b}}(s_1)$ always exceeds a fixed percentage of $V_c^{\pi_0}(s_1)$. We consider the objective:

$$\min_{\mathbf{b}} \mathbb{E}_{\mathcal{D}}[(Y_n^\pi(s_1) - V^\pi(s_1))^2] \tag{4.1}$$
$$\text{s.t.} \sum_{k'=1}^{k} V_c^{\mathbf{b}^{k'}}(s_1) \geqslant (1-\alpha) k V_c^{\pi_0}(s_1) \text{ for all } k \in [K]$$

where $Y_n(s_1)$ is our estimate of $V^\pi(s_1)$, $\alpha \in (0, 1]$ is the risk parameter, and the expectation is over the collected data set $\mathcal{D}$. We also make the following simplifying assumption. We assume π_0 is deterministic, i.e., will only select one action in any given state. W.l.o.g., we give this action the index 0 and refer to it as the *safe action*. The entire action set is $\mathcal{A} = \{0, 1, \ldots, A\}$. This assumption is reasonable in applications where existing, safe policies were created through non-learning methods or manually designed.

For analysis, we will estimate $V^\pi(s_1)$ with a certainty-equivalence estimator. We define the random variable representing the estimated future reward from state s at time-step ℓ as $Y_n^\pi(s, \ell) \coloneqq \sum_a \pi(a|s)\widehat{\mu}_n(s, a) + \gamma \sum_{s'} \widehat{P}_n(s'|s, a) Y_n^\pi(s', \ell+1)$ where $Y_n^\pi(s, \ell+1) \coloneqq 0$ if $\ell \geqslant L$, and $\widehat{\mu}_n(s, a)$ is an estimate of $\mu(s, a)$, both computed from $\mathcal{D}$. Finally, the estimate of $V^\pi(s_1)$ is computed as $Y_n^\pi(s_1) \coloneqq \sum_s d_0(s_1) Y_n^\pi(s_1, 0)$. Note that the total available budget of samples is n. We assume that there are K episodes and each episodic interaction terminates in at most L steps which implies $n = KL$.

We assume $V_c^{\mathbf{b}}(s_1)$ is known for $\mathbf{b} = \pi_0$ but not for any other policy. The constraint in (4.1) implies that the total constraint value over all deployed behavior policies should be above the total constraint value that can be obtained from the baseline policy π_0 till episode k with high probability. Observe that small values of α force the learner to be highly conservative,

whereas larger α values correspond to a weaker constraint. A similar setting has been studied for policy improvement by Wu et al. (2016); Yang et al. (2021b) for a variety of sequential decision-making settings. However, our objective is policy evaluation and we formulate a more general safety constraint in terms of $C(\cdot)$ while these prior works define the constraint in terms of $R(\cdot)$.

Similar to the recent works of Chowdhury et al. (2021); Ouhamma et al. (2023); Agarwal et al. (2019); Lattimore and Szepesvári (2020a) we assume the reward function $R(s,a) = \mathcal{N}(\mu(s,a), \sigma^2(s,a))$, where $\mathcal{N}$ denotes a Gaussian distribution with mean $\mu(s,a)$ and variance $\sigma^2(s,a)$. Similarly we assume a constraint function $C(s,a) = \mathcal{N}(\mu^c(s,a), \sigma^{c,(2)}(s,a))$, where $\mu^c(s,a)$ and $\sigma^{c,(2)}(s,a)$ are the mean and variance of $\mathcal{N}(\cdot)$. Note that this sub-Gaussian distribution assumption is required only for theoretical analysis, whereas our algorithm works for any bounded reward and cost functions. We assume that we have bounded reward and constraint mean $\mu(s,a), \mu^c(s,a) \in [0,\eta]$ respectively. Finally, we define the MSE of a behavior policy $\mathbf{b}$ for the target policy π at the end of budget n as

$$\mathcal{L}_n(\pi, \mathbf{b}) = \mathbb{E}_{\mathcal{D}}[(Y_n^\pi(s_1) - V^\pi(s_1))^2] \tag{4.2}$$

where the expectation is over dataset $\mathcal{D}$ which is collected by $\mathbf{b}$. Our main objective is to minimize the cumulative regret $\mathcal{R}_n$ subject to the safety constraint defined in (4.1). To define $\mathcal{R}_n$ we first define the MSE of a safe oracle behavior policy $\mathbf{b}_*^k$ that collects the dataset $\mathcal{D}$ as $\mathcal{L}_n^*(\pi, \mathbf{b}_*^k)$. We will formally describe such oracle policies in Section 4.4. Then the regret $\mathcal{R}_n$ is defined as

$$\mathcal{R}_n = \mathcal{L}_n(\pi, \mathbf{b}) - \mathcal{L}_n^*(\pi, \mathbf{b}_*^k). \tag{4.3}$$

4.3 Related Works

Our work lies at the intersection of two areas: 1) optimal data collection for policy evaluation, and 2) safe sequential decision-making. Optimal data collection for policy evaluation has been studied in reinforcement learning (Antos et al., 2008; Carpentier and Munos, 2012, 2011; Carpentier et al., 2015; Hanna et al., 2017a; Mukherjee et al., 2022a; Riquelme et al., 2017; Fontaine et al., 2021; Mukherjee et al., 2024g; Zhong et al., 2022b) without considering the safety constraints. In the bandit setting the optimal data collection has been studied in the context of estimating a weighted sum of the mean reward associated with each arm. Antos et al. (2008) study estimating the mean reward of each arm equally well and show that the optimal solution is to pull each arm proportional to the variance of its reward distribution. Since the variances are unknown a priori, they introduce an algorithm that pulls arms in proportion to the empirical variance of each reward distribution. A similar set of works by Carpentier and Munos (2012); Carpentier et al. (2015) extend the above work by introducing a weighting on each arm that is equivalent to the target policy action probabilities in our work. They show that the optimal solution is then to pull each arm proportional to the product of the standard deviation of the reward distribution and the arm weighting. The work of Riquelme et al. (2017); Fontaine et al. (2021); Mukherjee et al. (2024g) considers the linear bandit setting to study the policy evaluation setup where actions have different variances. Finally, Mukherjee et al. (2022a) study the policy evaluation setting for tabular MDP. However, these works only look into the policy evaluation setting without considering the safety constraint introduced in (4.1).

The safe sequential decision-making setup has recently attracted much attention in machine learning (Amodei et al., 2016; Turchetta et al., 2019) and reinforcement learning (Efroni et al., 2020; Wachi and Sui, 2020; Camilleri et al., 2022). In reinforcement learning, and specifically in the bandit

setting, safety has been studied in the context of policy improvement. In the bandit literature regret minimization under safety constraints has been studied in Wu et al. (2016); Kazerouni et al. (2017); Amani et al. (2019); Garcelon et al. (2020). In these works the safety requirements are encoded in the form of constraints on the cumulative rewards observed by the learner. These works refer to the setup as conservative bandits because exploration is limited by the constraints on the cumulative reward. The work of Wu et al. (2016) consider the setting of stochastic bandits for policy improvement with a safety constraint similar to (4.1). However, Kazerouni et al. (2017); Amani et al. (2019); Garcelon et al. (2020); Moradipari et al. (2021); Pacchiano et al. (2021); Hutchinson et al. (2024) study the linear bandit setting under safety constraints where the actions have features associated with them. Note that none of the above works study policy evaluation under safety constraints. Wan et al. (2022); Zhu and Kveton (2021, 2022b) analyzes off policy evaluation in the context of designing a non-adaptive policy using inverse probability weighting estimator (as opposed to designing an adaptive policy using certainty equivalence estimator in this work).

In the MDP setting the works of Efroni et al. (2020); Altman (2021); Wachi et al. (2024); Li et al. (2024a); Zheng et al. (2024); Xiong et al. (2024); Ding et al. (2024); Wang et al. (2024); Mazumdar et al. (2024) study different variations of the safe exploration in constraint MDPs in both offline and online policy improvement settings. The work of Yang et al. (2024) studies the safe policy improvement in constraint MDP setting under non-stationary policies. The work of Gupta et al. (2024) proposed a safe policy improvement approach for variable horizon setting such that the safe reinforcement learning agent uses a variable look-ahead horizon to avoid unsafe states. The constrained MDP problems have also been looked into from the lens of optimization where Chen et al. (2021b, 2022a); Qiu et al. (2020); Ding et al. (2020); Vaswani et al. (2022); Ding et al.

(2021); Liang et al. (2018); Ying et al. (2024) have proposed a primal-dual sampling-based algorithm to solve CMDPs for the policy improvement setting.

4.4 Intractability and Lower Bounds

In this section, we first define an oracle data collection strategy that ignores the constraints. We call this the unconstrained oracle. This oracle data collection algorithm can reach a regret bound of $\widetilde{O}(n^{-3/2})$ in the unconstrained setting (Carpentier and Munos, 2012; Carpentier et al., 2015; Mukherjee et al., 2022a). We then show how data collection for policy evaluation under safety constraints in MDPs is challenging compared to standard policy improvement challenges in constrained MDPs (Efroni et al., 2020; Vaswani et al., 2022) as well as safe data collection for policy evaluation in bandits (Zhu and Kveton, 2021; Wan et al., 2022; Zhu and Kveton, 2022b). To show this challenging aspect, we first discuss how the unconstrained oracle fails to satisfy the constraint and achieve the desired regret of $\widetilde{O}(n^{-3/2})$ in the constraint MDP setting. We then propose a safe variant of the oracle policy and finally, discuss a tractability condition that enables the safe oracle algorithm to achieve a regret bound of $\widetilde{O}(n^{-3/2})$.

Unconstrained Oracle

In this section, we discuss the unconstrained oracle data collection strategy that knows the variances of reward and constraint value but does not know the mean of either. Moreover, this oracle does not take into account the safety constraints in (4.1). For easier exposition of our results, we again state the learning procedure of the oracle. After observing n samples (state-action-reward tuples), the oracle computes the estimate of $V^{\pi}(s_1^1)$ as $Y_n^{\pi}(s_1^1) = \sum_{a=1}^{A} \pi(a|s_1^1)\big(\widehat{\mu}_n(s_1^1, a) + \sum_{s_j^{\ell+1}} P(s_j^2|s_1^1, a) Y_n(s_j^2)\big)$. Note that we defined $Y_n^{\pi}(s, \ell)$ before, but now we use $Y_n^{\pi}(s)$ and assume the time

step is implicit in the state for this finite-horizon MDP. Mukherjee et al. (2022a) shows that in the unconstrained setting, to reduce the $\mathrm{Var}(Y_n^\pi(s_1^1))$ the optimal sampling proportion of the oracle for any state s_i^ℓ is:

$$\mathbf{b}_*(a|s_i^\ell) \propto \Big(\pi^2(a|s_i^\ell)\big[\sigma^2(s_i^\ell, a) + \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) M^2(s_j^{\ell+1})\big]\Big)^{\frac{1}{2}} \tag{4.4}$$

where, $M(s_j^\ell)$ is the normalization factor defined as follows:

$$M(s_i^\ell) = \sum_a \Big(\pi^2(a|s_i^\ell)\big(\sigma^2(s_i^\ell, a) + \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) M^2(s_j^{\ell+1})\big)\Big)^{\frac{1}{2}}. \tag{4.5}$$

Observe from the definition of $\mathbf{b}_*(a|s_i^\ell)$ that the optimal proportion in the terminal states, i.e. $\mathbf{b}_*(a|s_j^L)$, do not affect subsequent states and only depends on the target probability $\pi^2(a|s_i^\ell)$ and variance $\sigma^2(s_i^\ell, a)$. The key difference is in the non-terminal states, s_i^{L-1}, where the optimal action proportion, $\mathbf{b}_*(a|s_i^{L-1})$ depends on the expected terminal state normalization factor $M(s_j^L)$ where s_j^L is a state sampled from $P(\cdot|s_i^{L-1}, a)$. The normalization factor, $M(s_j^L)$, captures the total contribution of state s_j^L to the variance of $Y_n^\pi(s_j^{L-1})$ and thus actions in the starting state must be chosen to 1) reduce variance in the immediate reward estimate and to 2) get to states that contribute more to the variance of the estimate. This observation is also noted in Mukherjee et al. (2022a). Finally, since $\mathbf{b}_*(a|s)$ also depends on $P(s'|s, a)$, it will put a low sampling proportion on actions a leading to such s' which has low transition probabilities.

Safe Oracle Algorithm for Safe Data Collection

The behavior policy defined in the previous section ignores the safety constraint and is thus inapplicable to our problem setting. In this section, we describe a safe variant of this oracle. We define a few notations before introducing the safe algorithm. Let $T_\ell^k(s, a) \coloneqq \sum_{k'=1}^{k-1} \sum_{\ell'=1}^{\ell-1} \mathbf{1}\{S_{\ell'}^{k'} =$

$s, A_{\ell'}^{k'} = a\}$ be the number of times (s, a) is visited before episode k. Let the mean reward estimate of (s, a) till episode k be computed as $\widehat{\mu}_{\ell}^{k}(s, a) \coloneqq (T_{\ell}^{k}(s, a))^{-1} \sum_{k'=1}^{k-1} \sum_{\ell'=1}^{\ell-1} \mathbf{1}\{S_{\ell'}^{k'} = s, A_{\ell'}^{k'} = a\} R_{\ell'}^{k'}$, where $R_{\ell'}^{k'}$ is the observed reward. Similarly define the constraint-values estimate $\widehat{\mu}_{c,\ell}^{k}(s, a)$ based on constraint value C_{ℓ}^{k}. Define the confidence interval at the time step L of k-th episode as $\beta_{L}^{k}(s, a) \coloneqq L\sqrt{\log(SAn(n+1))/T_{L}^{k}(s, a)}$ (Agarwal et al., 2019).

Let $Y_{c,L}^{\mathbf{b}^k}(s_1^1) = \sum_{a=1}^{A} \mathbf{b}^k(a|s_1^1)\big(\widehat{\mu}_{c,L}^{k}(s_1^1, a) + \sum_{s_j^{\ell+1}} P(s_j^2|s_1^1, a) Y_{c,L}^{\mathbf{b}^k}(s_j^2)\big)$ denote the empirical estimate of $V_{c}^{\mathbf{b}^k}(s_1^1)$ at the end of the k-th episode, and $\widehat{\mu}_{c,L}^{k}(s, a)$ is the empirical estimate of $\mu^c(s, a)$ at the end of the k-th episode. Note that the oracle algorithm knows the variances of reward $R(\cdot)$ and constraint-value $C(\cdot)$. Using this knowledge, it maintains a safety budget $\widehat{Z}_{L}^{k-1}$ where $\widehat{Z}_{L}^{k-1} \coloneqq \sum_{k'=1}^{k-1}(Y_{c,L}^{\mathbf{b}^{k'}}(s_1^1) - \beta_{L}^{k'}(s, a)) - (1-\alpha)(k-1)V_{c}^{\pi_0}(s_1^1)$ is the safety budget at the end the $k-1$-th episode. The $\underline{Y}_{c,L}^{\mathbf{b}^k}(s_1^1) = Y_{c,L}^{\mathbf{b}^k}(s_1^1) - \beta_{L}^{k}(s, a)$ is the lower confidence bound to the $Y_{c,L}^{\mathbf{b}^k}(s_1^1)$.

Exploration policy π_x: We require an exploration policy π_x as the oracle algorithm needs a good estimation of the constraint-value $\mu^c(s, a)$ and following the oracle proportion $\mathbf{b}_*(a|s)$ may not lead to a good estimation of $\mu^c(s, a)$. This exploration policy should ensure with high probability that the estimation error of $\mu^c(s, a)$ is low in each (s, a) for which $\pi(a|s) > 0$ and can be an optimal design based policy like PEDEL that explores the state space informatively (Wagenmaker and Jamieson, 2022) or other exploration policies (e.g., Dann et al. (2019); Ménard et al. (2020); Uehara et al. (2021)).

We now state the following safe oracle algorithm: At the k-th episode run the policy

$$\mathbf{b}_{*}^{k} = \begin{cases} \mathbf{b}_{*}, & \text{if } \widehat{Z}_{L}^{k-1} \geqslant 0, k > \sqrt{K} \\ \pi_0 & \text{if } \widehat{Z}_{L}^{k-1} < 0 \\ \pi_x, & \text{if } \widehat{Z}_{L}^{k-1} \geqslant 0, k \leqslant \sqrt{K} \end{cases}. \tag{4.6}$$

The safe oracle algorithm in (4.6) alternates between the optimal oracle policy $\mathbf{b}_*$ in (4.4) when the safety budget $\widehat{Z}_L^{k-1}$ at the start of the episode k is greater than 0, otherwise it falls back to running the baseline policy π_0. Additionally, the safe oracle conducts forced exploration for at most $\sqrt{K}$ episodes when $\widehat{Z}_L^{k-1} \geqslant 0$ using the exploration policy π_x to estimate $\mu^c(s,a)$. This is because following the oracle proportion $\mathbf{b}_*$ in (4.4) that samples high variance state-action tuples may not lead to a good estimate of $\mu^c(s,a)$.

An Intractable MDP

In this section, we now show that there exist MDPs where even a safe oracle algorithm may not be able to reach the desired $\widetilde{O}(n^{-3/2})$ regret bound. We then introduce the tractability condition which depends on the budget as the $\mathbf{b}_*$ needs to be run sufficient number of times to reach a regret of $\widetilde{O}(n^{-3/2})$. So a more benign MDP allows one to run $\mathbf{b}_*$ most of the time whereas a less benign MDP allows you to play $\mathbf{b}_*$ less. Hence tractability depends on the budget being sufficiently large and also depends on properties of the MDP and the risk parameter α. To show this challenging aspect of safe data collection, we first define a Tree MDP. Using Tree MDPs to analyze the hardness of learning in MDPs and deriving lower bounds is common in the literature (Jiang and Li, 2016; Weisz et al., 2021; Wagenmaker et al., 2022b; Jin et al., 2022). The tree MDP is defined as follows:

Definition 4.1. (***Tree MDP***) *An MDP is a discrete tree MDP* $\mathbf{T} \subset \mathbf{M}$ *in which:* (***1***) *There are* L *levels indexed by* ℓ *where* $\ell = 1, 2, \ldots, L$. (***2***) *Every state is represented as* s_i^ℓ *where* ℓ *is the level of the state* s *indexed by* i. (***3***) *The transition probabilities are such that one can only transition from a state in level* ℓ *to one in level* $\ell + 1$ *and each non-initial state can only be reached through one other state and only one action in that state. Formally,* $\forall s'$, $P(s'|s,a) \neq 0$

for only one state-action pair s, a *and if* s' *is in level* $\ell + 1$ *then* s *is in level* ℓ. *Finally,* $P(s_j^{L+1}|s_i^L, a) = 0, \forall a$. **(4)** *For simplicity, we assume that there is a single starting state* s_1^1 *(called the root). It is easy to extend our results to multiple starting states with a starting state distribution,* d_0*, by assuming that there is only one action available in the root that leads to each possible start state,* s*, with probability* $d_0(s)$. *The leaf states are denoted as* s_i^L. **(5)** *The interaction stops after* L *steps in state* s_i^L *after taking an action* a.

Proposition 1. *Fix an arbitrary* $n > 0$. *Then there exists an environment where no algorithm (including the safe oracle* $\mathbf{b}_*^k$*) can be run that will result in a regret* $\mathcal{R}_n = \mathcal{L}_n(\pi, \mathbf{b}_k^*) - \mathcal{L}_n^*(\pi, \mathbf{b}_*)$ *of* $\widetilde{O}(n^{-3/2})$ *while satisfying the safety constraint, where* $\mathbf{b}_*$ *is the unconstrained oracle.*

Proof (Overview) We first construct a worst-case 3 armed bandit environment (MDP with single state) such that $\mu^c(0) = 0.5$, $\mu^c(1) = 0.5 + \alpha$, $\mu^c(2) = 0$ and variance of $\sigma^{r,(2)}(0) = 0.001$, $\sigma^{r,(2)}(1) = 0.001$ and $\sigma^{r,(2)}(2) = 0.25$. So action $\{2\}$ has low constraint value (unsafe) but has high variance. So the safe oracle policy must sample the action 2 a large number of times to reach a regret of $\widetilde{O}(n^{-3/2})$. However, since action $\{2\}$ is unsafe, the safe oracle has to sample baseline action 0 a sufficient number of times to accrue some safety budget. Combining these two observations we show that achieving a regret rate of $\widetilde{O}(n^{-3/2})$ is impossible. The full proof is in Section C.1.

The key reason the above environment is intractable is that some trajectories taken by safe oracle has very less constraint value associated with them, compared to the trajectory taken by the baseline policy. To rule out such pathological MDPs, we define the *tractability* condition as follows: Let $\mathbf{b}^-$ be any behavior policy that minimizes $V_{\mathbf{b}}^c(s_1)$. Define $V_{\mathbf{b}^-}^c(s_1)$ as the value of the policy $\mathbf{b}^-$ starting from state s_1. This policy $\mathbf{b}^-$ suffers a value $V_{\mathbf{b}^-}^c(s_1)$ that is lower than any other behavior policy $\mathbf{b}$. So this policy $\mathbf{b}^-$ can be thought of as the worst possible behavior policy that can be followed by the agent during an episode. Then the tractability condition

states that

$$\sqrt{n} \geqslant \frac{\frac{1}{\alpha}\left(1 - \frac{V^c_{\mathbf{b}^-}(s_1)}{V^c_{\pi_0}(s_1)}\right)}{\frac{C_\sigma}{\alpha}\left(1 - \frac{V^c_{\mathbf{b}^-}(s_1)}{V^c_{\pi_0}(s_1)}\right) - 1} \tag{4.7}$$

where $C_\sigma \in (0,1)$ is a MDP dependent parameter that depends on the reward variance of state-action pairs such that $\frac{C_\sigma}{\alpha}\left(1 - \frac{V^c_{\mathbf{b}^-}(s_1)}{V^c_{\pi_0}(s_1)}\right) - 1 > 0$. The quantity $C_\sigma = \max_{s,a} \frac{\mathbf{b}_*(a|s)}{M(s)}$ where $\mathbf{b}_*(a|s)$ and $M(s)$ are defined in (4.4) and (4.5) respectively. So $C_\sigma \in (0,1)$ and it captures the worst case trajectory that can be followed by $\mathbf{b}_*$.

This condition in (4.7) gives us (1) the lower bound to the budget n to run the behavior policy $\mathbf{b}^-$ to achieve a regret bound of $\widetilde{O}(n^{-3/2})$ and satisfy the safety constraint; (2) $V^c_{\mathbf{b}^-}(s_1) < V^c_{\pi_0}(s_1)$ so that the RHS is positive, (3) depends on the reward variance of state action pairs in the MDP so that $\frac{C_\sigma}{\alpha}\left(1 - \frac{V^c_{\mathbf{b}^-}(s_1)}{V^c_{\pi_0}(s_1)}\right) - 1 > 0$, and (4) for smaller α (high risk) the R.H.S increases which increases the required budget n. We further discuss how this condition in (4.7) is derived in Theorem C.3. Then we define the following assumption.

Assumption 6. (***Tractability***) *We assume a sufficiently large budget* n *and an MDP* $\mathbf{M}$ *that satisfies the constraint in* (4.7)*. We call such an MDP* $\mathbf{M}$ *tractable.*

Assumption 6 ensures that even the worst possible behavior policy $\mathbf{b}^-$ that can reach a regret of $\widetilde{O}(n^{-3/2})$ has sufficient budget n to satisfy the safety constraint. Moving forward, we will define regret relative to this safe oracle $\mathbf{b}^K_*$ instead of the unconstrained oracle. Furthermore, we assume tractability in (6) such that the safe oracle decreases MSE at a comparable rate to the unconstrained oracle $\mathbf{b}_*$. Define the reward regret as $\mathcal{R}_n = \mathcal{L}_n(\pi, \mathbf{b}) - \mathcal{L}^*_n(\pi, \mathbf{b}^k_*)$ where $\mathcal{L}^*_n(\pi, \mathbf{b}^k_*)$ is the safe oracle MSE, and $\mathcal{L}_n(\pi, \mathbf{b})$ is the agnostic algorithm MSE that does not know reward or constraint-value variances. Now we present the first general lower bound

theorem for the safe data collection strategy in MDPs.

Theorem 1. (***Lower Bounds***) *Let* $\pi(a|s) = \frac{1}{A}$ *for each state* $s \in \mathcal{S}$. *Under Assumption 6 the regret* $\mathcal{R}_n = \mathcal{L}_n(\pi, \mathbf{b}) - \mathcal{L}_n^*(\pi, \mathbf{b}_*^k)$ *is lower bounded by*

$$\mathbb{E}\left[\mathcal{R}_n\right] \geqslant \begin{cases} \Omega\left(\max\left\{\frac{A^{1/3}}{n^{3/2}}, \left(\frac{H_{*,(1)}^2 A^{2/3}}{n^{3/2}}\right)\right\}\right), (\textbf{\textit{MAB}}) \\ \Omega\left(\max\left\{\frac{\sqrt{SAL^2}}{n^{3/2}}, \left(\frac{H_{*,(1)}^2 SAL^2}{n^{3/2}}\right)\right\}\right) (\textbf{\textit{MDP}}) \end{cases}$$

where, $\Delta_0 = V_c^{\mathbf{b}_*}(s_1^1) - V_c^{\pi_0}(s_1^1)$ *and* $H_{*,(1)} = \frac{1}{\alpha V_c^{\pi_0}(s_1^1)}(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0)$ *is the hardness parameter.*

Discussion: Theorem 1 shows that in the constrained setting the lower bound scales as $\Omega(H_{*,(1)}^2 n^{-3/2})$. Note that we can recover the lower bound for the unconstrained setting using this result. In the unconstrained bandit setting the bound scales as $O\left(A^{1/3} n^{-3/2}\right)$ which matches the lower bound of Carpentier and Munos (2012) (see their Theorem 5). We also establish the first lower bound for the unconstrained setting in data collection for policy evaluation in the tabular MDP setup that scales as $O\left(\sqrt{SAL^2} n^{-3/2}\right)$. The $H_{*,(1)}$ captures the hardness in learning in the MDP and consists of the gap Δ_0, $V_c^{\pi_0}(s_1^1)$ and α. Note that $H_{*,(1)}$ increases with α, and the Δ_0 captures how much constraint value the $\mathbf{b}_*$ can obtain compared to π_0. Finally, the smaller value of π_0 increases the hardness as the π_0 has to be run more times so that the safety constraint is not violated.

Proof (Overview) We first build two deterministic tree MDPs $\mathbf{T}$ and $\mathbf{T}'$ which differ in the variances at only one state. This leads to different optimal oracle behavior policies in $\mathbf{T}$ and $\mathbf{T}'$. Then using the divergence decomposition lemma for MDPs from Garivier and Kaufmann (2016); Wagenmaker et al. (2022b) we show in Theorem C.9 that in $\mathbf{T}$ the regret lower bound scales as $\Omega(\sqrt{SAL^2 \log(n)}/n^{3/2})$. Next, we follow a reduction-based proof technique to prove the reward regret lower bound in the constrained setting. Consider any sequential decision-making prob-

lem $\mathfrak{A}$ (for instance a multi-armed bandit problem, tabular RL) such that there exists a problem-dependent constant $\xi \in \mathbb{R}$ that only depends on on the number of actions in bandits, or state-action-horizon in tabular RL. Then for a large budget n and any algorithm we have from Theorem C.8 and Theorem C.9 that $\mathbb{E}[\mathcal{R}_n] \geqslant \frac{\xi}{n^{3/2}}$ for an MDP dependent parameter ξ. Then we lower bound how many times under the budget n the algorithm can run the baseline policy. This is lower bounded in step 2 as $\mathbb{E}[\mathcal{R}_n] \gtrsim \min\left\{\frac{\xi}{n^{3/2}}, \frac{(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)^2\xi^2}{(\alpha V_c^{\pi_0}(s_1^1))^2 n^{3/2}}\right\}$. We finish off the proof by noting that the quantity $H_{*,(1)} = \frac{1}{\alpha V_c^{\pi_0}(s_1^1)}(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0)$ is the hardness parameter when $\pi(a|s) = 1/A$, and substituting the value of $\xi = A^{1/3}$ for bandits (Theorem C.8) and $\xi = \sqrt{SAL^2}$ for $\mathbf{T}$ (Theorem C.9). Since $\mathbf{T} \subset \mathbf{M}$, this result is a lower bound to $\mathbf{M}$ as well. The full proof is in Section C.2. ■

4.5 Agnostic Algorithm for Safe Policy Evaluation

In this section, we introduce the more realistic agnostic algorithm that does not know the mean and variances of the reward and constraint values of the actions. We then analyze this algorithm and establish its finite-time MSE. We call this algorithm **Sa**fe **V**arianc**e** **R**eduction algorithm (abbreviated as SaVeR) as it reduces the variance of the estimated value of the target policy by following (4.4) while simultaneously satisfying the safety constraint (4.1) with high probability.

We introduce a few notations before presenting the algorithm. Define the upper confidence bound on the empirical reward variance as $\overline{\widehat{\sigma}}_L^k(s,a) := \widehat{\sigma}_L^k(s,a) + \beta_L^k(s,a)$, where $\beta_L^k(s,a)$ is the confidence interval defined in Section 4.4. We define the empirical sampling proportion for an arbitrary state-action (s_i^ℓ, a) as $\widehat{b}_\ell^k(a|s_i^\ell)$. Define the policy $\widehat{b}_{*,\ell}^k(a|s_i^\ell)$ as similar to $\mathbf{b}_*(a|s_i^\ell)$ defined in (4.4), but it uses plug-in estimate $\overline{\widehat{\sigma}}_\ell^k(s,a)$

instead of $\sigma_\ell^k(s,a)$. This is because the agnostic algorithm does not know the reward and constraint-value variances. We define $\widehat{Z}_L^{k-1}$ similar to (4.6). Finally, we define our algorithm, SaVeR, as follows: At episode k run the policy:

$$\widehat{b}^k = \begin{cases} \widehat{b}_*^k & \text{if } \widehat{Z}^{k-1} \geqslant 0, k > \sqrt{K} \\ \pi_0 & \text{if } \widehat{Z}^{k-1} < 0 \\ \pi_x & \text{if } \widehat{Z}^{k-1} \geqslant 0, k \leqslant \sqrt{K} \end{cases} \tag{4.8}$$

where $\widehat{b}_*^k$ for the episode k is defined as follows: For each time step $\ell = 1, 2, \ldots, L$ sample action $A_\ell^k = \arg\max_a \frac{\widehat{b}_*^k(a|s_j^\ell)}{T_\ell^k(s_j^\ell, a)}$, where $\widehat{b}_*^k(a|s_j^\ell)$ is the plug-in estimate of $\mathbf{b}_*(a|s_j^\ell)$ as defined in (4.4). SaVeR alternates between the exploration policy π_x, plugin optimal policy $\widehat{b}_*^k$, and baseline policy based on the safety budget $\widehat{Z}^k$ and the number of episodes K. In contrast to (4.8) the oracle policy in (4.6) uses the true oracle proportions $\mathbf{b}^*$ when $\widehat{Z}^{k-1} \geqslant 0, k > \sqrt{K}$. Also, observe that the action selection rule ensures that the ratio $\widehat{b}_{*,\ell}^k(a|s)/T_\ell^k(s,a) \approx 1$. It is a deterministic action selection rule and thus avoids inadvertently violating the safety constraint due to random sampling from the optimal proportions $\widehat{b}_\ell^k(a)$. Now we formally state the SaVeR for the tree MDP. At every episode $k \in [K]$ it generates a sampling history $\mathcal{H}^k := \{S_\ell^k, A_\ell^k, R(S_\ell^k, A_\ell^k), C(S_\ell^k, A_\ell^k)\}_{\ell=1}^L$ by selecting A_ℓ^k according to (4.8) and appends it to the dataset $\mathcal{D}$. After observing the feedback it updates the model parameters and estimates $\widehat{b}_1^{k+1}(a|s)$ for each s, a. It returns the dataset $\mathcal{D}$ to evaluate π. The pseudocode is in Algorithm 3.

We now present a theorem that gives the MSE of the agnostic algorithm SaVeR in the tree MDP in the following theorem. We define the problem complexity parameters $M = \sum_{\ell=1}^L \sum_{s_j^\ell} M(s_j^\ell)$ summed over all stated

Algorithm 3 Safe Variance Reduction (SaVeR) for T

Input: Risk Parameter $\alpha > 0$, target policy π.
Output: Dataset $\mathcal{D}$.
Initialize $\mathcal{D} = \emptyset$, $\widehat{b}_1(a|s)$ uniform over all actions.
for $k = 1, 2, \ldots, K$ **do**
  for $\ell = 1, 2, \ldots, L$ **do**
    Get $\mathcal{H}^k := \{S_\ell^k, A_\ell^k, R(S_\ell^k, A_\ell^k), C(S_\ell^k, A_\ell^k)\}_{\ell=1}^L$ by selecting $\mathbf{b}^k$ according to (4.8).
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathcal{H}^k, \widehat{b}^k)\}$
    Update model parameters and estimate $\widehat{b}_1^{k+1}(a|s)$ for each s, a
Return Dataset $\mathcal{D}$ to evaluate policy π.

$s \in [S]$. Define predicted agnostic constraint violation

$$\mathcal{C}_n(\pi, \widehat{b}^k) := \sum_{k=1}^{K} \mathbb{I}\{\widehat{Z}^k < 0\}$$

when taking actions according to (4.8). For scalars $x, y \in \mathbb{R}$ define $\min^+(x, y) := |\min(x, y)|$. Define the problem complexity parameter $H_{*,(2)} = \sum_{\ell=1}^{L} \sum_{s_j^\ell} H_{*,(2)}(s_j^\ell)$ where

$$H_{*,(2)}(s_j^\ell) = \frac{1}{\alpha \mu^c(s_j^\ell, 0)} \sum_{a \in \mathcal{A} \setminus \{0\}} \pi(a|s_j^\ell)\sigma(s_j^\ell, a) \overset{+}{\min} \{\Delta_c(s_j^\ell, a), \Delta_c(s_j^\ell, 0) - \Delta_c(s_j^\ell, a)\}. \tag{4.9}$$

Remark 4.2. *The quantity $H_{*,(2)}(s_j^\ell)$ signifies the total cost of maintaining the safety constraint at state s_j^ℓ by sampling action 0 instead of sampling based on $\pi(a)\sigma(a)$. Observe that $\Delta_c(s_j^\ell, 0) - \Delta_c(s_j^\ell, a) = \mu_c(s_j^\ell, a) - \mu_c(s_j^\ell, 0)$. So $\min^+\{\Delta_c(s_j^\ell, a), \Delta_c(s_j^\ell, 0) - \Delta_c(s_j^\ell, a)\}$ depends on how close is the action a to the best cost action $\mu^{*,c}(s_j^\ell)$ or the baseline action 0. Also observe that because of the $\min^+$ operator, this quantity cannot be 0. Further, observe that the gap is weighted by $\pi(a|s_j^\ell)\sigma(s_j^\ell, a)$ signifying that actions with low variance and target*

probability contribute less to the constraint violation MSE. Also, observe that higher risk setting ($\alpha \to 0$) *leads to higher* $H_{*,(2)}(s_j^\ell)$. *Finally, it can be easily verified that* $H_{*,(2)} > H_{*,(1)}$.

Now we present a theorem that we will use to bound the regret of SaVeR in Tree MDP $\mathbf{T}$ under Assumption 6.

Theorem 2. **(*informal*)** *The MSE of the SaVeR in* $\mathbf{T}$ *for* $\frac{n}{\log(SAn(n+1)/\delta)} \geqslant O((LSA^2)^2 + \frac{SA}{\Delta_{\min}^{c,(2)}} + \frac{1}{4H_{*,(2)}^2})$ *is bounded by* $\mathcal{L}_n(\pi, \widehat{b}^k) \leqslant \widetilde{O}\big(\frac{M^2(s_1^1)}{n} + \frac{M^2(s_1^1)}{n}(MLSA^2 + H_{*,(2)})^2 + \frac{(LSA^2)^2 H_{*,(2)}^2 M^2}{\min_s \mathbf{b}^{*,k,(3/2)}(s) n^{3/2}}\big)$ *with probability* $(1-\delta)$. *The total predicted constraint violations are bounded by* $\mathcal{C}_n(\pi, \widehat{b}^k) \leqslant \widetilde{O}\big(\frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + LSA^2 + \frac{(LSA^2)^2 H_{*,(2)}^2 M^2}{n^{1/2}}\big)$ *with probability* $(1-\delta)$, *where* $M_{\min} \coloneqq \min_s M(s)$.

Discussion: In Theorem 2 the first quantity upper bounding $\mathcal{L}_n(\pi, \widehat{b}^k)$ is denoted as the *safe MSE* when the safety budget $\widehat{Z}^k \geqslant 0$ and scales as $M^2(s_1^1)/n$. The second quantity is denoted as the *unsafe MSE* which is accumulated due to constraint violation ($\widehat{Z}^k < 0$) and sampling of the safe action 0. Finally, the third quantity is the MSE suffered due to estimation error of the variances $\sigma^2(s,a)$. Comparing the result of the Theorem 2 with the unconstrained setting of Mukherjee et al. (2022a) we have the additional quantity of $(MLSA^2 + H_{*,(2)})^2/n$ where $H_{*,(2)}$ is the problem-dependent quantity summed over all states. Observe that if all actions are safe then we have that $\mathcal{L}_n^*(\pi, \widehat{b}^k) = M^2(s_1^1)/n$ which recovers the MSE of the unconstraint setting in Carpentier and Munos (2011, 2012); Carpentier et al. (2015); Mukherjee et al. (2022a).

Proof (Overview) The agnostic SaVeR does not know the reward variances. The sampling rule in (4.8) ensures that the good variance event $\xi_{\nu,K}$ defined in (C.8) (step 2) holds such that SaVeR has good estimates of reward variances. Then, note that in the tree MDP $\mathbf{T}$ we have a closed form expression of $\mathbf{b}_*(s_j^\ell|a)$. We divide the total budget $n = n_f + n_u$ where n_f are the samples allocated when safety budget $\widehat{Z}^k \geqslant 0$. The n_f samples are also used by the exploration policy π_x to ensure a good estimate of

the constraint means as stated in the event $\xi_{c,K}$ (C.7). This is ensured by π_x and noting that $n > SA\log(1/\delta)/\Delta^2_{c,\min}$. The remaining samples from n_f are allocated for reducing the MSE by sampling according to $\arg\max_a(\mathbf{b}_*(a|s)/T^k_\ell(s,a))$. We again prove an upper and lower bound to $T_n(s,a)$ in (C.17) in step 4 and (C.18) in step 5. Finally using Theorem C.1 we can bound the MSE for the duration n_f for all actions $a \in \mathcal{A}\setminus\{0\}$ for each state s^ℓ_j in step 6. Now for an upper bound to constraint violations, we use the gap $\Delta^\alpha_c(s,a) \coloneqq (1-\alpha)\mu_{c,0}(s,a) - \mu_c(s,a)$ to bound how much each $a \in \mathcal{A}\setminus\{0\}$ in s^ℓ_j is underpulled and their pulls replaced by action $\{0\}$ weighted by $\pi(a|s^\ell_j)\sigma(s^\ell_j,a)$. This is captured by $H_{*,(2)}(s)$. Summing over all s, and horizon L gives the upper bound to the violations as shown in step 7. Finally, we also show a lower bound to constraint violations to bound the MSE for the duration when actions $a \in \mathcal{A}\setminus\{0\}$ are underpulled. This is shown in steps 8 and 9 where we equate the safety budget to 0 to obtain a lower bound to $T_n(s^\ell_j,0)$ for each state s^ℓ_j. Combining everything in step 10 gives the result. The proof is in Section C.3. ■

Note that we do not have a closed-form solution to $\mathbf{b}^k_*$ that both minimizes MSE as well as upholds (4.1) for all $k \in [K]$ (as opposed to Carpentier and Munos (2011); Mukherjee et al. (2022b)). Therefore, we now define two additional notions of regret. The first is the regret defined as $\overline{\mathcal{R}}_n = \mathcal{L}_n(\pi,\widehat{b}^k) - \overline{\mathcal{L}}^*_n(\pi,\mathbf{b}^k_*)$ where $\overline{\mathcal{L}}^*_n(\pi,\mathbf{b}^k_*)$ is the upper bound to the safe oracle MSE. The second is the constraint regret defined as follows: $\overline{\mathcal{R}}^c_n = \mathcal{C}_n(\pi,\widehat{b}^k) - \overline{\mathcal{C}}^*_n(\pi,\mathbf{b}^k_*)$ where $\overline{\mathcal{C}}^*_n(\pi,\mathbf{b}^k_*)$ is the upper bound to the oracle constraint violations. Note that the oracle knows the variances of reward and constraint-values for all state-action tuples (but does not know the mean of either). The following corollary bounds SaVeR regret.

Corollary 1. *Under Assumption 6, the constraint regret of SaVeR is bounded by* $\overline{\mathcal{R}}^c_n \leqslant O\big(\frac{\log(n)}{n^{1/2}}\big)$ *and the regret is bounded by* $\overline{\mathcal{R}}_n \leqslant O\big(\frac{\log(n)}{n^{3/2}}\big)$.

The proof is in Section C.4 and directly follows from Theorem 2, and Proposition 2. In Proposition 2 in Section C.4 we prove the MSE upper

bound of the oracle. Observe, that the regret decreases at a rate of $\widetilde{O}(n^{-3/2})$, faster than the rate of decrease of on-policy MSE of $\widetilde{O}(n^{-1})$. Thus, we have been able to answer the second main question of this paper affirmatively. We also state a constraint and regret upper bound in the bandit setting in Corollary 2 in Section C.4. Also, observe that our upper bound matches the rate in the lower bound shown in Theorem 1.

4.6 Extension to DAG

In this section, we approximate the solution in $\mathbf{T}$ to DAG $\mathcal{G}$ and formulate the safe algorithm for policy evaluation. We first define the DAG MDP in the following definition.

Definition 4.3. (***DAG MDP***) *A DAG MDP follows the same definition as the tree MDP in Theorem 4.1 except* $P(s'|s,a)$ *can be non-zero for any* s *in layer* ℓ, s' *in layer* $\ell+1$, *and any* a, *i.e., one can now reach* s' *through multiple previous state-action pairs.*

Then we state the following lemma from Mukherjee et al. (2022a).

Lemma 4.4. (***Proposition 3 of Mukherjee et al. (2022a)***) *Let* $\mathcal{G}$ *be a* 3*-depth,* A*-action DAG defined in Theorem 4.3. The minimal-MSE sampling proportions* $\mathbf{b}_*(a|s_1^1), \mathbf{b}_*(a|s_j^2)$ *depend on themselves such that* $\mathbf{b}(a|s_1^1) \propto f(1/\mathbf{b}(a|s_1^1))$ *and* $\mathbf{b}(a|s_j^2) \propto f(1/\mathbf{b}(a|s_j^2))$ *where* $f(\cdot)$ *is a function that hides other dependencies on variances of* s *and its children.*

The Theorem 4.4 (Mukherjee et al., 2022a) shows that one cannot derive a closed-form solution to $\mathbf{b}_*$ in $\mathcal{G}$ because of the existence of multiple paths to the same state resulting in a cyclical dependency. Note that in $\mathbf{T}$ there is only a single path to each state and this cyclical dependency does not arise. If we ignore the multiple path problem, we can approximate the optimal sampling proportion in $\mathcal{G}$ by using the tree formulation in the following

way: At every time t during an episode k call the Algorithm 4 to estimate $M_0(s)$ where $M_{t'}(s) \in \mathbb{R}^{L\times|\mathcal{S}|}$ stores the expected standard deviation of the state s at iteration t'. After L such iteration we use the value $B_0(s)$ to estimate $\mathbf{b}(a|s)$ as follows:

$$\mathbf{b}_*(a|s) \propto \sqrt{\pi^2(a|s)\left[\sigma^2(s,a)+\gamma^2\sum_{s'}P(s'|s,a)M_0^2(s)\right]}.$$

Note that for a terminal state s we have the transition probability $P(s'|s,a) = 0$ and then the $b(a|s) = \pi(a|s)\sigma(s,a)$. This iterative procedure follows from the tree formulation in Theorem C.2 and is necessary in $\mathcal{G}$ to take into account the multiple paths to a particular state. Algorithm 4 gives pseudocode for this procedure which takes inspiration from value-iteration for the episodic setting.

Algorithm 4 Estimate $B_0(s)$ for $\mathcal{G}$

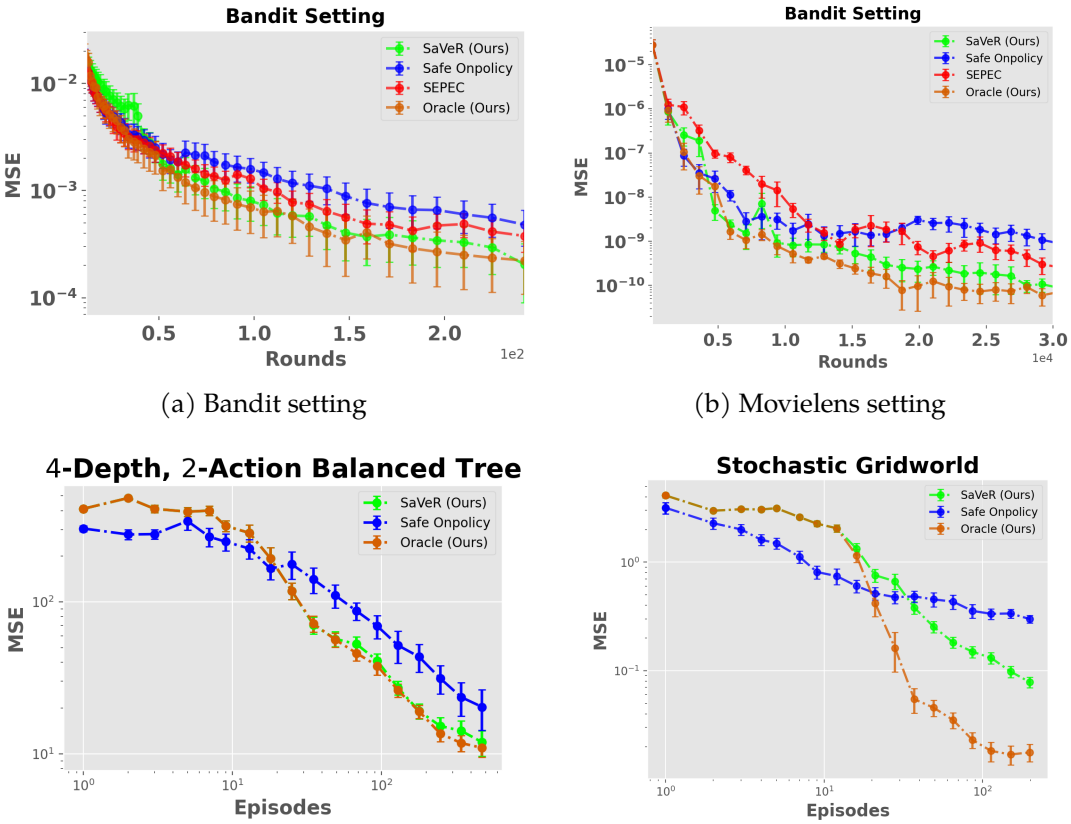
Initialize $B_L(s) = 0$ for all $s \in \mathcal{S}$
for $t' \in L-1,\dots,0$ **do**
    $B_{t'}(s) = \sum_a \left(\pi^2(a|s)\left(\sigma^2(s,a)\right.\right.$
        $\left.\left.+\gamma^2\sum_{s'}P(s'|s,a)B_{t'+1}^2(s)\right)\right)^{\frac{1}{2}}$
Return B_0.

Finally, the safe algorithm in $\mathcal{G}$ can be stated as follows: At episode k

$$\text{Play } \mathbf{b}^k = \begin{cases} \pi_e & \text{if } \widehat{Z}^k \geqslant 0, k \leqslant \sqrt{K} \\ \pi_{\widehat{b}^k} & \text{if } \widehat{Z}^k \geqslant 0, k > \sqrt{K} \\ \pi_0 & \text{if } \widehat{Z}^k < 0 \end{cases} \tag{4.10}$$

where $\pi_{\widehat{b}^k}$ for the episode k is defined as follows: For each time $\ell = 1,2,\dots,L$ sample action $A_\ell^k = \arg\max_a \frac{\widehat{b}^k(a|s_j^\ell)}{T_\ell^k(s_j^\ell,a)}$, where $\widehat{b}^k(a|s_j^\ell)$ is the plug-in estimate of $\mathbf{b}_*(a|s_j^\ell)$ that is obtained using Algorithm 4.

4.7 Experiments



(a) Bandit setting

(b) Movielens setting

(c) Tree MDP

(d) Grid MDP

Figure 4.1: MSE in different settings. The vertical axis (log-scaled) gives MSE and the horizontal axis is the number of episodes (or rounds for bandits). Confidence bars show one standard error.

In this section, we show numerical experiments validating our theoretical results. The full experimental details and numerical results are in Section C.6. We test the oracle, and SaVeR algorithm and introduce a method that we call safe on-policy. The safe on-policy algorithm follows the target policy π when the safety budget is positive and plays baseline policy π_0 when the safety budget is negative. We also test against the

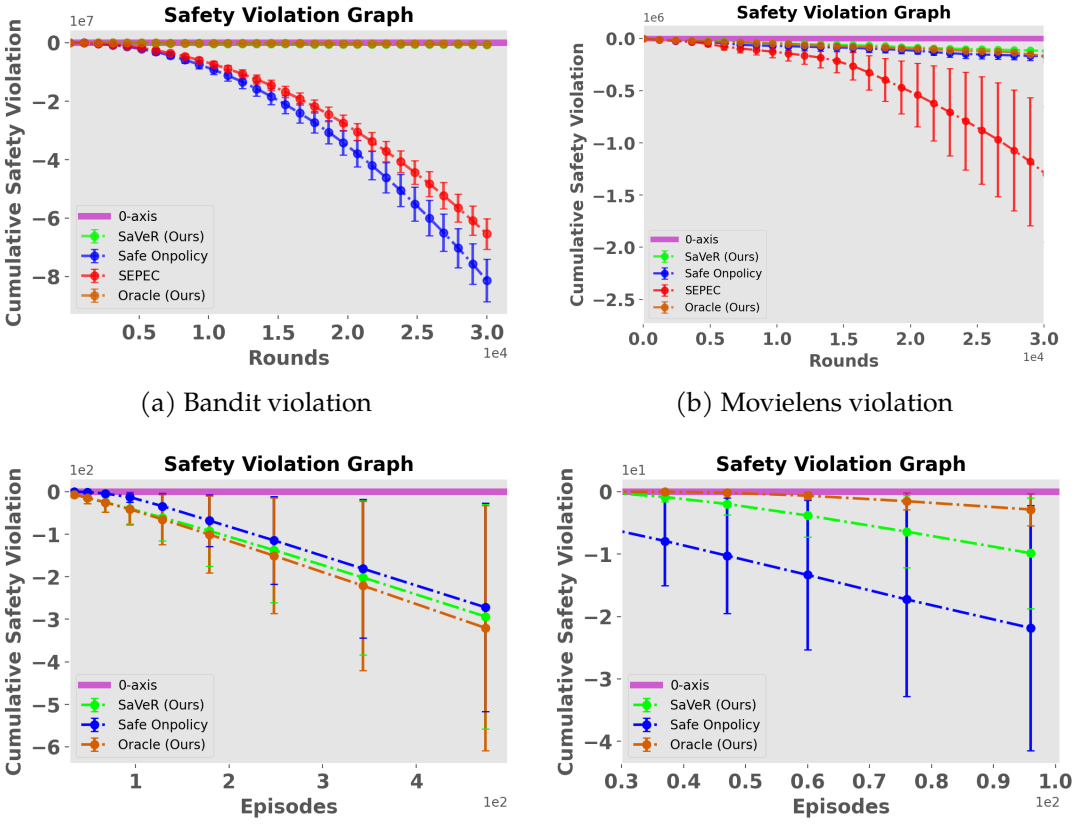
SEPEC (Wan et al., 2022) algorithm for the bandit setting which uses importance sampling to safely collect data for policy evaluation. Note that the bandit setting consists of a single state and every episode K consists of a single time step $L = 1$. Figure 4.1 shows the MSE obtained by each algorithm for a varying number of episodes. In Figure 4.2, we show that all algorithms respect the constraint but that the oracle and SaVeR are not excessively conservative.

Experiment 1 (Bandit): We implement a general bandit environment with $A = 11$ and show that SaVeR achieves lower MSE than SEPEC and safe on-policy algorithm as the number of rounds increases. The performance is shown in Figure 4.1a. From Figure 4.2a we see that SaVeR, and oracle do not oversample the safe action but allocate the right amount to be just safe. They allocate more samples to reduce the MSE, whereas the safe on-policy and SEPEC over-sample the safe action instead of focusing on reducing the MSE.

Experiment 2 (Movielens): We conduct this experiment on the real-life Movielens 1M dataset (Lam and Herlocker, 2016) for $A = 30$ actions and show that SaVeR achieves lower MSE than safe on-policy and SEPEC algorithm as the number of rounds increases. The performance is shown in Figure 4.1b. From Figure 4.2b, we see that SaVeR and oracle SaVeR, and the oracle do not oversample the safe action compared to SEPEC.

Experiment 3 (Tree): We experiment with a 4-depth 2-action deterministic tree MDP consisting of 15 states. With increasing episodes SaVeR reaches lower MSE than safe on-policy and eventually matches the oracle's MSE in Figure 4.1c. In Figure 4.2c the SaVeR and oracle run the baseline policy almost similar number of times compared to the safe on-policy.

Experiment 4 (Gridworld): This setting consist of a 4×4 stochastic gridworld of 16 grid cells. We point out that Gridworld has a DAG structure (due to the finite horizon) which violates the tree structure assumption under which the oracle and SaVeR bounds were derived. Nevertheless,



(a) Bandit violation

(b) Movielens violation

(c) Tree MDP violation

(d) Grid MDP violation

Figure 4.2: The vertical axis gives cumulative constraint violation and the horizontal axis is the number of episodes/rounds. The 0-axis is shown in pink. A safe algorithm has its plot below the 0-axis with the plot showing the cumulative unsafe budget.

both SaVeR and oracle reach lower MSE with increasing episodes compared to safe onpolicy in Figure 4.1d. We use (4.10) to estimate $\widehat{b}$ in this setting. In Figure 4.2d we see that SaVeR allocates more samples to reduce the MSE, whereas the safe on-policy runs the baseline policy more instead of focusing on reducing the MSE.

4.8 Conclusions and Future Directions

In this paper, we studied the question of how to take action to build a dataset for minimal-variance policy evaluation of a fixed target policy under a safety constraint (4.1). We developed a theoretical foundation for data collection in policy evaluation by showing that there exists a class of MDPs (namely tree-structured MDPs $\mathbf{T}$) where safe policy evaluation is intractable. We then showed the necessary condition for $\mathbf{T}$ to be tractable such that the optimal behavior policy can collect data without violating safety constraints. We then proved the first lower bound for this setting under the tractability conditions that scales as $\widetilde{\Omega}(n^{-3/2})$, where $\widetilde{\Omega}$ hides log factors. We then introduced a practical algorithm, SaVeR, that approximates the optimal behavior strategy by computing an upper confidence bound on the variance of the cumulative cost in place of the true cost variances in the optimal behavior strategy. We bound the finite-sample regret (excess MSE) of SaVeR and show that it scales as $\widetilde{O}(n^{-3/2})$ matching the lower bound. Hence, we answer both the questions raised in the introduction positively. In the future, we would like to extend our derivation of optimal data collection strategies and regret analysis of SaVeR to linear/contextual bandits and more general MDPs.

Part III

Adaptive Data Collection for Multi-task Learning

5 MULTI-TASK REPRESENTATION LEARNING FOR FIXED CONFIDENCE PURE EXPLORATION IN BILINEAR BANDITS

Bilinear bandits (Jun et al., 2019; Lu et al., 2021; Kang et al., 2022) are an important class of sequential decision-making problems. In bilinear bandits (as opposed to the standard linear bandit setting) we are given a pair of arms $\mathbf{x}_t \in \mathbb{R}^{d_1}$ and $\mathbf{z}_t \in \mathbb{R}^{d_2}$ at every round t and the interaction of this pair of arms with a low-rank hidden parameter, $\boldsymbol{\Theta}_* \in \mathbb{R}^{d_1 \times d_2}$ generates the noisy feedback (reward) $r_t = \mathbf{x}_t^\top \boldsymbol{\Theta}_* \mathbf{z}_t + \eta_t$. The η_t is random 1-subGaussian noise.

A lot of real-world applications exhibit the above bilinear feedback structure, particularly applications that involve selecting pairs of items and evaluating their compatibility. For example, in a drug discovery application, scientists may want to determine whether a particular (drug, protein) pair interacts in the desired way (Luo et al., 2017). Likewise, an online dating service might match a pair of people and gather feedback about their compatibility (Shen et al., 2023a). A clothing website's recommendation system may suggest a pair of items (top, bottom) for a customer based on their likelihood of matching (Reyes et al., 2021). In all of these scenarios, the two items are considered as a single unit, and the system must utilize available feature vectors $(\mathbf{x}_t, \mathbf{z}_t)$ to learn which features of the pairs are most indicative of positive feedback in order to make effective recommendations. All the previous works in this setting (Jun et al., 2019; Lu et al., 2021; Kang et al., 2022) exclusively focused on maximizing the number of pairs with desired interactions discovered over time (regret minimization). However, in many real-world applications where obtaining a sample is expensive and time-consuming, e.g., clinical trials (Zhao et al., 2009; Zhang et al., 2012), it is often desirable to identify the optimal option using as few samples as possible, i.e., we face the pure exploration scenario (Fiez et al., 2019; Katz-Samuels et al., 2020) rather

than regret minimization.

Moreover, in various decision-making scenarios, we may encounter multiple interrelated tasks such as treatment planning for different diseases (Bragman et al., 2018) and content optimization for multiple websites (Agarwal et al., 2009). Often, there exists a shared representation among these tasks, such as the features of drugs or the representations of website items. Therefore, we can leverage this shared representation to accelerate learning. This area of research is called multi-task representation learning and has recently generated a lot of attention in machine learning (Bengio et al., 2013; Li et al., 2014; Maurer et al., 2016; Du et al., 2020; Tripuraneni et al., 2021). There are many applications of this multi-task representation learning in real-world settings. For instance, in clinical treatment planning, we seek to determine the optimal treatments for multiple diseases, and there may exist a low-dimensional representation common to multiple diseases. To avoid the time-consuming process of conducting clinical trials for individual tasks and collecting samples, we utilize the shared representation and decrease the number of required samples.

The above multi-task representation learning naturally shows up in bilinear bandit setting as follows: Let there be M tasks indexed as $m = 1, 2, \ldots, M$ with each task having its own hidden parameter $\mathbf{\Theta}_{m,*} \in \mathbb{R}^{d_1 \times d_2}$. Let each $\mathbf{\Theta}_{m,*}$ has a decomposition of $\mathbf{\Theta}_{m,*} = \mathbf{B}_1 \mathbf{S}_{m,*} \mathbf{B}_2^\top$, where $\mathbf{B}_1 \in \mathbb{R}^{d_1 \times k_1}$ and $\mathbf{B}_2 \in \mathbb{R}^{d_2 \times k_2}$ are shared across tasks, but $\mathbf{S}_{m,*} \in \mathbb{R}^{k_1 \times k_2}$ is specific for task m. We assume that $k_1, k_2 \ll d_1, d_2$ and $M \gg d_1, d_2$. Thus, $\mathbf{B}_1$ and $\mathbf{B}_2$ provide a means of dimensionality reduction. Furthermore, we assume that each $\mathbf{S}_{m,*}$ has rank $r \ll \min\{k_1, k_2\}$. In the terminology of multi-task representation learning $\mathbf{B}_1, \mathbf{B}_2$ are called *feature extractors* and $\mathbf{x}_{m,t}, \mathbf{z}_{m,t}$ are called *rich observations* (Yang et al., 2020, 2022a; Du et al.,

2023). The reward for the task $m \in \{1, 2, \ldots, M\}$ at round t is

$$r_{m,t} = \mathbf{x}_{m,t}^\top \mathbf{\Theta}_{m,*} \mathbf{z}_{m,t} + \eta_{m,t} = \underbrace{\mathbf{x}_{m,t}^\top \mathbf{B}_1}_{\mathbf{g}_{m,t}^\top} \mathbf{S}_{m,*} \underbrace{\mathbf{B}_2^\top \mathbf{z}_{m,t}}_{\mathbf{x}_{m,t}} + \eta_{m,t}$$
$$= \mathbf{g}_{m,t}^\top \mathbf{S}_{m,*} \mathbf{x}_{m,t} + \eta_{m,t}. \tag{5.1}$$

Observe that similar to the learning procedure in Yang et al. (2020, 2022a), at each round $t = 1, 2, \cdots$, for each task $m \in [M]$, the learner selects a left and right action $\mathbf{x}_{m,t} \in \mathcal{X}$ and $\mathbf{z}_{m,t} \in \mathcal{Z}$. After the player commits the batch of actions for each task $\{\mathbf{x}_{m,t}, \mathbf{z}_{m,t} : m \in [M]\}$, it receives the batch of rewards $\{r_{m,t} : m \in [M]\}$. Also note that in (5.1) we define the $\widetilde{\mathbf{g}}_{m,t} \in \mathbb{R}^{k_1}, \widetilde{\mathbf{v}}_{m,t} \in \mathbb{R}^{k_2}$ as the latent features, and both $\widetilde{\mathbf{g}}_{m,t}, \widetilde{\mathbf{v}}_{m,t}$ are unknown to the learner and needs to be learned for each task m (hence the name multi-task representation learning).

In this paper, we focus on pure exploration for multi-task representation learning in bilinear bandits where the goal is to find the optimal left arm $\mathbf{x}_{m,*}$ and right arm $\mathbf{z}_{m,*}$ for each task m with a minimum number of samples (fixed confidence setting). First, consider a single-task setting and let $\mathbf{\Theta}_*$ have low rank r. Let the SVD of the $\mathbf{\Theta}_* = \overline{U}\mathbf{D}\mathbf{V}^\top$. Prima-facie, if $\overline{U}$ and $\mathbf{V}$ are known then one might want to project all the left and right arms in the $r \times r$ subspace of $\overline{U}$ and $\mathbf{V}$ and reduce the bilinear bandit problem into a r^2 dimension linear bandit setting. Then one can apply one of the algorithms from Soare et al. (2014); Fiez et al. (2019); Katz-Samuels et al. (2020) to solve this r^2 dimensional linear bandit pure exploration problem. Following the analysis of this line of work (in linear bandits) (Mason et al., 2021; Mukherjee et al., 2022b, 2023a) one might conjecture that a sample complexity bound of $\widetilde{O}(r^2/\Delta^2)$ is possible where Δ is the minimum reward gap and $\widetilde{O}(\cdot)$ hides log factors. Similarly, for the multi-task setting one might be tempted to use the linear bandit analysis of Du et al. (2023) to convert this problem into M concurrent r^2 dimensional linear bandit

problems with shared representation and achieve a sample complexity bound of $\widetilde{O}(Mr^2/\Delta^2)$. However, these matrices (subspaces) are unknown and so there is a model mismatch as noted in the regret analysis of bilinear bandits (Jun et al., 2019; Lu et al., 2021; Kang et al., 2022). Thus it is difficult to apply the r^2 dimensional linear bandit sample complexity analysis. Following the regret analysis of bilinear bandit setting by Jun et al. (2019); Lu et al. (2021); Kang et al. (2022) we know that the effective dimension is actually $(d_1 + d_2)r$. Similarly for the multi-task representation learning the effective dimension should scale with the learned latent features $(k_1 + k_2)r$. Hence the natural questions to ask are these:

1) Can we design a single-task pure exploration bilinear bandit algorithm whose sample complexity scales as $\widetilde{O}((d_1 + d_2)r/\Delta^2)$*?*

2) Can we design an algorithm for multi-task pure exploration bilinear bandit problem that can learn the latent features and has sample complexity that scales as $\widetilde{O}(M(k_1 + k_2)r/\Delta^2)$*?*

In this paper, we answer both these questions affirmatively. In doing so, we make the following novel contributions to the growing literature of multi-task representation learning in online settings:

1) We formulate the multi-task bilinear representation learning problem. To our knowledge, this is the first work that explores pure exploration in a multi-task bilinear representation learning setting.

2) We proposed the algorithm GOBLIN for a single-task pure exploration bilinear bandit setting whose sample complexity scales as $\widetilde{O}((d_1 + d_2)r/\Delta^2)$. This improves over RAGE (Fiez et al., 2019) whose sample complexity scales as $\widetilde{O}((d_1 d_2)/\Delta^2)$.

3) Our algorithm GOBLIN for multi-task pure exploration bilinear bandit problem learns the latent features and has sample complexity that scales as $\widetilde{O}(M(k_1 + k_2)r/\Delta^2)$. This improves over DouExpDes (Du et al.,

2023) whose samples complexity scales as $\widetilde{O}(M(k_1+k_2)/\Delta^2)$.

5.1 Preliminaries

Preliminaries: We assume that $\|\mathbf{x}\|_2 \leqslant 1$, $\|\mathbf{z}\|_2 \leqslant 1$, $\|\mathbf{\Theta}_*\|_F \leqslant S_0$ and the r-th largest singular value of $\mathbf{\Theta}_* \in \mathbb{R}^{d_1 \times d_2}$ is S_r. Let $p := d_1 d_2$ denote the ambient dimension, and $k = (d_1 + d_2)r$ denote the effective dimension. Let $[n] := \{1, 2, \ldots, n\}$. Let $\mathbf{x}_*, \mathbf{z}_* := \arg\max_{\mathbf{x},\mathbf{z}} \mathbf{x}^\top \mathbf{\Theta}_* \mathbf{z}$. For any $\mathbf{x}, \mathbf{z}$ define the gap $\Delta(\mathbf{x}, \mathbf{z}) := \mathbf{x}_*^\top \mathbf{\Theta}_* \mathbf{z}_* - \mathbf{x}^\top \mathbf{\Theta}_* \mathbf{z}$ and furthermore $\Delta = \min_{\mathbf{x} \neq \mathbf{x}_*, \mathbf{z} \neq \mathbf{z}_*} \Delta(\mathbf{x}, \mathbf{z})$. Similarly, for any arbitrary vector $\mathbf{w} \in \mathcal{W}$ define the gap of $\mathbf{w} \in \mathbb{R}^p$ as $\Delta(\mathbf{w}) := (\mathbf{w}_* - \mathbf{w})^\top \theta_*$, for some $\theta_* \in \mathbb{R}^p$ and furthermore, $\Delta = \min_{\mathbf{w} \neq \mathbf{w}_*} \Delta(\mathbf{w})$. If $\mathbf{A} \in \mathbb{R}^{d \times d}_{\geqslant 0}$ is a positive semidefinite matrix, and $\mathbf{w} \in \mathbb{R}^p$ is a vector, let $\|\mathbf{w}\|_{\mathbf{A}}^2 := \mathbf{w}^\top \mathbf{A} \mathbf{w}$ denote the induced semi-norm. Given any vector $\mathbf{b} \in \mathbb{R}^{|\mathcal{W}|}$ we denote the $\mathbf{w}$-th component as $\mathbf{b}_\mathbf{w}$. Let $\Delta_\mathcal{W} := \left\{\mathbf{b} \in \mathbb{R}^{|\mathcal{W}|} : \mathbf{b}_\mathbf{w} \geqslant 0, \sum_{\mathbf{w} \in \mathcal{W}} \mathbf{b}_\mathbf{w} = 1\right\}$ denote the set of probability distributions on $\mathcal{W}$. We define $\mathcal{Y}(\mathcal{W}) = \{\mathbf{w} - \mathbf{w}' : \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}, \mathbf{w} \neq \mathbf{w}'\}$ as the directions obtained from the differences between each pair of arms and $\mathcal{Y}^*(\mathcal{W}) = \{\mathbf{w}_* - \mathbf{w} : \forall \mathbf{w} \in \mathcal{W} \backslash \mathbf{w}_*\}$ as the directions obtained from the differences between the optimal arm and each suboptimal arm.

5.2 Pure Exploration in Single-Task Bilinear Bandits

In this section, we consider pure exploration in a single-task bilinear bandit setting as a warm-up to the main goal of learning representations for the multi-task bilinear bandit. To our knowledge, this is the first study of pure exploration in single-task bilinear bandits. We first recall the single-task bilinear bandit setting as follows: At every round $t = 1, 2, \ldots$ the learner observes the reward $r_t = \mathbf{x}_t^\top \mathbf{\Theta}_* \mathbf{z}_t + \eta_t$ where the low rank hidden parameter $\mathbf{\Theta}_* \in \mathbb{R}^{d_1 \times d_2}$ is unknown to the learner, $\mathbf{x}_t \in \mathbb{R}^{d_1}$, $\mathbf{z}_t \in \mathbb{R}^{d_2}$ are

visible to the learner, and η_t is a 1-sub-Gaussian noise. We assume that the matrix $\boldsymbol{\Theta}_*$ has a low rank r which is known to the learner and $d_1, d_2 \gg r$. Finally recall that the goal is to identify the optimal left and right arms $\mathbf{x}_*, \mathbf{z}_*$ with a minimum number of samples.

We propose a phase-based, two-stage arm elimination algorithm called **G-O**ptimal Design for **Bilin**ear Bandits (abbreviated as GOBLIN). GOBLIN proceeds in phases indexed by $\ell = 1, 2, \ldots$ As this is a pure-exploration problem, the total number of samples is controlled by the total phases which depends on the intrinsic problem complexity. Each phase ℓ of GOBLIN consists of two stages; the estimation of $\boldsymbol{\Theta}_*$ stage, which runs for τ_ℓ^E rounds, and pure exploration in rotated arms stage that runs for τ_ℓ^G rounds. We will define τ_ℓ^E in Section 5.2, while rotated arms and τ_ℓ^G are defined in Section 5.2. At the end of every phase, GOBLIN eliminates sub-optimal arms to build the active set for the next phase and stops when only the optimal left and right arms are remaining. Now we discuss the individual stages that occur at every phase ℓ of GOBLIN.

Estimating Subspaces of $\boldsymbol{\Theta}_*$ (Stage 1 of the ℓ-th phase)

In the first stage of phase ℓ, GOBLIN estimates the row and column subspaces $\boldsymbol{\Theta}_*$. Then GOBLIN uses these estimates to reduce the bilinear bandit problem in the original ambient dimension $p := d_1 d_2$ to a lower effective dimension $k := (d_1 + d_2)r$. To do this, GOBLIN first vectorizes the $\mathbf{x} \in \mathbb{R}^{d_1}, \mathbf{z} \in \mathbb{R}^{d_2}$ into a new vector $\overline{\mathbf{w}} \in \mathbb{R}^p$ and then solves the E-optimal design in Step 3 of Algorithm 5 (Pukelsheim, 2006; Jun et al., 2019; Du et al., 2023). Let the solution to the E-optimal design problem at the stage 1 of ℓ-th phase be denoted by $\mathbf{b}_\ell^E$. Then GOBLIN samples each $\overline{\mathbf{w}}$ for $\lceil \tau_\ell^E \mathbf{b}_{\ell,\overline{\mathbf{w}}}^E \rceil$ times, where $\tau_\ell^E = \widetilde{O}(\sqrt{d_1 d_2 r}/S_r)$ (step 7 of Algorithm 5). In this paper, we sample an arm $\lceil \tau_\ell^E \mathbf{b}_{\ell,\overline{\mathbf{w}}}^E \rceil$ number of times. However, this may lead to over-sampling of an arm than what the design (G or E-optimal) is actually suggesting. However, we can match the number of allocations of

an arm to the design using an *efficcient Rounding Procedures* (see Pukelsheim (2006); Fiez et al. (2019)). Let $\widehat{\mathbf{\Theta}}_\ell$ be estimate of $\mathbf{\Theta}_*$ in stage 1 of phase ℓ. GOBLIN estimates this by solving the following well-defined regularized minimization problem with nuclear norm penalty:

$$\widehat{\mathbf{\Theta}}_\ell = \underset{\mathbf{\Theta}\in\mathbb{R}^{d_1\times d_2}}{\arg\min}\, L_\ell(\mathbf{\Theta}) + \gamma_\ell\|\mathbf{\Theta}\|_{\text{nuc}}, \quad L_\ell(\mathbf{\Theta}) = \langle\mathbf{\Theta},\mathbf{\Theta}\rangle - \frac{2}{\tau_\ell^E}\sum_{s=1}^{\tau_\ell^E}\langle\widetilde{\psi}_\nu(r_s\cdot Q(\mathbf{x}_s\mathbf{z}_s^\top)),\mathbf{\Theta}\rangle \tag{5.2}$$

where $Q(\cdot)$, $\widetilde{\psi}_\nu(\cdot)$, are appropriate functions stated in Theorem D.5, D.7 respectively in Section D.3. The $Q(\cdot)$ function takes as input the rank-one matrix $\mathbf{x}_s\mathbf{z}_s^\top$ which is obtained after reshaping $\overline{\mathbf{w}}_s$. Note that $\mathbf{x}_s$, and $\mathbf{z}_s$ are the observed vectors in d_1 and d_2 dimension and $\widehat{\mathbf{\Theta}}_\ell \in \mathbb{R}^{d_1\times d_2}$ Finally, set the regularization parameter $\gamma_\ell := 4\sqrt{\frac{2\left(4+S_0^2\right)C\,d_1\,d_2\log(2(d_1+d_2)/\delta)}{\tau_\ell^E}}$. This is in step 8 of Algorithm 5.

Optimal Design for Rotated Arms (Stage 2 of ℓ-th phase)

In stage 2 of phase ℓ, GOBLIN leverages the information about the learned sub-space of $\mathbf{\Theta}_*$ to rotate the arm set and then run the optimal design on the rotated arm set. Once we recover $\widehat{\mathbf{\Theta}}_\ell$, one might be tempted to run a pure exploration algorithm (Soare et al., 2014; Fiez et al., 2019; Katz-Samuels et al., 2020; Zhu et al., 2021) to identify $\mathbf{x}_*$ and $\mathbf{z}_*$. However, then the sample complexity will scale with d_1d_2. In contrast GOBLIN uses the information about the learned sub-space of $\mathbf{\Theta}_*$ to reduce the problem from ambient dimension d_1d_2 to effective dimension $(d_1+d_2)r$. This reduction is done as follows: Let $\widehat{\mathbf{\Theta}}_\ell = \widehat{\mathbf{U}}_\ell\widehat{\mathbf{D}}_\ell\widehat{\mathbf{V}}_\ell^\top$ be the SVD of $\widehat{\mathbf{\Theta}}_\ell$ in the ℓ-th phase. Let $\widehat{\mathbf{U}}_\perp^\ell$ and $\widehat{\mathbf{V}}_\perp^\ell$ be orthonormal bases of the complementary subspaces of $\widehat{\mathbf{U}}_\ell$ and $\widehat{\mathbf{V}}_\ell$ respectively. Let $\mathcal{X}_\ell$ and $\mathcal{Z}_\ell$ be the active set of arms in the stage 2 of phase ℓ. Then rotate the arm sets such that new rotated arm sets are

as follows:

$$\underline{\mathcal{X}}_\ell = \{\underline{\mathbf{x}} = [\widehat{\mathbf{U}}_\ell \widehat{\mathbf{U}}_\ell^\perp]^\top \mathbf{x} \mid \mathbf{x} \in \mathcal{X}_\ell\}, \underline{\mathcal{Z}}_\ell = \{\underline{\mathbf{z}} = [\widehat{\mathbf{V}}_\ell \widehat{\mathbf{V}}_\ell^\perp]^\top \mathbf{z} \mid \mathbf{z} \in \mathcal{Z}_\ell\}. \tag{5.3}$$

Let $\widehat{\mathbf{H}}_\ell = [\widehat{\mathbf{U}}_\ell \widehat{\mathbf{U}}_\ell^\perp]^\top \widehat{\mathbf{\Theta}}_\ell [\widehat{\mathbf{V}}_\ell \widehat{\mathbf{V}}_\ell^\perp]$. Then define vectorized arm set so that the last $(d_1 - r) \cdot (d_2 - r)$ components are from the complementary subspaces as follows:

$$\begin{aligned}
\underline{\mathcal{W}}_\ell = \Big\{ \big[\mathbf{vec}\left(\underline{\mathbf{x}}_{1:r} \underline{\mathbf{z}}_{1:r}^\top\right); \mathbf{vec}\left(\underline{\mathbf{x}}_{r+1:d_1} \underline{\mathbf{z}}_{1:r}^\top\right); \mathbf{vec}\left(\underline{\mathbf{x}}_{1:r} \underline{\mathbf{z}}_{r+1:d_2}^\top\right); \\
\mathbf{vec}\left(\underline{\mathbf{x}}_{r+1:d_1} \underline{\mathbf{z}}_{r+1:d_2}^\top\right) \big] \in \mathbb{R}^{d_1 d_2} : \underline{\mathbf{x}} \in \mathcal{X}_\ell, \underline{\mathbf{z}} \in \mathcal{Z}_\ell \Big\} \\
\widehat{\theta}_{\ell,1:k} = [\mathbf{vec}(\widehat{\mathbf{H}}_{\ell,1:r,1:r}); \mathbf{vec}(\widehat{\mathbf{H}}_{\ell,r+1:d_1,1:r}); \mathbf{vec}(\widehat{\mathbf{H}}_{\ell,1:r,r+1:d_2})], \\
\widehat{\theta}_{\ell,k+1:p} = \mathbf{vec}(\widehat{\mathbf{H}}_{\ell,r+1:d_1,r+1:d_2}).
\end{aligned} \tag{5.4}$$

which implies $\|\widehat{\theta}_{k+1:p}\|_2 = O\left(d_1 d_2 r / \tau_\ell^E\right)$ by Theorem D.3 in Section D.1. So the last $p - k$ components of $\widehat{\theta}_\ell$ are very small compared to the first k components. Hence, GOBLIN has now reduced the $d_1 d_2$ dimensional linear bandit to $(d_1 + d_2)r$ dimensional linear bandit using (5.3), (5.4). This is shown in step 10 of Algorithm 5.

Now in stage 2 of phase ℓ, GOBLIN implements G-optimal design (Pukelsheim, 2006; Fiez et al., 2019) in the rotated arm set $\underline{\mathcal{X}}_\ell, \underline{\mathcal{Z}}_\ell$ defined in (5.3). To do this, first GOBLIN defines the rotated vector $\underline{\mathbf{w}} = [\underline{\mathbf{x}}_{1:d_1}; \underline{\mathbf{z}}_{1:d_2}] \in \mathbb{R}^p$ that belong to the set $\underline{\mathcal{W}}_\ell$. Then GOBLIN solves the G-optimal design (Pukelsheim, 2006) as follows:

$$\widehat{b}_\ell^G = \arg\min_{\mathbf{b}_{\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}}, \underline{\mathbf{w}}' \in \underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}} \in \mathcal{W}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}} \underline{\mathbf{w}}^\top + \mathbf{\Lambda}_\ell / n)^{-1}}. \tag{5.5}$$

This is shown in step 11 of Algorithm 5 and $\mathbf{\Lambda}_\ell$ is defined in (5.6). It can be shown that sampling according to $\widehat{b}_\ell^G$ leads to the optimal sample complexity. This is discussed in Theorem D.4 in Section D.2. The key point to note from (5.5) is that due to the estimation in the rotated arm space $\underline{\mathcal{W}}_\ell$ we are guaranteed that the support of $\mathbf{supp}(\widehat{b}_\ell^G) \leqslant \widetilde{O}(k(k+1)/2)$

(Pukelsheim, 2006). On the other hand, if the G-optimal design of Fiez et al. (2019); Katz-Samuels et al. (2020) are run in $d_1 d_2$ dimension then the support of $\widehat{b}_\ell^G$ will scale with $d_1 d_2$ which will lead to higher sample complexity. Then GOBLIN samples each $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell$ for $\lceil \tau_\ell^G \mathbf{b}_{\ell,\underline{\mathbf{w}}}^G \rceil$ times, where $\tau_\ell^G := \lceil \frac{8B_*^\ell \rho^G(\mathcal{Y}(\mathcal{W}_\ell)) \log(4\ell^2|\mathcal{W}|/\delta)}{\epsilon_\ell^2} \rceil$. Note that the total length of phase ℓ, combining stages 1 and 2 is $(\tau_\ell^E + \tau_\ell^G)$ rounds. Observe that the stage 1 design is on the whole arm set $\overline{\mathcal{W}}$ whereas stage 2 design is on the refined active set $\underline{\mathcal{W}}_\ell$.

Let the observed features in stage 2 of phase ℓ be denoted by $\underline{\mathbf{W}}_\ell \in \mathbb{R}^{\tau_\ell^G \times p}$, and $\mathbf{r}_\ell \in \mathbb{R}^{\tau_\ell^G}$ be the observed rewards. Define the diagonal matrix $\boldsymbol{\Lambda}_\ell$ as

$$\boldsymbol{\Lambda}_\ell = \mathbf{diag}[\underbrace{\lambda, \ldots, \lambda}_{k}, \underbrace{\lambda_\ell^\perp, \ldots, \lambda_\ell^\perp}_{p-k}] \tag{5.6}$$

where, $\lambda_\ell^\perp := \tau_{\ell-1}^G / 8k \log(1 + \tau_{\ell-1}^G/\lambda) \gg \lambda$. Deviating from Soare et al. (2014); Fiez et al. (2019) GOBLIN constructs a regularized least square estimator at phase ℓ as follows

$$\widehat{\theta}_\ell = \underset{\theta \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \|\underline{\mathbf{W}}_\ell \theta - \mathbf{r}_\ell\|_2^2 + \frac{1}{2} \|\theta\|_{\boldsymbol{\Lambda}_\ell}^2. \tag{5.7}$$

This regularized least square estimator in (5.7) forces the last $p - k$ components of $\widehat{\theta}_\ell$ to be very small compared to the first k components. Then GOBLIN builds the estimate $\widehat{\theta}_\ell$ from (5.7) only from the observations from this phase (step 13 in Algorithm 5) and eliminates sub-optimal actions in step 14 in Algorithm 5 using the estimator $\widehat{\theta}_\ell$. Finally GOBLIN eliminates sub-optimal arms to build the next phase active set $\underline{\mathcal{W}}_\ell$ and stops when $|\underline{\mathcal{W}}_\ell| = 1$. GOBLIN outputs the arm in $\underline{\mathcal{W}}_\ell$ and reshapes it to get the $\widehat{\mathbf{x}}_*$ and $\widehat{\mathbf{z}}_*$. The full pseudocode is presented in Algorithm 5.

Algorithm 5 G-Optimal Design for Bilinear Bandits (GOBLIN) for single-task setting

Input: arm set $\mathcal{X}, \mathcal{Z}$, confidence δ, rank r of $\boldsymbol{\Theta}_*$, spectral bound S_r of $\boldsymbol{\Theta}_*$, $S, S_\ell^\perp \coloneqq \frac{8d_1d_2r}{\tau_\ell^E S_r^2}\log\left(\frac{d_1+d_2}{\delta_\ell}\right), \lambda, \lambda_\ell^\perp \coloneqq \tau_{\ell-1}^G/8(d_1+d_2)r\log(1+\frac{\tau_{\ell-1}^G}{\lambda})$. Let $p \coloneqq d_1d_2$, $k \coloneqq (d_1+d_2)r$.
Let $\underline{\mathcal{W}}_1 \leftarrow \underline{\mathcal{W}}, \ell \leftarrow 1$, $\tau_0^G \coloneqq \log(4\ell^2|\mathcal{X}|/\delta)$. Define $\boldsymbol{\Lambda}_\ell$ as in (5.6), $B_*^\ell \coloneqq (8\sqrt{\lambda}S + \sqrt{\lambda_\ell^\perp}S_\ell^\perp)$.
Define a vectorized arm $\overline{\mathbf{w}} \coloneqq [\mathbf{x}_{1:d_1}; \mathbf{z}_{1:d_2}]$ and $\overline{\mathbf{w}} \in \overline{\mathcal{W}}$. Let $\tau_\ell^E \coloneqq \frac{\sqrt{8d_1d_2r\log(4\ell^2|\mathcal{W}|/\delta_\ell)}}{S_r}$. Let the E-optimal design be $\mathbf{b}_\ell^E \coloneqq \arg\min_{\mathbf{b}\in\triangle_{\overline{\mathcal{W}}}} \left\|\left(\sum_{\overline{\mathbf{w}}\in\overline{\mathcal{W}}} \mathbf{b}_{\overline{\mathbf{w}}}\overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top\right)^{-1}\right\|$.
while $|\underline{\mathcal{W}}_\ell| > 1$ **do**
  $\epsilon_\ell = 2^{-\ell}$, $\delta_\ell = \delta/\ell^2$.
  (Stage 1:) Explore the Low-Rank Subspace
  Pull arm $\overline{\mathbf{w}} \in \overline{\mathcal{W}}$ exactly $\left\lceil \widehat{b}_{\ell,\overline{\mathbf{w}}}^E \tau_\ell^E \right\rceil$ times and observe rewards r_t, for $t = 1, \ldots, \tau_\ell^E$.
  Compute $\widehat{\boldsymbol{\Theta}}_\ell$ using (5.2).
  (Stage 2:) Reduction to low dimensional linear bandits
  Let the SVD of $\widehat{\boldsymbol{\Theta}}_\ell = \widehat{\mathbf{U}}_\ell\widehat{\mathbf{D}}_\ell\widehat{\mathbf{V}}_\ell^\top$. Rotate arms in active set $\underline{\mathcal{W}}_{\ell-1}$ to build $\underline{\mathcal{W}}_\ell$ following (5.4).
  Let $\widehat{b}_\ell^G \coloneqq \arg\min_{\mathbf{b}_{\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell/n)^{-1}}$.
  Define $\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \coloneqq \min_{\mathbf{b}_{\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell/n)^{-1}}$.
  Set $\tau_\ell^G \coloneqq \left\lceil \frac{64B_*^\ell\rho^G(\mathcal{Y}(\mathcal{W}_\ell))\log(4\ell^2|\mathcal{W}|/\delta_\ell)}{\epsilon_\ell^2} \right\rceil$. Then pull arm $\underline{\mathbf{w}} \in \underline{\mathcal{W}}$ exactly $\left\lceil \widehat{b}_{\ell,\underline{\mathbf{w}}}^G \tau_\ell^G \right\rceil$ times and construct the least squares estimator $\widehat{\theta}_\ell$ using only the observations of this phase where $\widehat{\theta}_\ell$ is defined in (5.7). Note that $\widehat{\theta}_\ell$ is also rotated following (5.4).
  Eliminate arms such that $\underline{\mathcal{W}}_{\ell+1} \leftarrow \underline{\mathcal{W}}_\ell \backslash \{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell : \max_{\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell} \langle \underline{\mathbf{w}}' - \underline{\mathbf{w}}, \widehat{\theta}_\ell\rangle > 2\epsilon_\ell\}$
  $\ell \leftarrow \ell + 1$
Output the arm in $\underline{\mathcal{W}}_\ell$ and reshape to get the $\widehat{\mathbf{x}}_*$ and $\widehat{\mathbf{z}}_*$

Sample Complexity Analysis of Single-Task GOBLIN

We now analyze the sample complexity of GOBLIN in the single-task setting through the following theorem.

Theorem 1. (***informal***) *With probability at least* $1-\delta$, *GOBLIN returns the best arms* $\mathbf{x}_*, \mathbf{z}_*$, *and the number of samples used is bounded by* $\widetilde{O}\left(\frac{(d_1+d_2)r}{\Delta^2} + \frac{\sqrt{d_1 d_2 r}}{S_r}\right)$.

Discussion 1. In Theorem 1 the first quantity is the number of samples needed to identify the best arms $\mathbf{x}_*, \mathbf{z}_*$ while the second quantity is the number of samples to learn $\mathbf{\Theta}_*$ (which is required to find the best arms). Note that the magnitude of S_r would be free of d_1, d_2 since $\mathbf{\Theta}_*$ contains only r nonzero singular values and $\|\mathbf{\Theta}_*\| \leqslant 1$, and hence we assume that $S_r = \Theta(1/\sqrt{r})$ (Kang et al., 2022). So the sample complexity of single-task GOBLIN scales as $\widetilde{O}(\frac{(d_1+d_2)r}{\Delta^2})$. However, if one runs RAGE (Fiez et al., 2019) on the arms in $\mathcal{X}, \mathcal{Z}$ then the sample complexity will scale as $\widetilde{O}(\frac{d_1 d_2}{\Delta^2})$.

Proof (Overview) of Theorem 1: Step 1 (Subspace estimation in high dimension): We denote the vectorized arms in high dimension as $\overline{\mathbf{w}} \in \overline{\mathcal{W}}$. We run the E-optimal design to sample the arms in $\overline{\mathcal{W}}$. Note that this E-optimal design satisfies the distribution assumption of Kang et al. (2022) which enables us to apply the Theorem D.3 in Section D.1. This leads to $\|\widehat{\mathbf{\Theta}}_\ell - \mathbf{\Theta}_*\|_F^2 \leqslant \frac{C_1 d_1 d_2 r \log(2(d_1+d_2)/\delta)}{\tau_\ell^E}$ for some $C_1 > 0$. Also, note that in the first stage of the ℓ-th phase by setting $\tau_\ell^E = \frac{\sqrt{8 d_1 d_2 r \log(4\ell^2|\mathcal{W}|/\delta_\ell)}}{S_r}$ and sampling each arm $\overline{\mathbf{w}} \in \overline{\mathcal{W}}$ exactly $\lceil \widehat{b}_{\ell,\overline{\mathbf{w}}}^E \tau_\ell^E \rceil$ times we are guaranteed that $\|\theta^*_{k+1:p}\|_2 = O(d_1 d_2 r/\tau_\ell^E)$. Summing up over $\ell = 1$ to $\lceil \log_2(4\Delta^{-1}) \rceil$ we get that the total sample complexity of the first stage is bounded by $\widetilde{O}(\sqrt{d_1 d_2 r}/S_r)$.

Step 2 (Effective dimension for rotated arms): We rotate the arms $\overline{\mathbf{w}} \in \overline{\mathcal{W}}$ in high dimension to get the rotated arms $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell$ in step 10 of Algorithm 5. Then we show that the effective dimension of $\underline{\mathbf{w}}$ scales $8k \log(1 + \tau_{\ell-1}^G/\lambda)$ when $\lambda_\ell^\perp = \frac{\tau_{\ell-1}^G}{8k \log(1+\tau_{\ell-1}^G/\lambda)}$ in Theorem D.11 of Sec-

tion D.4. Note that this requires a different proof technique than Valko et al. (2014) where the budget n is given apriori and effective dimension scales with $\log(n)$. This step also diverges from the pure exploration proof technique of Fiez et al. (2019); Katz-Samuels et al. (2020) as there is no parameter $\lambda_\ell^\perp$ to control during phase ℓ, and the effective dimensions in those papers do not depend on phase length.

Step 3 (Bounded Support): For any phase ℓ, we can show that $1 \leqslant \rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \leqslant p/\gamma_{\mathcal{Y}}^2$ where, $\gamma_{\mathcal{Y}} = \max\{c > 0 : c\mathcal{Y} \subset \text{conv}(\underline{\mathcal{W}} \cup -\underline{\mathcal{W}})\}$ is the gauge norm of $\mathcal{Y}$ (Rockafellar, 2015). Note that this is a worst-case dependence when $\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))$ scales with p. Substituting this value of $\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))$ in the definition of $\lambda_\ell^\perp$ we can show that $\boldsymbol{\Lambda}_\ell$ does not depend on $\underline{\mathbf{w}}$ or $\mathbf{y} = \underline{\mathbf{w}} - \underline{\mathbf{w}}'$. Then following Theorem 21.1 in Lattimore and Szepesvári (2020a) we can show that the G-optimal design $\widehat{b}_\ell^G$ is equivalent to D-optimal design $\widehat{b}_\ell^D = \arg\max_{\mathbf{b}} \log \frac{|\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell|}{|\boldsymbol{\Lambda}_\ell|}$. Then using Frank-Wolfe algorithm (Jamieson and Jain, 2022) we can show the support $\widehat{b}_\ell^G$ or equivalently $\widehat{b}_\ell^D$ is bounded by at most $\frac{8k\log(1+\tau_{\ell-1}^G/\lambda)(8k\log(1+\tau_{\ell-1}^G/\lambda)+1)}{2}$. This is shown in Theorem D.13 (Section D.4).

Step 4 (Phase length and Elimination): Using the Theorem D.13, concentration Theorem D.9, and using the log determinant inequality in Theorem D.11 and Proposition 1 (Section D.4) we show that the phase length in the second stage is given by $\tau_\ell^G = \lceil \frac{8B_*^\ell \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell))\log(2|\mathcal{W}|/\delta)}{(\mathbf{x}^\top(\widehat{\boldsymbol{\theta}}_\ell - \boldsymbol{\theta}^*))^2} \rceil$. This is discussed in Discussion 3 (Section D.4). We show in Theorem D.14 (Section D.4) that setting this phase length and sampling each active arm in $\underline{\mathcal{W}}_\ell$ exactly $\lceil \widehat{b}_{\ell,\underline{\mathbf{w}}} \tau_\ell^G \rceil$ times results in the elimination of sub-optimal actions with high probability.

Step 5 (Total Samples): We first show that the total samples in the second phase are bounded by $O(\frac{k}{\gamma_{\mathcal{Y}}^2}\log(\frac{k\log_2(\Delta^{-1})|\underline{\mathcal{W}}|}{\delta})\lceil\log_2(\Delta^{-1})\rceil)$ where the effective dimension $k = (d_1 + d_2)r$. Finally, we combine the total samples of phase ℓ as $(\tau_\ell^E + \tau_\ell^G)$. The final sample complexity is given by summing over all phases from $\ell = 1$ to $\lceil \log_2(4\Delta^{-1}) \rceil$. The claim of the

theorem follows by noting $\widetilde{O}(k/\gamma_y^2) \leqslant \widetilde{O}(k/\Delta^2)$.

5.3 Multi-task Representation Learning

In this section, we extend GOBLIN to multi-task representation learning for the bilinear bandit setting. In the multi-task setting, we now have M tasks, where each task $m \in [M]$ has a reward model stated in (5.1). The learning proceeds as follows: At each round $t = 1, 2, \cdots$, for each task $m \in [M]$, the learner selects a left and right action $\mathbf{x}_{m,t} \in \mathcal{X}$ and $\mathbf{z}_{m,t} \in \mathcal{Z}$. After the player commits the batch of actions for each task $\{\mathbf{x}_{m,t}, \mathbf{z}_{m,t} : m \in [M]\}$, it receives the batch of rewards $\{r_{m,t} : m \in [M]\}$. Finally recall that the goal is to identify the optimal left and right arms $\mathbf{x}_{m,*}, \mathbf{z}_{m,*}$ for each task m with a minimum number of samples. We now state the following assumptions to enable representation learning across tasks.

Assumption 7. (***Low-rank Tasks***) *We assume that the hidden parameter* $\mathbf{\Theta}_{m,*}$ *for all the* $m \in [M]$ *have a decomposition* $\mathbf{\Theta}_{m,*} = \mathbf{B}_1 \mathbf{S}_{m,*} \mathbf{B}_2^\top$ *and each* $\mathbf{S}_{m,*}$ *has rank* r.

This is similar to the assumptions in Yang et al. (2020, 2022a); Du et al. (2023) ensuring the feature extractors are shared across tasks in the bilinear bandit setting.

Assumption 8. (***Diverse Tasks***) *We assume that* $\sigma_{\min}(\frac{1}{M}\sum_{m=1}^{M} \mathbf{\Theta}_{m,*}) \geqslant \frac{c_0}{S_r}$, *for some* $c_0 > 0$, S_r *is the* r*-th largest singular value of* $\mathbf{\Theta}_{m,*}$ *and* $\sigma_{\min}(\mathbf{A})$ *denotes the minimum eigenvalue of matrix* $\mathbf{A}$.

This assumption is similar to the diverse tasks assumption of Yang et al. (2020, 2022a); Tripuraneni et al. (2021); Du et al. (2023) and ensures the possibility of recovering the feature extractors $\mathbf{B}_1$ and $\mathbf{B}_2$ shared across tasks.

Our extension of GOBLIN to the multi-task setting is now a phase-based, *three-stage* arm elimination algorithm. In GOBLIN each phase

$\ell = 1, 2, \ldots$ consists of three stages; the first stage for estimation of feature extractors $\mathbf{B}_1, \mathbf{B}_2$, which runs for τ_ℓ^{E} rounds, the second stage for estimation of $\mathbf{S}_{m,*}$ which runs for $\sum_m \widetilde{\tau}_{m,\ell}^{\mathsf{E}}$ rounds, and a third stage of pure exploration with rotated arms that runs for $\sum_m \tau_{m,\ell}^{\mathsf{G}}$ rounds. We will define $\tau_{m,\ell}^{\mathsf{E}}$ in Section 5.3, $\widetilde{\tau}_{m,\ell}^{\mathsf{E}}$ in Section 5.3, while the rotated arms and $\tau_{m,\ell}^{\mathsf{G}}$ are defined in Section 5.3. At the end of every phase, GOBLIN eliminates sub-optimal arms to build the active set for the next phase and stops when only the optimal left and right arms are remaining. Now we discuss the individual stages that occur at every phase $\ell = 1, 2, \ldots$ for multi-task GOBLIN.

Estimating Feature Extractors $\mathbf{B}_1$ and $\mathbf{B}_2$ (Stage 1 of Phase ℓ)

In the first stage of phase ℓ, GOBLIN leverages the batch of rewards $\{r_{m,t} : m \in [M]\}$ at every round t from M tasks to learn the feature extractors $\mathbf{B}_1$ and $\mathbf{B}_2$. To do this, GOBLIN first vectorizes the $\mathbf{x} \in \mathcal{X}, \mathbf{z} \in \mathcal{Z}$ into a new vector $\overline{\mathbf{w}} = [\mathbf{x}_{1:d_1}; \mathbf{z}_{1:d_2}] \in \overline{\mathcal{W}}_m$ and then solves the E-optimal design in step 3 of Algorithm 6. Similar to the single-task setting (Section 5.2) GOBLIN samples each $\overline{\mathbf{w}} \in \overline{\mathcal{W}}_m$ for $\lceil \tau_\ell^{\mathsf{E}} \mathbf{b}_{\ell,\overline{\mathbf{w}}}^{\mathsf{E}} \rceil$ times for each task m, where $\tau_\ell^{\mathsf{E}} = \widetilde{O}(\sqrt{d_1 d_2 r}/S_r)$ and $\mathbf{b}_{\ell,\overline{\mathbf{w}}}^{\mathsf{E}}$ is the solution to E-optimal design on $\overline{\mathbf{w}}$. Let the sampled arms for each task m at round s be denoted by $\mathbf{x}_{m,s}$, $\mathbf{z}_{m,s}$ which is obtained after reshaping $\overline{\mathbf{w}}_s$. Then it builds the estimator $\widehat{Z}_\ell$ as follows:

$$\widehat{Z}_\ell = \underset{\Theta \in \mathbb{R}^{d_1 \times d_2}}{\arg\min}\, L_\ell(\Theta) + \gamma_\ell \|\Theta\|_{\text{nuc}},$$

$$L_\ell(\Theta) = \langle \Theta, \Theta \rangle - \frac{2}{M\tau_\ell^{\mathsf{E}}} \sum_{m=1}^{M} \sum_{s=1}^{\tau_\ell^{\mathsf{E}}} \langle \widetilde{\psi}_\nu(r_{m,s} \cdot Q(\mathbf{x}_{m,s}\mathbf{z}_{m,s}^\top)), \Theta \rangle \tag{5.8}$$

where $\widetilde{\psi}$ is defined in Theorem D.7 and score function Q is defined in Theorem D.5. Then it performs SVD decomposition on $\widehat{Z}_\ell$, and let $\widehat{B}_1$, $\widehat{B}_2$

be the top-k_1 and top-k_2 left and right singular vectors of $\widehat{Z}_\ell$ respectively. These are the estimation of the feature extractors $\mathbf{B}_1$ and $\mathbf{B}_2$.

Estimating Hidden Parameter $\mathbf{S}_{m,*}$ per Task (Stage 2 of phase ℓ)

In the second stage of phase ℓ, the goal is to recover the hidden parameter $\mathbf{S}_{m,*}$ for each task m. GOBLIN proceeds as follows: First, let $\widetilde{\mathbf{g}}_m = \mathbf{x}^\top \widehat{B}_{1,\ell}$ and $\widetilde{\mathbf{v}}_m = \mathbf{z}^\top \widehat{B}_{2,\ell}$ be the latent left and right arm respectively for each m. Then GOBLIN defines the vector $\widetilde{\mathbf{w}} = [\widetilde{\mathbf{g}}_m; \widetilde{\mathbf{v}}_m] \in \widetilde{\mathcal{W}}_m$ and then solves the E-optimal design in step 11 of Algorithm 6. It then samples for each task m, the latent arm $\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}_m$ for $\lceil \widetilde{\tau}^E_{m,\ell} \widetilde{\mathbf{b}}^E_{m,\ell,\widetilde{\mathbf{w}}} \rceil$ times, where $\widetilde{\tau}^E_{m,\ell} \coloneqq \widetilde{O}(\sqrt{k_1 k_2 r}/S_r)$ and $\widetilde{\mathbf{b}}^E_{m,\ell,\widetilde{\mathbf{w}}}$ is the solution to E-optimal design on $\widetilde{\mathbf{w}}$. Then it builds estimator $\widehat{\mathbf{S}}_{m,\ell}$ for each task m in step 12 as follows:

$$\widehat{\mathbf{S}}_{m,\ell} = \underset{\mathbf{\Theta} \in \mathbb{R}^{k_1 \times k_2}}{\arg\min}\, L'_\ell(\mathbf{\Theta}) + \gamma_\ell \|\mathbf{\Theta}\|_{\text{nuc}},$$

$$L'_\ell(\mathbf{\Theta}) = \langle \mathbf{\Theta}, \mathbf{\Theta} \rangle - \frac{2}{\widetilde{\tau}^E_{m,\ell}} \sum_{s=1}^{\widetilde{\tau}^E_{m,\ell}} \langle \widetilde{\psi}_\nu(r_{m,s} \cdot Q(\widetilde{\mathbf{g}}_{m,s} \widetilde{\mathbf{v}}^\top_{m,s})), \mathbf{\Theta} \rangle \tag{5.9}$$

Once GOBLIN recovers the $\widehat{\mathbf{S}}_{m,\ell}$ for each task m it has reduced the $d_1 d_2$ bilinear bandit to a $k_1 k_2$ dimension bilinear bandit where the left and right arms are $\widetilde{\mathbf{g}}_m \in \mathcal{G}_m$, $\widetilde{\mathbf{v}}_m \in \mathcal{V}_m$ respectively.

Optimal Design for Rotated Arms per Task (Stage 3 of phase ℓ)

In the third stage of phase ℓ, similar to Algorithm 5, the multi-task GOBLIN defines the rotated arm set $\underline{\mathcal{G}}_m, \underline{\mathcal{V}}_m$ for each task m for these $k_1 k_2$ dimensional bilinear bandits. Let the SVD of $\widehat{\mathbf{S}}_{m,\ell} = \widehat{\mathbf{U}}_{m,\ell} \widehat{\mathbf{D}}_{m,\ell} \widehat{\mathbf{V}}^\top_{m,\ell}$. Define $\widehat{\mathbf{H}}_{m,\ell} = [\widehat{\mathbf{U}}_{m,\ell} \widehat{\mathbf{U}}^\perp_{m,\ell}]^\top \widehat{\mathbf{S}}_{m,\ell} [\widehat{\mathbf{V}}_{m,\ell} \widehat{\mathbf{V}}^\perp_{m,\ell}]$. Then define the vectorized arm set so

that the last $(k_1 - r) \cdot (k_2 - r)$ components are from the complementary subspaces as follows:

$$\begin{aligned}
\underline{\mathcal{W}}_{m,\ell} = \Big\{ \big[\mathbf{vec}\left(\widetilde{\mathbf{g}}_{m,1:r}\widetilde{\mathbf{v}}_{m,1:r}^\top\right); \mathbf{vec}\left(\widetilde{\mathbf{g}}_{m,r+1:k_1}\widetilde{\mathbf{v}}_{m,1:r}^\top\right); \mathbf{vec}\left(\widetilde{\mathbf{g}}_{m,1:r}\widetilde{\mathbf{v}}_{m,r+1:k_2}^\top\right); \\
\mathbf{vec}\left(\widetilde{\mathbf{g}}_{m,r+1:k_1}\widetilde{\mathbf{v}}_{m,r+1:k_2}^\top\right) \big] \Big\} \\
\widehat{\theta}_{m,\ell,1:k} = [\mathbf{vec}(\widehat{\mathbf{H}}_{m,\ell,1:r,1:r}); \mathbf{vec}(\widehat{\mathbf{H}}_{m,\ell,r+1:k_1,1:r}); \mathbf{vec}(\widehat{\mathbf{H}}_{m,\ell,1:r,r+1:k_2})], \\
\widehat{\theta}_{m,\ell,k+1:p} = \mathbf{vec}(\widehat{\mathbf{H}}_{m,\ell,r+1:k_1,r+1:k_2}).
\end{aligned} \tag{5.10}$$

This is shown in step 14 of Algorithm 6. Now we proceed similarly to Section 5.2. We construct a per-task optimal design for the rotated arm set $\underline{\mathcal{V}}_m, \underline{\mathcal{G}}_m$ and define the $\underline{\mathbf{w}} = [\widetilde{\mathbf{g}}_{m,1:d_1}; \widetilde{\mathbf{v}}_{m,1:d_2}]$ and $\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}_m$ where $\widetilde{\mathbf{g}}_m \in \underline{\mathcal{G}}_m$ and $\widetilde{\mathbf{v}}_m \in \underline{\mathcal{V}}_m$ respectively. Following (5.5) we know that to minimize the sample complexity for the m-th bilinear bandit we need to sample according to G-optimal design

$$\widehat{b}^G_{m,\ell} = \underset{\mathbf{b}_{m,\underline{\mathbf{w}}}}{\arg\min} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}' \in \underline{\mathcal{W}}_{m,\ell}} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_m} \mathbf{b}_{m,\underline{\mathbf{w}}} \underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \mathbf{\Lambda}_{m,\ell}/n)^{-1}} \tag{5.11}$$

Then GOBLIN runs G-optimal design on the arm set $\underline{\mathcal{W}}_{m,\ell}$ following the (5.11) and then samples each $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_{m,\ell}$ for $\lceil \tau^G_{m,\ell} \widehat{b}^G_{m,\ell,\underline{\mathbf{w}}} \rceil$ times where $\widehat{b}^G_{m,\ell,\underline{\mathbf{w}}}$ is the solution to the G-optimal design, and τ^G_ℓ is defined in step 17 of Algorithm 6. So the total length of phase ℓ, combining stages 1, 2 and 3 is $(\tau^E_\ell + \sum_m \widetilde{\tau}^E_{m,\ell} + \sum_m \tau^G_{m,\ell})$ rounds. Observe that the stage 1 and 2 design is on the whole arm set $\overline{\mathcal{W}}$, $\widetilde{\mathcal{W}}_m$ whereas the stage 3 design is on the refined active set $\underline{\mathcal{W}}_{m,\ell}$. Let at the stage 3 of ℓ-th phase the actions sampled be denoted by the matrix $\underline{\mathbf{W}}_{m,\ell} \in \mathbb{R}^{\tau^G_{m,\ell} \times k_1 k_2}$ and observed rewards $\mathbf{r}_m \in \mathbb{R}^{\tau^G_{m,\ell} \times k_1 k_2}$. Define the positive diagonal matrix $\mathbf{\Lambda}_{m,\ell}$ according to (5.6) but set $p = k_1 k_2$ and $k = (k_1 + k_2)r$. Then similar to Section 5.2 we can build for each task m only from the observations from this phase

$$\widehat{\theta}_{m,\ell} = \arg\min_\theta \tfrac{1}{2}\|\underline{\mathbf{W}}_{m,\ell}\theta - \mathbf{r}_m\|_2^2 + \tfrac{1}{2}\|\theta\|^2_{\mathbf{\Lambda}_{m,\ell}} \tag{5.12}$$

Finally GOBLIN eliminates the sub-optimal arms using the estimator $\widehat{\theta}_{m,\ell}$ to build the next phase active set $\underline{\mathcal{W}}_{m,\ell}$ and stops when $|\underline{\mathcal{W}}_{m,\ell}| = 1$. The full pseudo-code is given in Algorithm 6.

Sample Complexity analysis of Multi-task GOBLIN

We now present the sample complexity of GOBLIN for the multi-task setting.

Theorem 2. **(*informal*)** *With probability at least* $1-\delta$, *GOBLIN returns the best arms* $\mathbf{x}_{m,*}$, $\mathbf{z}_{m,*}$ *for each task* m, *and the total number of samples is bounded by* $\widetilde{O}\left(\frac{M(k_1+k_2)r}{\Delta^2} + \frac{M\sqrt{k_1k_2r}}{S_r} + \frac{\sqrt{d_1d_2r}}{S_r}\right)$.

Discussion 2. In Theorem 2 the first quantity is the sample complexity to identify the best arms $\mathbf{x}_{m,*}$, $\mathbf{z}_{m,*}$ and the second quantity is the number of samples to learn $\mathbf{S}_{m,*}$ for each task m. This is required to rotate the arms to reach the effective dimension of $(k_1+k_2)r$. Finally, the third quantity is the number of samples needed to learn $\mathbf{\Theta}_{m,*}$ (which in turn is used to estimate the feature extractors $\mathbf{B}_1$ and $\mathbf{B}_2$ to learn the $\mathbf{S}_{m,*}$). Again we assume that $S_r = \Theta(1/\sqrt{r})$ (Kang et al., 2022). So the sample complexity of multi-task GOBLIN scales as $\widetilde{O}(M(k_1+k_2)r/\Delta^2)$. However, if one runs DouExpDes (Du et al., 2023) then the sample complexity will scale as $\widetilde{O}(M(k_1k_2)/\Delta^2)$ which is worse than GOBLINwhen $r \ll k_1$ or k_2.

Proof (Overview) of Theorem 2: Step 1 (Subspace estimation in high dimension): The first steps diverge from the proof technique of Theorem 1. We now build the average estimator $\widehat{Z}_\ell$ to estimate the quantity $\mathbf{Z}_* = \frac{1}{M}\sum_{m=1}^{M}\mathbf{\Theta}_{*,m}$ using (5.8). This requires us to modify the Theorem D.3 in Section D.1 and apply Stein's lemma (Theorem D.1) to get a bound of $\|\widehat{Z}_\ell - \mathbf{Z}_*\|_F^2 \leqslant \frac{C_1 d_1 d_2 r \log(2(d_1+d_2)/\delta)}{\tau_\ell^E}$ for some $C_1 > 0$. This is shown in Theorem D.18 in Section D.5. Summing up over $\ell = 1$ to $\lceil \log_2(4\Delta^{-1}) \rceil$ we get that the total samples complexity of the first stage is bounded by $\widetilde{O}(\sqrt{d_1d_2r}/S_r)$.

Algorithm 6 G-Optimal Design for Bilinear Bandits (GOBLIN) for multi-task setting

Input: arm set $\mathcal{X}, \mathcal{Z}$, confidence δ, rank r of $\mathbf{\Theta}_*$, spectral bound S_r of $\mathbf{\Theta}_*$, S, $S^{\perp}_{m,\ell} = \frac{8k_1k_2r}{\widetilde{\tau}^{E}_{m,\ell} S_r^2} \log(\frac{k_1+k_2}{\delta_\ell})$, $\lambda, \lambda^{\perp}_{m,\ell} = \frac{\tau^{G}_{m,\ell-1}}{(8(k_1+k_2)r\log(1+\tau^{G}_{m,\ell-1}/\lambda))}$. Let $p = k_1k_2$, $k = (k_1+k_2)r$.

Let $\underline{\mathcal{W}}_{m,1} \leftarrow \underline{\mathcal{W}}_m$, $\ell \leftarrow 1$, $\tau^{G}_0 = \log(4\ell^2|\mathcal{X}|/\delta)$. Define $\mathbf{\Lambda}_{m,\ell}$ as in (5.6), $B^{\ell}_{m,*} := (8\sqrt{\lambda}S + \sqrt{\lambda^{\perp}_{m,\ell} S^{\perp}_{m,\ell}})$

Define arm $\overline{\mathbf{w}} = [\mathbf{x}_{m,1:d_1}; \mathbf{z}_{m,1:d_2}]$ and $\overline{\mathbf{w}} \in \overline{\mathcal{W}}_m$. Let $\tau^{E}_\ell = \frac{\sqrt{8d_1d_2r\log(4\ell^2|\mathcal{W}|/\delta_\ell)}}{S_r}$. Let E-optimal design be $\mathbf{b}^{E}_\ell = \arg\min_{\mathbf{b}\in\triangle_{\overline{\mathcal{W}}_m}} \left\|(\sum_{\overline{\mathbf{w}}\in\overline{\mathcal{W}}_m} \mathbf{b}_{\overline{\mathbf{w}}} \overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top)^{-1}\right\|$.

while $\exists m \in [M], \left|\underline{\mathcal{W}}_{m,\ell}\right| > 1$ **do**

  $\epsilon_\ell = 2^{-\ell}$, $\delta_\ell = \delta/\ell^2$.

  (Stage 1:) Explore the Low-Rank Subspace

  Pull arm $\overline{\mathbf{w}} \in \overline{\mathcal{W}}_m$ exactly $\lceil \widehat{b}^{E}_{\ell,\overline{\mathbf{w}}} \tau^{E}_\ell \rceil$ times for each task m and observe rewards $\{r_{m,t}\}_{t=1}^{\tau^{E}_\ell}$.

  Compute $\widehat{Z}_\ell$ using (5.8).

  (Stage 2:) Build $\widehat{\mathbf{S}}_{m,\ell}$ for each task m

  Let $\widehat{B}_{1,\ell}$, $\widehat{B}_{2,\ell}$ be the top-k_1 left and top-k_2 right singular vectors of $\widehat{Z}_\ell$ respectively. Build $\widetilde{\mathbf{g}}_m = \mathbf{x}^\top \widehat{B}_{1,\ell}$ and $\widetilde{\mathbf{v}}_m = \mathbf{z}^\top \widehat{B}_{2,\ell}$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ for each m.

  Define a vectorized arm $\widetilde{\mathbf{w}} = [\widetilde{\mathbf{g}}_{m,1:k_1}; \widetilde{\mathbf{v}}_{m,1:k_2}]$ and $\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}_m$ for each m. Let $\widetilde{\tau}^{E}_{m,\ell} = \frac{\sqrt{8k_1k_2r\log(4\ell^2|\mathcal{W}|/\delta_\ell)}}{S_r}$, and $\widetilde{\mathbf{b}}^{E}_{m,\ell} = \arg\min_{\mathbf{b}_m\in\triangle_{\widetilde{\mathcal{W}}_m}} \left\|\left(\sum_{\widetilde{\mathbf{w}}\in\widetilde{\mathcal{W}}_m} \mathbf{b}_{m,\widetilde{\mathbf{w}}} \widetilde{\mathbf{w}}\,\widetilde{\mathbf{w}}^\top\right)^{-1}\right\|$.

  Pull arm $\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}_m$ exactly $\left\lceil \widetilde{\mathbf{b}}^{E}_{m,\ell,\widetilde{\mathbf{w}}} \widetilde{\tau}^{E}_{m,\ell} \right\rceil$ times and observe rewards $r_{m,t}$, for $t = 1, \ldots, \widetilde{\tau}^{E}_{m,\ell}$, for each task m. Then compute $\widehat{\mathbf{S}}_{m,\ell}$ using (5.9) for each m.

  (Stage 3:) Reduction to low dimensional linear bandits for each task m

  SVD of $\widehat{\mathbf{S}}_{m,\ell} = \widehat{\mathbf{U}}_{m,\ell}\widehat{\mathbf{D}}_{m,\ell}\widehat{\mathbf{V}}^\top_{m,\ell}$. Rotate arms in active set $\underline{\mathcal{W}}_{m,\ell-1}$ to build $\underline{\mathcal{W}}_{m,\ell}$ using (5.10).

  Let $\widehat{b}^{G}_{m,\ell} = \arg\min_{\mathbf{b}_{m,\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_{m,\ell}} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}}_m\in\underline{\mathcal{W}}_m} \mathbf{b}_{m,\underline{\mathbf{w}}_m} \underline{\mathbf{w}}_m^\top + \mathbf{\Lambda}_{m,\ell}/n)^{-1}}$.

  Let $\rho^{G}(\mathcal{Y}(\underline{\mathcal{W}}_{m,\ell})) = \min_{\mathbf{b}_{m,\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_{m,\ell}} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_m} \mathbf{b}_{m,\underline{\mathbf{w}}} \underline{\mathbf{w}}^\top + \frac{\mathbf{\Lambda}_{m,\ell}}{n})^{-1}}$.

  Set $\tau^{G}_{m,\ell} = \frac{64B^{\ell}_{m,*}\rho^{G}(\mathcal{Y}(\mathcal{W}_{m,\ell}))\log(4\ell^2|\mathcal{W}_m|/\delta_\ell)}{\epsilon^2_\ell}$. Then pull arm $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_m$ for each task m exactly $\lceil \widehat{b}_{m,\ell,\underline{\mathbf{w}}} \tau^{G}_{m,\ell} \rceil$ times and construct the least squares estimator $\widehat{\theta}_{m,\ell}$ using only the observations of this phase where $\widehat{\theta}_{m,\ell}$ is defined in (5.12).

  Eliminate arms such that $\underline{\mathcal{W}}_{m,\ell+1} \leftarrow \underline{\mathcal{W}}_{m,\ell} \setminus \left\{\underline{\mathbf{w}}_m \in \underline{\mathcal{W}}_{m,\ell} : \max_{\underline{\mathbf{w}}'_m\in\underline{\mathcal{W}}_{m,\ell}} \left\langle \underline{\mathbf{w}}'_m - \underline{\mathbf{w}}_m, \widehat{\theta}_{m,\ell} \right\rangle > 2\epsilon_{m,\ell}\right\}$

  $\ell \leftarrow \ell + 1$

Output the arm in $\underline{\mathcal{W}}_{m,\ell}$ and reshape to get the $\widehat{\mathbf{x}}_{m,*}$ and $\widehat{\mathbf{z}}_{m,*}$ for each task m.

Step 2 (Estimation of left and right feature extractors): Now using the estimator in (5.8) we get a good estimation of the feature extractors $\mathbf{B}_1$ and $\mathbf{B}_2$. Let $\widehat{\mathrm{B}}_{1,\ell}$, $\widehat{\mathrm{B}}_{2,\ell}$ be the top-k_1 left and top-k_2 right singular vectors of $\widehat{Z}_\ell$ respectively. Then using the Davis-Kahan sin θ Theorem (Bhatia, 2013) in Theorem D.20, D.21 (Section D.5) we have $\|(\widehat{\mathrm{B}}_{1,\ell}^{\perp})^\top \mathbf{B}_1\|, \|(\widehat{\mathrm{B}}_{2,\ell}^{\perp})^\top \mathbf{B}_2\| \leqslant \widetilde{O}(\sqrt{(d_1+d_2)r/M\tau_\ell^E})$.

Step 3 (Estimation of $\widehat{\mathbf{S}}_{m,\ell}$ in low dimension): Now we estimate the quantity $\widehat{\mathbf{S}}_{m,\ell} \in \mathbb{R}^{k_1\times k_2}$ for each task m. To do this we first build the latent arms $\widetilde{\mathbf{g}}_m = \mathbf{x}^\top \widehat{\mathbf{U}}_\ell$ and $\widetilde{\mathbf{v}}_m = \mathbf{z}^\top \widehat{\mathbf{V}}_\ell$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$ for each m, and sample them following the E-optimal design in step 12 of Algorithm 6. We also show in Theorem D.22 (Section D.5) that $\sigma_{\min}(\sum_{\widetilde{\mathbf{w}}\in\widetilde{\mathcal{W}}} \mathbf{b}_{\widetilde{\mathbf{w}}}\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^\top) > 0$ which enables us to sample following E-optimal design. Then use the estimator in (5.9). Then in Theorem D.25 we show that $\|\widehat{\mathbf{S}}_{m,\ell} - \mu^*\mathbf{S}_{m,*}\|_F^2 \leqslant C_1 k_1 k_2 r \log\left(\frac{2(k_1+k_2)}{\delta_\ell}\right)/\tau_{m,\ell}^E$ holds with probability greater than $(1-\delta)$. Also, note that in the second phase by setting $\widetilde{\tau}_{m,\ell}^E = \sqrt{8k_1k_2r\log(4\ell^2|\mathcal{W}|/\delta_\ell)}/S_r$ and sampling each arm $\overline{\mathbf{w}} \in \overline{\mathcal{W}}$ exactly $\lceil \widehat{b}_{\ell,\overline{\mathbf{w}}}^E \widetilde{\tau}_{m,\ell}^E \rceil$ times we are guaranteed that $\|\theta^*_{k+1:p}\|_2 = O(k_1k_2r/\widetilde{\tau}_{m,\ell}^E)$ in the ℓ-th phase. Summing up over $\ell = 1$ to $\lceil \log_2(4\Delta^{-1})\rceil$ across each task M we get that the total samples complexity of the second stage is bounded by $\widetilde{O}(M\sqrt{k_1k_2r}/S_r)$.

Step 4 (Convert to k_1k_2 bilinear bandits): Once GOBLIN recovers $\widehat{\mathbf{S}}_{m,\tau_\ell^E}$ it rotates the arm set following (5.10) to build $\underline{\mathcal{W}}_m$ to get the k_1k_2 bilinear bandits. The rest of the steps follow the same way as in steps 2, 3 and 4 of proof of Theorem 1.

Step 5 (Total Samples): We show the total samples in the third phase are bounded by $O(\frac{k}{\gamma_y^2}\log(\frac{k\log_2(\Delta^{-1})|\underline{\mathcal{W}}|}{\delta})\lceil\log_2(\Delta^{-1})\rceil)$ where the effective dimension $k = (k_1+k_2)r$. The total samples of phase ℓ is given by $\tau_\ell^E + \sum_m(\widetilde{\tau}_{m,\ell}^E + \tau_{m,\ell}^G)$. Finally, we get the total sample complexity by summing over all phases from $\ell = 1$ to $\lceil \log_2(4\Delta^{-1})\rceil$. The claim of the theorem follows by noting $\widetilde{O}(k/\gamma_y^2) \leqslant \widetilde{O}(k/\Delta^2)$.

5.4 Experiments

In this section, we conduct proof-of-concept experiments on both single and multi-task bilinear bandits. In the single-task experiment, we compare against the state-of-the-art RAGE algorithm (Fiez et al., 2019). We show in Figure 5.1 (left) that GOBLIN requires fewer samples than the RAGE with an increasing number of arms. In the multi-task experiment, we compare against the state-of-the-art DouExpDes algorithm (Du et al., 2023). We show in Figure 5.1 (right) that GOBLIN requires fewer samples than DouExpDes with an increasing number of tasks. As experiments are not a central contribution, we defer a fuller description of the experimental set-up to Section D.6.

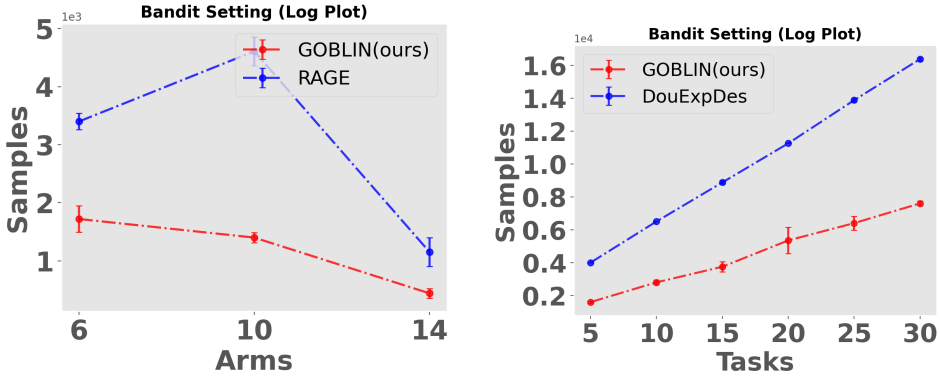


Figure 5.1: (Left) Single-task experiment: results show the number of samples required to identify the optimal action pair for differing numbers of actions. (Right) Multi-task experiment: results show the number of samples required to identify the optimal action pair for varying numbers of tasks. Note the scale of the samples in top left corner of the plots.

5.5 Conclusions and Future Directions

In this paper, we formulated the first pure exploration multi-task representation learning problem. We introduce an algorithm, GOBLIN that achieves a sample complexity bound of $\widetilde{O}((d_1 + d_2)r/\Delta^2)$ which improves upon the $\widetilde{O}((d_1 d_2)/\Delta^2)$ sample complexity of RAGE (Fiez et al., 2019) in a single-task setting. We then extend GOBLIN for multi-task pure exploration bilinear bandit problems by learning latent features which enables sample complexity that scales as $\widetilde{O}(M(k_1 + k_2)r/\Delta^2)$ which improves over the $\widetilde{O}(M(k_1 k_2)/\Delta^2)$ sample complexity of DouExpDes (Du et al., 2023). Our analysis opens an exciting opportunity to analyze representation learning in the kernel and neural bandits (Zhu et al., 2021; Mason et al., 2021). We can leverage the fact that this type of optimal design does not require the arm set to be an ellipsoid (Du et al., 2023) which enables us to extend our analysis to non-linear representations.

6 PRETRAINING DECISION TRANSFORMERS WITH REWARD PREDICTION FOR IN-CONTEXT MULTI-TASK STRUCTURED BANDIT LEARNING

In this paper, we study multi-task bandit learning with the goal of learning an algorithm that discovers and exploits structure in a family of related tasks. In multi-task bandit learning, we have multiple distinct bandit tasks for which we want to learn a policy. Though distinct, the tasks share some structure, which we hope to leverage to speed up learning on new instances in this task family. Traditionally, the study of such structured bandit problems has relied on knowledge of the problem structure like linear bandits (Li et al., 2010; Abbasi-Yadkori et al., 2011; Degenne et al., 2020), bilinear bandits (Jun et al., 2019), hierarchical bandits (Hong et al., 2022a,b), Lipschitz bandits (Bubeck et al., 2008, 2011; Magureanu et al., 2014), other structured bandits settings (Riquelme et al., 2018; Lattimore and Szepesvári, 2019; Dong et al., 2021) and even linear and bilinear multi-task bandit settings (Yang et al., 2022a; Du et al., 2023; Mukherjee et al., 2023b). When structure is unknown an alternative is to adopt sophisticated model classes, such as kernel machines or neural networks, exemplified by kernel or neural bandits (Valko et al., 2013; Chowdhury and Gopalan, 2017; Zhou et al., 2020; Dai et al., 2022). However, these approaches are also costly as they learn complex, nonlinear models from the ground up without any prior data (Justus et al., 2018; Zambaldi et al., 2018).

In this paper, we consider an alternative approach of synthesizing a bandit algorithm from historical data where the data comes from recorded bandit interactions with past instances of our target task family. Concretely, we are given a set of state-action-reward tuples obtained by running some bandit algorithm in various instances from the task family. We then aim to train a transformer (Vaswani et al., 2017) from this data such that it can

learn in-context to solve new task instances. Laskin et al. (2022) consider a similar goal and introduce the Algorithm Distillation (AD) method, however, AD aims to copy the algorithm used in the historical data and thus is limited by the ability of the data collection algorithm. Lee et al. (2023) develop an approach, DPT, that enables learning a transformer that obtains lower regret in-context bandit learning compared to the algorithm used to produce the historical data. However, this approach requires knowledge of the optimal action at each stage of the decision process. In real problems, this assumption is hard to satisfy and we will show that DPT performs poorly when the optimal action is only approximately known. With this past work in mind, the goal of this paper is to answer the question:

Can we learn an in-context bandit learning algorithm that obtains lower regret than the algorithm used to produce the training data without knowledge of the optimal action in each training task?

To answer this question, we introduce a new pre-training methodology, called **Pre**-trained **De**cision **T**ransf**o**rmer with **R**eward Estimation (Pre-DeToR) that obviates the need for knowledge of the optimal action in the in-context data — a piece of information that is often inaccessible. Our key observation is that while the mean rewards of each action change from task to task, certain probabilistic dependencies are persistent across all tasks with a given structure (Yang et al., 2020, 2022a; Mukherjee et al., 2023b). These probabilistic dependencies can be learned from the pretraining data and exploited to better estimate mean rewards and improve performance in a new unknown test task. The nature of the probabilistic dependencies depends on the specific structure of the bandit and can be complex (i.e., higher-order dependencies beyond simple correlations). We propose to use transformer models as a general-purpose architecture to capture the unknown dependencies by training transformers to predict the mean

rewards in each of the given trajectories (Mirchandani et al., 2023; Zhao et al., 2023). The key idea is that transformers have the capacity to discover and exploit complex dependencies in order to predict the rewards of all possible actions in each task from a *small* history of action-reward pairs in a new task. This paper demonstrates how such an approach can achieve lower regret by outperforming state-of-the-art baselines, relying solely on historical data, without the need for any supplementary information like the action features or knowledge of the complex reward models. We also show that the shared actions across the tasks are vital for PreDeToR to exploit the latent structure. We show that PreDeToR learns to adapt, in-context, to novel actions and new tasks as long as the number of new actions is small compared to shared actions across the tasks.

Contributions

1. We introduce a new pre-training procedure of learning the underlying reward structure and a decision algorithm. Moreover, PreDeToR by predicting the next reward for all arms circumvents the issue of requiring access to the optimal (or approximately optimal) action during training time.

2. We demonstrate empirically that this training procedure results in lower regret in a wide series of tasks (such as linear, nonlinear, bilinear, and latent bandits) compared to prior in-context learning algorithms and bandit algorithms with privileged knowledge of the common structure.

3. We also show that our training procedure leverages the shared latent structure and is robust to a small number of new actions introduced both during training and testing time.

4. Finally, we theoretically analyze the generalization ability of Pre-DeToR through the lens of algorithmic stability and new results for the transformer setting.

6.1 Background

In this section, we first introduce our notation and the multi-task, structured bandit setting. We then formalize the in-context bandit learning model studied in Laskin et al. (2022); Lee et al. (2023); Sinii et al. (2023); Lin et al. (2023); Ma et al. (2023); Liu et al. (2023e,b).

Preliminaries

In this paper, we consider the multi-task linear bandit setting (Du et al., 2023; Yang et al., 2020, 2022a). In the multi-task setting, we have a family of related bandit problems that share an action set $\mathcal{A}$ and also a common action feature space $\mathcal{X}$. The actions in $\mathcal{A}$ are indexed by $a = 1, 2, \ldots, A$. The feature of each action is denoted by $\mathbf{x}(a) \in \mathbb{R}^d$ and $d \ll A$. A policy, π, is a probability distribution over the actions.

Define $[n] = \{1, 2, \ldots, n\}$. In a multi-task structured bandit setting the expected reward for each action in each task is assumed to be an unknown function of the hidden parameter and action features (Lattimore and Szepesvári, 2020a; Gupta et al., 2020a). The interaction proceeds iteratively over n rounds for each task $m \in [M]$. At each round $t \in [n]$ for each task $m \in [M]$, the learner selects an action $I_{m,t} \in \mathcal{A}$ and observes the reward $r_{m,t} = f(\mathbf{x}(I_{m,t}), \theta_{m,*}) + \eta_{m,t}$, where $\theta_{m,*} \in \mathbb{R}^d$ is the hidden parameter specific to the task m to be learned by the learner. The function $f(\cdot, \cdot)$ is the unknown reward structure. This can be $f(\mathbf{x}(I_{m,t}), \theta_{m,*}) = \mathbf{x}(I_{m,t})^\top \theta_{m,*}$ for the linear setting or even more complex correlation between features and $\theta_{m,*}$ (Filippi et al., 2010a; Abbasi-Yadkori et al., 2011; Riquelme et al., 2018; Lattimore and Szepesvári, 2019; Dong et al., 2021).

In our paper, we assume that there exist weak demonstrators denoted by π^w. These weak demonstrators are stochastic A-armed bandit algorithms like Upper Confidence Bound (UCB) (Auer et al., 2002; Auer and Ortner, 2010) or Thompson Sampling (Thompson, 1933; Agrawal and Goyal, 2012; Russo et al., 2018; Zhu and Tan, 2020). We refer to these algorithms as weak demonstrators because they do not use knowledge of task structure or arm feature vectors to plan their sampling policy. In contrast to a weak demonstrator, a strong demonstrator, like LinUCB, uses feature vectors and knowledge of task structure to conduct informative exploration. Whereas weak demonstrators always exist, there are many real-world settings with no known strong demonstrator algorithm or where the feature vectors are unobserved and the learner can only use the history of rewards and actions.

In-Context Learning Model

Similar to Lee et al. (2023); Sinii et al. (2023); Lin et al. (2023); Ma et al. (2023); Liu et al. (2023e,b) we assume the in-context learning model. We first discuss the pretraining procedure.

Pretraining: Let $\mathcal{T}_{\text{pre}}$ denote the distribution over tasks m at the time of pretraining. Let $\mathcal{D}_{\text{pre}}$ be the distribution over all possible interactions that the π^w can generate. We first sample a task $m \sim \mathcal{T}_{\text{pre}}$ and then a context $\mathcal{H}_m$ which is a sequence of interactions for n rounds conditioned on the task m such that $\mathcal{H}_m \sim \mathcal{D}_{\text{pre}}(\cdot|m)$. So $\mathcal{H}_m = \{I_{m,t}, r_{m,t}\}_{t=1}^{n}$. We call this dataset $\mathcal{H}_m$ an in-context dataset as it contains the contextual information about the task m. We denote the samples in $\mathcal{H}_m$ till round t as $\mathcal{H}_m^t = \{I_{m,s}, r_{m,s}\}_{s=1}^{t-1}$. This dataset $\mathcal{H}_m$ can be collected in several ways: (1) random interactions within m, (2) demonstrations from an expert, and (3) rollouts of an algorithm. Finally, we train a causal GPT-2 transformer model $\mathbf{T}$ parameterized by $\boldsymbol{\Theta}$ (where $\boldsymbol{\Theta}$ are all transformer parameters) on this dataset $\mathcal{D}_{\text{pre}}$. Specifically, we define $\mathbf{T}_{\boldsymbol{\Theta}}(\cdot \mid \mathcal{H}_m^t)$ as the transformer

model that observes the dataset $\mathcal{H}_m^t$ till round t and then produces a distribution over the actions. Our primary novelty lies in our training procedure which we explain in detail in Section 6.2.

Testing: We now discuss the testing procedure for our setting. Let $\mathcal{T}_{\text{test}}$ denote the distribution over test tasks $m \in [M_{\text{test}}]$ at the time of testing. Let $\mathcal{D}_{\text{test}}$ denote a distribution over all possible interactions that can be generated by π^w during test time. At deployment time, the dataset $\mathcal{H}_m^0 \leftarrow \{\emptyset\}$ is initialized empty. At each round t, an action is sampled from the trained transformer model $I_t \sim \mathbf{T}_{\boldsymbol{\Theta}}(\cdot \mid \mathcal{H}_m^t)$. The sampled action and resulting reward, r_t, are then added to $\mathcal{H}_m^t$ to form $\mathcal{H}_m^{t+1}$ and the process repeats for n total rounds. Finally, note that in this testing phase, the model parameter $\boldsymbol{\Theta}$ is not updated. Finally, the goal of the learner is to minimize cumulative regret for all task $m \in [M_{\text{test}}]$ defined as follows: $\mathbb{E}[R_n] = \frac{1}{M_{\text{test}}} \sum_{m=1}^{M_{\text{test}}} \sum_{t=1}^{n} \max_{a \in \mathcal{A}} f(\mathbf{x}(a), \theta_{m,*}) - f(\mathbf{x}(I_t), \theta_{m,*})$.

Related In-context Learning Algorithms

In this section, we discuss related algorithms for in-context decision-making. For completeness, we describe the DPT and AD training procedure and algorithm now. During training, DPT first samples $m \sim \mathcal{T}_{\text{pre}}$ and then an in-context dataset $\mathcal{H}_m \sim \mathcal{D}_{\text{pre}}(\cdot|, m)$. It adds this $\mathcal{H}_m$ to the training dataset $\mathcal{H}_{\text{train}}$, and repeats to collect M_{pre} such training tasks. For each task m, DPT requires the optimal action $a_{m,*} = \arg\max_a f(\mathbf{x}(m, a), \theta_{m,*})$ where $f(\mathbf{x}(m, a), \theta_{m,*})$ is the expected reward for the action a in task m. Since the optimal action is usually not known in advance, in Section 6.3 we introduce a practical variant of DPT that approximates the optimal action with the best action identified during task interaction. During training DPT minimizes the cross-entropy loss:

$$\mathcal{L}_t^{\text{DPT}} = \text{cross-entropy}(\mathbf{T}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m^t), p(a_{m,*})) \tag{6.1}$$

where $p(a_{m,*}) \in \triangle^{\mathcal{A}}$ is a one-hot vector such that $p(j) = 1$ when $j = a_{m,*}$ and 0 otherwise. This loss is then back-propagated and used to update the model parameter $\boldsymbol{\Theta}$.

During test time evaluation for online setting the DPT selects $I_t \sim \text{softmax}_a^\tau(\mathbf{T}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m^t))$ where we define the $\text{softmax}_a^\tau(\mathbf{x})$ over a A dimensional vector $\mathbf{x} \in \mathbb{R}^A$ as $\text{softmax}_a^\tau(\mathbf{x}(a)) = \exp(\mathbf{x}(a)/\tau)/\sum_{a'=1}^{A}\exp(\mathbf{x}(a')/\tau)$ which produces a distribution over actions weighted by the temperature parameter $\tau > 0$. Therefore this sampling procedure has a high probability of choosing the predicted optimal action as well as induce sufficient exploration. In the online setting, the DPT observes the reward $r_t(I_t)$ which is added to $\mathcal{H}_m^t$. So the $\mathcal{H}_m$ during online testing consists of $\{I_t, r_t\}_{t=1}^n$ collected during testing. This interaction procedure is conducted for each test task $m \in [M_{\text{test}}]$. In the testing phase, the model parameter $\boldsymbol{\Theta}$ is not updated.

An alternative to DPT that does *not* require knowledge of the optimal action is the AD approach (Laskin et al., 2022; Lu et al., 2023). In AD, the learner aims to predict the next action of the demonstrator. So it minimizes the cross-entropy loss as follows:

$$\mathcal{L}_t^{\text{AD}} = \text{cross-entropy}(\mathbf{T}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m^t), p(I_{m,t})) \tag{6.2}$$

where $p(I_{m,t})$ is a one-hot vector such that $p(j) = 1$ when $j = I_{m,t}$ (the true action taken by the demonstrator) and 0 otherwise. At deployment time, AD selects $I_t \sim \text{softmax}_a^\tau(\mathbf{T}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m^t))$. Note that the objective of AD is to match the performance of the demonstrator. In the next section, we introduce a new method that can improve upon the demonstrator without knowledge of the optimal action.

Related Works

In this section, we briefly discuss related works. In-context decision making (Laskin et al., 2022; Lee et al., 2023) has emerged as an attractive alternative in Reinforcement Learning (RL) compared to updating the model parameters after collection of new data (Mnih et al., 2013; François-Lavet et al., 2018). In RL the contextual data takes the form of state-action-reward tuples representing a dataset of interactions with an unknown environment (task). In this paper, we will refer to this as the in-context data. Recall that in many real-world settings, the underlying task can be structured with correlated features, and the reward can be highly non-linear. So specialized bandit algorithms fail to learn in these tasks. To circumvent this issue, a learner can first collect in-context data consisting of just action indices I_t and rewards r_t. Then it can leverage the representation learning capability of deep neural networks to learn a pattern across the in-context data and subsequently derive a near-optimal policy (Lee et al., 2023; Mirchandani et al., 2023). We refer to this learning framework as an in-context decision-making setting.

The in-context decision-making setting of Sinii et al. (2023) also allows changing the action space by learning an embedding over the action space yet also requires the optimal action during training. In contrast we do not require the optimal action as well as show that we can generalize to new actions without learning an embedding over them. Similarly, Lin et al. (2023) study the in-context decision-making setting of Laskin et al. (2022); Lee et al. (2023), but they also require a greedy approximation of the optimal action. The Ma et al. (2023) also studies a similar setting for hierarchical RL where they stitch together sub-optimal trajectories and predict the next action during test time. Similarly, Liu et al. (2023e) studies the in-context decision-making setting to predict action instead of learning a reward correlation from a short horizon setting. In contrast we do not require a greedy approximation of the optimal action, deal with

short horizon setting and changing action sets during training and testing, and predict the estimated means of the actions instead of predicting the optimal action. A survey of the in-context decision-making approaches can be found in Liu et al. (2023b).

In the in-context decision-making setting, the learning model is first trained on supervised input-output examples with the in-context data during training. Then during test time, the model is asked to complete a new input (related to the context provided) without any update to the model parameters (Xie et al., 2021; Min et al., 2022). Motivated by this, Lee et al. (2023) recently proposed the Decision Pretrained Transformers (DPT) that exhibit the following properties: **(1)** During supervised pretraining of DPT, predicting optimal actions alone gives rise to near-optimal decision-making algorithms for unforeseen task during test time. Note that DPT does not update model parameters during test time and, therefore, conducts in-context learning on the unforeseen task. **(2)** DPT improves over the in-context data used to pretrain it by exploiting latent structure. However, DPT either requires the optimal action during training or if it needs to approximate the optimal action. For approximating the optimal action, it requires a large amount of data from the underlying task.

At the same time, learning the underlying data pattern from a few examples during training is becoming more relevant in many domains like chatbot interaction (Madotto et al., 2021; Semnani et al., 2023), recommendation systems, healthcare (Ge et al., 2022; Liu et al., 2023c), etc. This is referred to as few-shot learning. However, most current RL decision-making systems (including in-context learners like DPT) require an enormous amount of data to learn a good policy.

The in-context learning framework is related to the meta-learning framework (Bengio et al., 1990; Schaul and Schmidhuber, 2010). Broadly, these techniques aim to learn the underlying latent shared structure within

the training distribution of tasks, facilitating faster learning of novel tasks during test time. In the context of decision-making and reinforcement learning (RL), there exists a frequent choice regarding the specific 'structure' to be learned, be it the task dynamics (Fu et al., 2016; Nagabandi et al., 2018; Landolfi et al., 2019), a task context identifier (Rakelly et al., 2019; Zintgraf et al., 2019; Liu et al., 2021), or temporally extended skills and options (Perkins and Precup, 1999; Gupta et al., 2018; Jiang et al., 2022).

However, as we noted in the Chapter 6, one can do a greedy approximation of the optimal action from the historical data using a weak demonstrator and a neural network policy (Finn et al., 2017; Rothfuss et al., 2018). Moreover, the in-context framework generally is more agnostic where it learns the policy of the demonstrator (Duan et al., 2016; Wang et al., 2016; Mishra et al., 2017). Note that both DPT-greedy and PreDeToR are different than algorithmic distillation (Laskin et al., 2022; Lu et al., 2023) as they do not distill an existing RL algorithm. moreover, in contrast to DPT-greedy which is trained to predict the optimal action, the PreDeToR is trained to predict the reward for each of the actions. This enables the PreDeToR (similar to DPT-greedy) to show to potentially emergent online and offline strategies at test time that automatically align with the task structure, resembling posterior sampling.

As we discussed in the Chapter 6, in decision-making, RL, and imitation learning the transformer models are trained using autoregressive action prediction (Yang et al., 2023). Similar methods have also been used in Large language models (Vaswani et al., 2017; Roberts et al., 2019). One of the more notable examples is the Decision Transformers (abbreviated as DT) which utilizes a transformer to autoregressively model sequences of actions from offline experience data, conditioned on the achieved return (Chen et al., 2021a; Janner et al., 2021). This approach has also been shown to be effective for multi-task settings (Lee et al., 2022), and multi-task imitation learning with transformers (Reed et al., 2022; Brohan et al.,

2022; Shafiullah et al., 2022). However, the DT methods are not known to improve upon their in-context data, which is the main thrust of this paper (Brandfonbrener et al., 2022; Yang et al., 2022b).

Our work is also closely related to the offline RL setting. In offline RL, the algorithms can formulate a policy from existing data sets of state, action, reward, and next-state interactions. Recently, the idea of pessimism has also been introduced in an offline setting to address the challenge of distribution shift (Kumar et al., 2020; Yu et al., 2021; Liu et al., 2020; Ghasemipour et al., 2022). Another approach to solve this issue is policy regularization (Fujimoto et al., 2019; Kumar et al., 2019; Wu et al., 2019; Siegel et al., 2020; Liu et al., 2019), or reuse data for related task (Li et al., 2020; Mitchell et al., 2021), or additional collection of data along with offline data (Pong et al., 2022). However, all of these approaches still have to take into account the issue of distributional shifts. In contrast PreDeToR and DPT-greedy leverages the decision transformers to avoid these issues. Both of these methods can also be linked to posterior sampling. Such connections between sequence modeling with transformers and posterior sampling have also been made in Chen et al. (2021a); Müller et al. (2021); Lee et al. (2023); Yang et al. (2023).

6.2 Proposed Algorithm PreDeToR

We now introduce our main algorithmic contribution, PreDeToR (which stands for **Pre**-trained **De**cision **T**rans**f**o**r**mer with **R**eward Estimation).

Pre-training Next Reward Prediction

The key idea behind PreDeToR is to leverage the in-context learning ability of transformers to infer the reward of each arm in a given test task. By training this in-context ability on a set of training tasks, the transformer can implicitly learn structure in the task family and exploit this structure to infer rewards without trying every single arm. Thus, in contrast to

DPT and AD that output actions directly, PreDeToR outputs a scalar value reward prediction for each arm. To this effect, we append a linear layer of dimension A on top of a causal GPT2 model, denoted by $\text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m)$, and use a least-squares loss to train the transformer to predict the reward for each action with these outputs. Note that we use $\text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m)$ to denote a reward prediction transformer and $\mathbf{T}_{\boldsymbol{\Theta}}(\cdot|\mathcal{H}_m)$ as the transformer that predicts a distribution over actions (as in DPT and AD). At every round t the transformer predicts the *next reward* for each of the actions $a \in \mathcal{A}$ for the task m based on $\mathcal{H}^t_m = \{I_{m,s}, r_{m,s}\}_{s=1}^{t-1}$. This predicted reward is denoted by $\widehat{r}_{m,t+1}(a)$ for each $a \in \mathcal{A}$.

Loss calculation: For each training task, m, we calculate the loss at each round, t, using the transformer's prediction $\hat{r}_{m,t}(I_{m,t})$ and the actual observed reward $r_{m,t}$ that followed action $I_{m,t}$. We use a least-squares loss function:

$$\mathcal{L}_t = (\widehat{r}_{m,t}(I_{m,t}) - r_{m,t})^2 \tag{6.3}$$

and hence minimizing this loss will minimize the mean squared-error of the transformer's predictions. The loss is calculated using (6.3) and is backpropagated to update the model parameter $\boldsymbol{\Theta}$.

Exploratory Demonstrator: Observe from the loss definition in (6.3) that it is calculated from the observed true reward and action from the dataset $\mathcal{H}_m$. In order for the transformer to learn accurate reward predictions during training, we require that the weak demonstrator is sufficiently exploratory such that it collects $\mathcal{H}_m$ such that $\mathcal{H}_m$ contains some reward $r_{m,t}$ for each action a. We discuss in detail the impact of the demonstrator on PreDeToR (-τ) training in Section E.12.

Deploying PreDeToR

At deployment time, PreDeToR learns in-context to predict the mean reward of each arm on an unseen task and acts greedily with respect to this

prediction. That is, at deployment time, a new task is sampled, $m \sim \mathcal{T}_{\text{test}}$, and the dataset $\mathcal{H}_m^0$ is initialized empty. Then at every round t, PreDeToR chooses $I_t = \arg\max_{a \in \mathcal{A}} \text{TF}^{\mathbf{r}}_{\Theta}(\widehat{r}_{m,t}(a) \mid \mathcal{H}_m^t)$ which is the action with the highest predicted reward and $\widehat{r}_{m,t}(a)$ is the predicted reward of action a. Note that PreDeToR is a greedy policy and thus may fail to conduct sufficient exploration. To remedy this potential limitation, we also introduce a soft variant, PreDeToR-τ that chooses $I_t \sim \text{softmax}_a^{\tau}(\text{TF}^{\mathbf{r}}_{\Theta}(\widehat{\mathbf{r}}_{m,t}(a) \mid \mathcal{H}_m^t))$. For both PreDeToR and PreDeToR-τ, the observed reward $r_t(I_t)$ is added to the dataset $\mathcal{H}_m$ and then used to predict the reward at the next round $t+1$. The full pseudocode of using PreDeToR for online interaction is shown in Algorithm 7. In Section E.14, we discuss how PreDeToR (-τ) can be deployed for offline learning.

6.3 Empirical Study: Non-Linear Structure

Having introduced PreDeToR, we now investigate its performance in diverse bandit settings compared to other in-context learning algorithms. In our first set of experiments, we use a bandit setting with a common non-linear structure across tasks. Ideally, a good learner would leverage the structure, however, we choose the structure such that no existing algorithms are well-suited to the non-linear structure. This setting is thus a good testbed for establishing that in-context learning can discover and exploit common structure. Moreover, each task only consists of a few rounds of interactions. This setting is quite common in recommender settings where user interaction with the system lasts only for a few rounds and has an underlying non-linear structure (Kwon et al., 2022; Tomkins et al., 2020). We show that PreDeToR achieves lower regret than other in-context algorithms for the non-linear structured bandit setting. We study the performance of PreDeToR in the large horizon setting in Section E.6.

Baselines: We first discuss the baselines used in this setting.

Algorithm 7 **Pre**-trained **De**cision **T**rans**f**ormer with **R**eward Estimation (PreDeToR)

Collecting Pretraining Dataset
Initialize empty pretraining dataset $\mathcal{H}_{\text{train}}$
for i in $[M_{\text{pre}}]$ **do**
    Sample task $m \sim \mathcal{T}_{\text{pre}}$, in-context dataset $\mathcal{H}_m \sim \mathcal{D}_{\text{pre}}(\cdot|m)$ and add this to $\mathcal{H}_{\text{train}}$.
Pretraining model on dataset
Initialize model $\text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}$ with parameters $\boldsymbol{\Theta}$
while not converged **do**
    Sample $\mathcal{H}_m$ from $\mathcal{H}_{\text{train}}$ and predict $\widehat{r}_{m,t}$ for action $(I_{m,t})$ for all $t \in [n]$
    Compute loss in (6.3) with respect to $r_{m,t}$ and backpropagate to update model parameter $\boldsymbol{\Theta}$.
Online test-time deployment
Sample unknown task $m \sim \mathcal{T}_{\text{test}}$ and initialize empty $\mathcal{H}^0_m = \{\emptyset\}$
for $t = 1, 2, \ldots, n$ **do**
    Use $\text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}$ on m at round t to choose

$$I_t \begin{cases} = \arg\max_{a \in \mathcal{A}} \text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}\left(\widehat{r}_{m,t}(a) \mid \mathcal{H}^t_m\right), & \textbf{PreDeToR} \\ \sim \text{softmax}^{\tau}_a \text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}\left(\widehat{r}_{m,t}(a) \mid \mathcal{H}^t_m\right), & \textbf{PreDeToR-}\tau \end{cases}$$

    Add $\{I_t, r_t\}$ to $\mathcal{H}^t_m$ to form $\mathcal{H}^{t+1}_m$.

(1) PreDeToR: This is our proposed method shown in Algorithm 7.

(2) PreDeToR-τ: This is the proposed exploratory method shown in Algorithm 7 and we fix $\tau = 0.05$.

(3) DPT-greedy: This baseline is the greedy approximation of the DPT algorithm from Lee et al. (2023) which is discussed in Section 6.1. Note that we choose DPT-greedy as a representative example of similar in-context decision-making algorithms studied in Lee et al. (2023); Sinii et al. (2023); Lin et al. (2023); Ma et al. (2023); Liu et al. (2023e,b) all of which require the optimal action (or its greedy approximation). DPT-greedy estimates the optimal arm using the reward estimates for each arm during

each task.

(4) AD: This is the Algorithmic Distillation method (Laskin et al., 2022; Lu et al., 2021) discussed in Section 6.1.

(5) Thomp: This baseline is the celebrated stochastic A-action bandit Thompson Sampling algorithm from Thompson (1933); Agrawal and Goyal (2012); Russo et al. (2018); Zhu and Tan (2020). We choose Thomp as the weak demonstrator π^w as it does not make use of arm features. Thomp is also a stochastic algorithm that induces more exploration in the demonstrations.

(6) LinUCB: (Linear Upper Confidence Bound): This baseline is the Upper Confidence Bound algorithm for the linear bandit setting that leverages the linear structure and feature of the arms to select the most promising action as well as conducting exploration. We choose LinUCB as a baseline for each test task to show the limitations of algorithms that use linear feedback structure as an underlying assumption to select actions. Note that LinUCB requires oracle access to features to select actions per task.

(7) MLinGreedy: This is the multi-task linear regression bandit algorithm proposed by Yang et al. (2021a). This algorithm assumes that there is a common low-dimensional feature extractor shared between the tasks and the reward of each task linearly depends on this feature extractor. We choose MLinGreedy as a baseline to show the limitations of algorithms that use linear feedback structure *across tasks* as an underlying assumption to select actions. Note that MLinGreedy requires oracle access to the action features to select actions as opposed to DPT, AD, and PreDeToR.

We describe in detail the baselines Thomp, LinUCB, and MLinGreedy for interested readers in Section E.1.

Outcomes: Before presenting the result we discuss the main outcomes from our experimental results in this section:

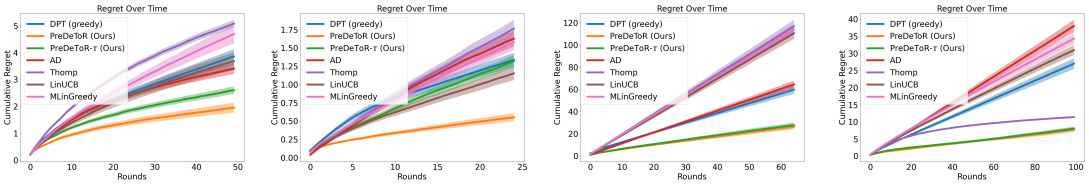
Finding 1: PreDeToR (-τ) lowers regret compared to other baselines under unknown, non-linear structure. It learns to exploit the latent structure of the underlying tasks from in-context data even when it is trained without the optimal action $a_{m,*}$ (or its approximation) and without action features $\mathcal{X}$.

Experimental Result: These findings are reported in Figure 6.1. In Figure 6.1a we show the non-linear bandit setting for horizon $n = 50$, $M_{pre} = 100000$, $M_{test} = 200$, $A = 6$, and $d = 2$. The demonstrator π^w is the Thomp algorithm. We observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy. Note that for this low data regime (short horizon) the DPT-greedy does not have a good estimation of $\widehat{a}_{m,*}$ which results in a poor prediction of optimal action $\widehat{a}_{m,t,*}$. This results in higher regret. The PreDeToR (-τ) has lower regret than LinUCB, and MLinGreedy, which fail to perform well in this non-linear setting due to their algorithmic design and linear feedback assumption. Finally, PreDeToR-τ performs slightly better than PreDeToR in both settings as it conducts more exploration.

In Figure 6.1b we show the non-linear bandit setting for horizon $n = 25$, $M_{pre} = 100000$, $M_{test} = 200$, $A = 6$, and $d = 2$ where the norm of the $\theta_{m,*}$ determines the reward of the actions which also is a non-linear function $\theta_{m,*}$ and action features. This setting is similar to the wheel bandit setting of Riquelme et al. (2018). Again, we observe that PreDeToR has lower cumulative regret than all the other baselines.

Finally in Figure 6.1c and Figure 6.1d we show the performance of PreDeToR against other baselines in real-world datasets Movielens and Yelp. The Movielens dataset consists of more than 32 million ratings of 200,000 users and 80,000 movies (Harper and Konstan, 2015) where each entry consists of user-id, movie-id, rating, and timestamp. The Yelp dataset (Asghar, 2016) consists of ratings of 1300 business categories by 150,000 users. Each entry is summarized as user-id, business-id, rating,

and timestamp. Previously structured bandit works (Deshpande and Montanari, 2012; Hong et al., 2023) directly fit a linear structure or low-rank factorization to estimate the $\theta_{m,*}$ and simulate the ratings. However, we directly use the user-ids and movie-ids (or business-ids) to build a histogram of ratings per user and calculate the mean rating per movie (or business-id) per task. Define this as the $\{\mu_{m,a}\}_{a=1}^{A}$. This is then used to simulate the rating for n horizon per movie per task where the data collection algorithm is uniform sampling. Note that this does not require estimation of user or movie features, and PreDeToR (-τ) learns to exploit the latent structure of user-movie (or business) rating correlations directly from the data. From Figure 6.1c and Figure 6.1d we see that PreDeToR, and PreDeToR-τ outperform all the other baselines in these settings.



(a) Non-linear bandit (b) Feature bandit (c) Movielens (d) Yelp

Figure 6.1: Non-linear regime. The horizontal axis is the number of rounds. Confidence bars show one standard error.

6.4 Empirical Study: Linear Structure and Understanding the Exploration of PreDeToR

The previous experiments were conducted in a non-linear structured setting where we are unaware of a provably near-optimal algorithm. To assess how close PreDeToR's regret is to optimal, in this section, we consider a *linear* setting for which there exist well-understood algorithms (Abbasi-Yadkori et al., 2011; Lattimore and Szepesvári, 2020a). Such algo-

rithms provide a strong upper bound for PreDeToR. We summarize the key finding below:

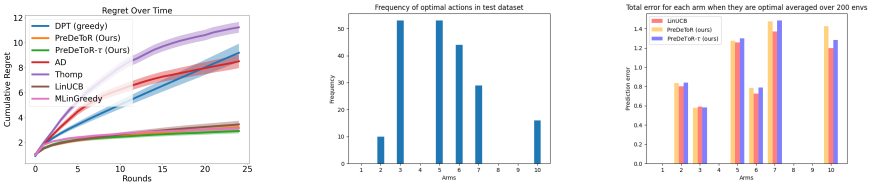
Finding 2: PreDeToR (-τ) matches the performance of the optimal algorithm LinUCB in linear bandit setting as it learns to exploit the latent structure across tasks from in-context data and without access to features.

In Figure 6.2 we first show the linear bandit setting for horizon $n = 25$, $M_{pre} = 200000$, $M_{test} = 200$, $A = 10$, and $d = 2$. Note that the length of the context (the number of rounds) is an artifact of the transformer architecture and computational complexity. This is because the self-attention takes in as input a length-n sequence of tokens of size d, and requires $O\left(dn^2\right)$ time to compute the output (Keles et al., 2023). Further empirical setting details are stated in Section E.1.

We observe from Figure 6.2 that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, and AD. Note that for this low data (short horizon) regime, the DPT-greedy does not have a good estimation of $\widehat{a}_{m,*}$ which results in a poor prediction of optimal action $\widehat{a}_{m,t,*}$. This results in higher regret. Observe that PreDeToR (-τ) performs quite similarly to LinUCB and lowers regret compared to Thomp which also shows that PreDeToR is able to exploit the latent linear structure and reward correlation of the underlying tasks. Note that LinUCB is close to the optimal algorithm for this linear bandit setting. PreDeToR outperforms AD as the main objective of AD is to match the performance of its demonstrator. In this short horizon, we see that MLinGreedy performs similarly to LinUCB.

We also show how the prediction error of the optimal action by PreDeToR is small compared to LinUCB in the linear bandit setting. In Figure 6.2b we first show how the 10 actions are distributed in the $M_{test} = 200$ test tasks. In Figure 6.2b for each bar, the frequency indicates the number of tasks where the action (shown in the x-axis) is the optimal action. Then,

in Figure 6.2c, we show the prediction error of PreDeToR (-τ) for each task $m \in [M_{\text{test}}]$. The prediction error is calculated as $(\widehat{\mu}_{m,n,*}(a) - \mu_{m,*}(a))^2$ where $\widehat{\mu}_{m,n,*}(a) = \max_a \widehat{\theta}_{m,n}^\top \mathbf{x}_m(a)$ is the empirical mean at the end of round n, and $\mu_{*,m}(a) = \max_a \theta_{m,*}^\top \mathbf{x}_m(a)$ is the true mean of the optimal action in task m. Then we average the prediction error for the action $a \in [A]$ by the number of times the action a is the optimal action in some task m. From the Figure 6.2c, we see that for actions $\{2, 3, 5, 6, 7, 10\}$, the prediction error of PreDeToR is either close or smaller than LinUCB. Note that LinUCB estimates the empirical mean directly from the test task, whereas PreDeToR has a strong prior based on the training data. So PreDeToR is able to estimate the reward of the optimal action quite well from the training dataset $\mathcal{D}_{\text{pre}}$. This shows the power of PreDeToR to go beyond the in-context decision-making setting studied in Lee et al. (2023); Lin et al. (2023); Ma et al. (2023); Sinii et al. (2023); Liu et al. (2023e) which require long horizons/trajectories and optimal action during training to learn a near-optimal policy. We discuss how exploration of PreDeToR (-τ) results in low cumulative regret in Section E.10.



(a) Linear Bandit setting (b) Test action distribution (c) Test Prediction Error

Figure 6.2: Linear Expt. The horizontal axis is the number of rounds. Confidence bars show one standard error.

6.5 Empirical Study: Importance of Shared Structure and Introducing New Arms

One of our central claims is that PreDeToR (-τ) internally learns and leverages the shared structure across the training and testing tasks. To validate this claim, in this section, we consider the introduction of new actions at test time that do *not* follow the structure of training time. These experiments are particularly important as they show the extent to which PreDeToR(-τ) is leveraging the latent structure and the shared correlation between the actions and rewards.

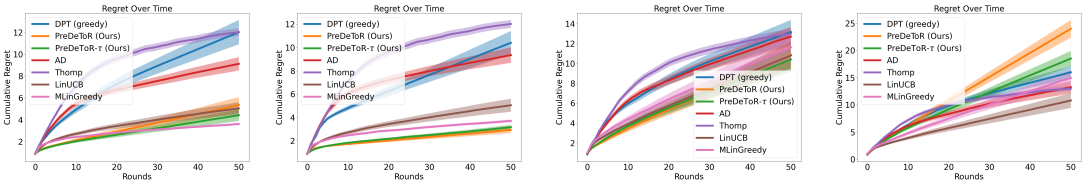
Invariant actions: We denote the set of actions fixed across the different tasks in the pretraining in-context dataset as $\mathcal{A}^{\text{inv}}$. Therefore these action features $\mathbf{x}(a) \in \mathbb{R}^d$ for $a \in \mathcal{A}^{\text{inv}}$ are fixed across the different tasks m. Note that these invariant actions help the transformer $\mathbf{T_w}$ to learn the latent structure and the reward correlation across the different tasks. Therefore, as the structure breaks down, PreDeToR starts performing worse than other baselines.

New actions: However, we also want to test how robust is PreDeToR (-τ) to new actions not seen during training time. To this effect, for each task $m \in [M_{\text{pre}}]$ and $m \in [M_{\text{test}}]$ we introduce $A - |\mathcal{A}^{\text{inv}}|$ new actions. *That is both for train and test tasks, we introduce new actions.* For each of these new actions $a \in [A - |\mathcal{A}^{\text{inv}}|]$ we choose the features $\mathbf{x}(m, a)$ randomly from $\mathcal{X} \subseteq \mathbb{R}^d$. Note the transformer now trains on a dataset $\mathcal{H}_m \subseteq \mathcal{D}_{\text{pre}} \neq \mathcal{D}_{\text{test}}$.

Baselines: We implement the same baselines discussed in Section 6.3.

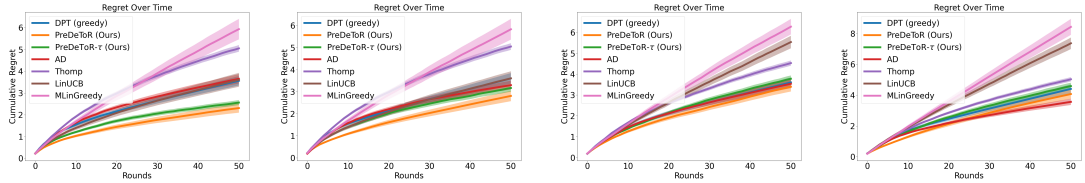
Outcomes: Again before presenting the result we discuss the main outcomes from our experimental results of introducing new actions during data collection and evaluation:

> **Finding 3:** PreDeToR (-τ) performance degrades as the shared structure breaks down.



(a) Linear 0 new action (b) Linear 1 new action (c) Linear 5 new actions (d) Linear 10 new actions

Figure 6.3: New action experiments. The horizontal axis is the number of rounds. Confidence bars show one standard error.



(a) Non-linear 0 new action (b) Non-linear 1 new action (c) Non-linear 5 new actions (d) Non-linear 10 new actions

Figure 6.4: New action experiments with non-linear setting.

Experimental Result: We observe these outcomes in Figure 6.3 and Figure 6.4. We consider the linear and non-linear bandit setting of horizon $n = 50$, $M_{pre} = 100000$, $M_{test} = 200$, $A = 10$, and $d = 2$. Here during data collection and during collecting the test data, we randomly select between $0, 1, 5$, and 10 new actions from $\mathbb{R}^d$ for each task m. So the number of invariant actions is $|\mathcal{A}^{inv}| \in \{10, 5, 1, 0\}$. Again, the demonstrator π^w is the Thomp algorithm. From Figure 6.3a, 6.3b, 6.3c, and 6.3d, we observe that when the number of invariant actions is less than PreDeToR (-τ) has lower cumulative regret than DPT-greedy, and AD. Observe that PreDeToR (-τ) matches LinUCB and has lower regret than DPT-greedy, and AD when $\mathcal{A}^{inv}| \in \{10, 5, 1\}$. This shows that PreDeToR (-τ) is able to exploit the latent linear structure of the underlying tasks. However, as the number of invariant actions decreases we see that PreDeToR(-τ) performance drops

and becomes similar to the unstructured bandits Thomp.

Similarly in Figure 6.4a, 6.4b, 6.4c, and 6.4d we show the performance of PreDeToR in the non-linear bandit setting. Observe that LinUCB, MLin-Greedy fails to perform well in this non-linear setting due to their assumption of linear rewards. Again note that PreDeToR (-τ) has lower regret than DPT-greedy, and AD when $\mathcal{A}^{\text{inv}}| \in \{10, 1\}$. This shows that PreDeToR (-τ) is able to exploit the latent linear structure of the underlying tasks. However, as the number of invariant actions decreases we see that PreDeToR(-τ) performance drops and becomes similar to AD.

We also empirically study the test performance of PreDeToR (-τ) in other *non-linear* bandit settings such as bilinear bandits (Section E.2), latent bandits (Section E.3), draw a connection between PreDeToR and Bayesian estimators (Section E.4), and perform sensitivity and ablation studies in Section E.5, E.7, E.8, E.9. We discuss data collection algorithms in Section E.12 and the offline setting in Section E.14. Due to space constraints, we refer the interested reader to the relevant section in the appendices.

6.6 Theoretical Analysis of Generalization

In this section, we present a theoretical analysis of how PreDeToR-τ generalizes to an unknown target task given a set of source tasks. We observe that PreDeToR-τ's performance hinges on a low excess error on the predicted reward of the actions of the unknown target task based on the in-context data. Thus, in our analysis, we show that, in low-data regimes, PreDeToR-τ has a low expected excess risk for the unknown target task as the number of source tasks increases. This is summarized as follows:

Finding 4: PreDeToR (-τ) has a low expected excess risk for the unknown target task as the number of source tasks increases. Moreover, the transfer learning risk of PreDeToR-τ (once trained on the M source tasks) scales with $\widetilde{O}(1/\sqrt{M})$.

To show this, we proceed as follows: Suppose we have the training data set $\mathcal{H}_{all} = \{\mathcal{H}_m\}_{m=1}^{M_{pre}}$, where the task $m \sim \mathcal{T}$ with a distribution $\mathcal{T}$ and the task data $\mathcal{H}_m$ is generated from a distribution $\mathcal{D}_{pre}(\cdot|m)$. For illustration purposes, here we consider the training data distribution $\mathcal{D}_{pre}(\cdot|m)$ where the actions are sampled following soft-LinUCB (a stochastic variant of LinUCB) (Chu et al., 2011). Given the loss function in eq. (6.3), we can define the task m training loss of PreDeToR-τ as $\widehat{\mathcal{L}}_m(\mathrm{TF}^{\mathbf{r}}_{\mathbf{\Theta}}) = \frac{1}{n}\sum_{t=1}^{n} \ell(r_{m,t}, \mathrm{TF}^{\mathbf{r}}_{\mathbf{\Theta}}(\widehat{r}_{m,t}(I_{m,t})|\mathcal{H}_m^t)) = \frac{1}{n}\sum_{t=1}^{n}(\mathrm{TF}^{\mathbf{r}}_{\mathbf{\Theta}}(\widehat{r}_{m,t}(I_{m,t})|\mathcal{H}_m^t) - r_{m,t})^2$. We drop the notation $\mathbf{\Theta}, \mathbf{r}$ from $\mathrm{TF}^{\mathbf{r}}_{\mathbf{\Theta}}$ for simplicity and let $M = M_{pre}$. We define

$$\widehat{\mathbf{T}} = \underset{\mathbf{T}\in \mathrm{Alg}}{\arg\min}\, \widehat{\mathcal{L}}_{\mathcal{H}_{all}}(\mathbf{T}) := \frac{1}{M}\sum_{m=1}^{M} \widehat{\mathcal{L}}_m(\mathbf{T}), \quad \text{(ERM)} \tag{6.4}$$

where Alg denotes the space of algorithms induced by the $\mathbf{T}$. Let $\mathcal{L}_m(\mathbf{T}) = \mathbb{E}_{\mathcal{H}_m}\big[\widehat{\mathcal{L}}_m(\mathbf{T})\big]$ and $\mathcal{L}_{MTL}(\mathbf{T}) = \mathbb{E}\big[\widehat{\mathcal{L}}_{\mathcal{H}_{all}}(\mathbf{T})\big] = \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}_m(\mathbf{T})$ be the corresponding population risks. For the ERM in (6.4), we want to bound the following excess Multi-Task Learning (MTL) risk of PreDeToR-τ

$$\mathcal{R}_{MTL}(\widehat{\mathbf{T}}) = \mathcal{L}_{MTL}(\widehat{\mathbf{T}}) - \min_{\mathbf{T}\in \mathrm{Alg}} \mathcal{L}_{MTL}(\mathbf{T}). \tag{6.5}$$

Note that for in-context learning, a training sample (I_t, r_t) impacts all future decisions of the algorithm from time step $t+1$ to n. Therefore, we need to control the stability of the input perturbation of the learning algorithm learned by the transformer. We introduce the following stability condition.

Assumption 9. (*Error stability* (*Bousquet and Elisseeff, 2002; Li et al., 2023*)). *Let* $\mathcal{H} = (I_t, r_t)_{t=1}^n$ *be a sequence in* $[A] \times [0,1]$ *with* $n \geqslant 1$ *and* $\mathcal{H}'$ *be the sequence where the* t'*th sample of* $\mathcal{H}$ *is replaced by* (I'_t, r'_t). *Error stability holds for a distribution* $(I, r) \sim \mathcal{D}$ *if there exists a* $K > 0$ *such that for any* $\mathcal{H}, (I'_t, r'_t) \in ([A] \times [0,1]), t \leqslant n$, *and* $\mathbf{T} \in \text{Alg}$, *we have*

$$\left|\mathbb{E}_{(I,r)}\left[\ell(r, \mathbf{T}(\widehat{r}(I)|\mathcal{H})) - \ell\left(r, \mathbf{T}(\widehat{r}(I)|\mathcal{H}')\right)\right]\right| \leqslant \tfrac{K}{n}.$$

Let ρ *be a distance metric on* Alg. *Pairwise error stability holds if for all* $\mathbf{T}, \mathbf{T}' \in$ Alg *we have*

$$\left|\mathbb{E}_{(x,y)}\left[\ell(r, \mathbf{T}(\widehat{r}(I)|\mathcal{H})) - \ell\left(r, \mathbf{T}'(\widehat{r}(I)|\mathcal{H})\right) - \ell(r, \mathbf{T}(\widehat{r}(I)|\mathcal{H}')) + \ell\left(r, \mathbf{T}'(\widehat{r}(I)|\mathcal{H}')\right)\right]\right| \leqslant \tfrac{K\rho(\mathbf{T},\mathbf{T}')}{n}.$$

Now we present the Multi-task learning (MTL) risk of PreDeToR-τ.

Theorem 6.1. (***PreDeToR risk***) *Suppose error stability Assumption 9 holds and assume loss function* $\ell(\cdot,\cdot)$ *is* C*-Lipschitz for all* $r_t \in [0, B]$ *and horizon* $n \geqslant 1$. *Let* $\widehat{\mathbf{T}}$ *be the empirical solution of* (*ERM*) *and* $\mathcal{N}(\mathcal{A}, \rho, \epsilon)$ *be the covering number of the algorithm space* Alg *following Definition E.2 and E.3. Then with probability at least* $1 - 2\delta$, *the excess MTL risk of PreDeToR-*τ *is bounded by*

$$\mathcal{R}_{\text{MTL}}(\widehat{\mathbf{T}}) \leqslant 4\tfrac{C}{\sqrt{nM}} + 2(B + K\log n)\sqrt{\tfrac{\log(\mathcal{N}(\text{Alg},\rho,\varepsilon)/\delta)}{cnM}},$$

where $\mathcal{N}(\text{Alg}, \rho, \varepsilon)$ *is the covering number of transformer* $\widehat{\mathbf{T}}$ *and* $\epsilon = 1/\sqrt{nM}$.

The proof of Theorem 6.1 is provided in Section E.16. From Theorem 6.1 we see that in low-data regime with a small horizon n, as the number of tasks M increases the MTL risk decreases. We further discuss the stability factor K and covering number $\mathcal{N}(\text{Alg}, \rho, \varepsilon)$ in Theorem E.4, and E.5.

We now present the transfer learning risk of PreDeToR-τ for an unknown target task $g \sim \mathcal{T}$ with the test dataset $\mathcal{H}_g \sim \mathcal{D}_{\text{test}}(\cdot|g)$. Note that

the test data distribution $\mathcal{D}_{\text{test}}(\cdot|g)$ is such that the actions are sampled following soft-LinUCB.

Theorem 6.2. (***Transfer risk***) *Consider the setting of Theorem 6.1 and assume the training source tasks are independently drawn from task distribution $\mathcal{T}$. Let $\widehat{TF}$ be the empirical solution of (ERM) and $g \sim \mathcal{T}$. Define the expected excess transfer learning risk $\mathbb{E}_g[\mathcal{R}_g] = \mathbb{E}_g\left[\mathcal{L}_g(\widehat{\mathbf{T}})\right] - \arg\min_{\mathbf{T}\in \text{Alg}} \mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right]$. Then with probability at least $1-2\delta$, the $\mathbb{E}_g\left[\mathcal{R}_g\right] \leqslant 4\frac{C}{\sqrt{M}} + 2B\sqrt{\frac{\log(\mathcal{N}(\text{Alg},\rho,\varepsilon)/\delta)}{M}}$, where $\mathcal{N}(\text{Alg}, \rho, \varepsilon)$ is the covering number of $\widehat{\mathbf{T}}$ and $\epsilon = \frac{1}{\sqrt{M}}$.*

The proof is given in Section E.16. This shows that for the transfer learning risk of PreDeToR-τ (once trained on the M source tasks) scales with $\widetilde{O}(1/\sqrt{M})$. This is because the unseen target task $g \sim \mathcal{T}$ induces a distribution shift, which, typically, cannot be mitigated with more samples n per task. A similar observation is provided in Lin et al. (2023). We further discuss this in Theorem E.7. We also observe a similar phenomenon empirically; see the discussion in Section E.13.

6.7 Conclusions, Limitations and Future Works

In this paper, we studied the supervised pretraining of decision transformers in the multi-task structured bandit setting when the knowledge of the optimal action is unavailable. Moreover, our proposed methods PreDeToR (-τ) do not need to know the action representations or the reward structure and learn these in-context with the help of offline data. The PreDeToR (-τ) predict the reward for the next action of each action during pretraining and can generalize well in-context in several regimes spanning low-data, new actions, and structured bandit settings like linear, non-linear, bilinear, latent bandits. The PreDeToR (-τ) outperforms other in-context algorithms like AD, DPT-greedy in most of the experiments. Finally, we theoretically analyze PreDeToR-τ and show that pretraining

it in M source tasks leads to a low expected excess error on a target task drawn from the same task distribution $\mathcal{T}$. In future, we want to extend our PreDeToR (-τ) to MDP setting (Sutton and Barto, 2018; Agarwal et al., 2019), and constraint MDP setting (Efroni et al., 2020; Gu et al., 2022).

Part IV

Adaptive Data Collection for Preference Elicitation, Prompt Designing, and LLM Alignment

7 OPTIMAL DESIGN FOR HUMAN PREFERENCE ELICITATION

Reinforcement learning from human feedback (*RLHF*) has been effective in aligning and fine-tuning *large language models* (*LLMs*) (Rafailov et al., 2023; Kang et al., 2023; Casper et al., 2023; Shen et al., 2023b; Kaufmann et al., 2024). The main difference from classic *reinforcement learning* (*RL*) (Sutton and Barto, 2018) is that the agent learns from human feedback, which is expressed as preferences for different potential choices (Akrour et al., 2012; Lepird et al., 2015; Sadigh et al., 2017; Bıyık et al., 2020; Wu and Sun, 2023). The human feedback allows LLMs to be adapted beyond the distribution of data that was used for their pre-training and generate answers that are more preferred by humans (Casper et al., 2023). The feedback can be incorporated by learning a preference model. When the human decides between two choices, the *Bradley-Terry-Luce* (*BTL*) model (Bradley and Terry, 1952) can be used. For multiple choices, the *Plackett-Luce* (*PL*) model (Plackett, 1975; Luce, 2005) can be adopted. A good preference model should correctly rank answers to many potential questions. Therefore, learning of a good preference model can be viewed as learning to rank, and we adopt this view in this work. Learning to rank has been studied extensively in both offline (Burges, 2010) and online (Radlinski et al., 2008; Kveton et al., 2015; Szörényi et al., 2015; Sui et al., 2018; Lattimore et al., 2018) settings.

To effectively learn preference models, we study efficient methods for human preference elicitation. We formalize this problem as follows. We have a set of L *lists* representing *questions*, each with K *items* representing *answers*. The objective of the agent is to learn to rank all items in all lists. The agent can query humans for feedback. Each query is a question with K answers represented as a list. The human provides feedback on it. We study two feedback models: absolute and ranking. In the absolute

feedback model, a human provides noisy feedback for each item in the list. This setting is motivated by how annotators assign relevance judgments in search (Hofmann et al., 2013; MS MARCO, 2016). The ranking feedback is motivated by learning reward models in RLHF (Rafailov et al., 2023; Kang et al., 2023; Casper et al., 2023; Shen et al., 2023b; Kaufmann et al., 2024). In this model, a human ranks all items in the list according to their preferences. While $K = 2$ is arguably the most common case, we study $K \geqslant 2$ for the sake of generality and allowing a higher-capacity communication channel with the human (Zhu et al., 2023b). The agent has a budget for the number of queries. To learn efficiently within the budget, it needs to elicit preferences from the most informative lists, which allows it to learn to rank all other lists. Our main contribution is an efficient algorithm for computing the distribution of the most informative lists.

Our work touches on many topics. Learning of reward models from human feedback is at the center of RLHF (Ouyang et al., 2022) and its recent popularity has led to major theory developments, including analyses of regret minimization in RLHF (Chen et al., 2022b; Wang et al., 2023c; Wu and Sun, 2023; Xiong et al., 2023; Opoku-Agyemang, 2023; Saha et al., 2023). These works propose and analyze adaptive algorithms that interact with the environment to learn highly-rewarding policies. Such policies are usually hard to deploy in practice because they may harm user experience due to over-exploration (Dudík et al., 2014; Swaminathan and Joachims, 2015). Therefore, Zhu et al. (2023b) studied RLHF from ranking feedback in the offline setting with a fixed dataset. We study how to collect an *informative dataset for offline learning to rank* with both absolute and ranking feedback. We approach this problem as an optimal design, a methodology for computing optimal information-gathering policies (Pukelsheim, 2006; Fedorov, 2013). The policies are non-adaptive and thus can be precomputed, which is one of their advantages. The main technical contribution of this work is a matrix generalization of the Kiefer-Wolfowitz theorem

(Kiefer and Wolfowitz, 1960), which allows us to formulate optimal designs for ranked lists and solve them efficiently. Optimal designs have become a standard tool in exploration (Lattimore and Szepesvári, 2020a; Katz-Samuels et al., 2020, 2021; Mukherjee et al., 2022b; Jamieson and Jain, 2022) and adaptive algorithms can be obtained by combining them with elimination. Therefore, optimal designs are also a natural stepping stone to other solutions.

We make the following contributions:

1. We develop a novel approach for human preference elicitation. The key idea is to generalize the Kiefer-Wolfowitz theorem (Kiefer and Wolfowitz, 1960) to matrices (section 7.3), which then allows us to compute information-gathering policies for ranked lists.

2. We propose an algorithm that uses an optimal design to collect absolute human feedback (section 7.4), where a human provides noisy feedback for each item in the queried list. A least-squares estimator is then used to learn a preference model. The resulting algorithm is both computationally and statistically efficient. We bound its prediction error (section 7.4) and ranking loss (section 7.4), and show that both decrease with the sample size.

3. We propose an algorithm that uses an optimal design to collect ranking human feedback (section 7.5), where a human ranks all items in the list according to their preferences. An estimator of Zhu et al. (2023b) is then used to learn a preference model. Our approach is both computationally and statistically efficient, and we bound its prediction error (section 7.5) and ranking loss (section 7.5). These results mimic the absolute feedback setting and show the generality of our framework.

4. We compare our algorithms to multiple baselines in several experiments. We observe that the algorithms achieve a lower ranking loss

than the baselines.

7.1 Setting

Notation: Let $[K] = \{1, \dots, K\}$. Let Δ^L be the probability simplex over $[L]$. For any distribution $\pi \in \Delta^L$, we get $\sum_{i=1}^{L} \pi(i) = 1$. Let $\Pi_2(K) = \{(j,k) \in [K]^2 : j < k\}$ be the set of all pairs over $[K]$ where the first entry is lower than the second one. Let $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^\top \mathbf{A}\mathbf{x}$ for any positive-definite $\mathbf{A} \in \mathbb{R}^{d \times d}$ and $\mathbf{x} \in \mathbb{R}^d$. We use $\tilde{O}$ for the big-O notation up to logarithmic factors. Specifically, for any function f, we write $\tilde{O}(f(n))$ if it is $O(f(n) \log^k f(n))$ for some $k > 0$. Let $\text{supp}(\pi)$ be the support of distribution π or a random variable.

Setup: We learn to rank L lists, each with K items. An item $k \in [K]$ in list $i \in [L]$ is represented by a feature vector $\mathbf{x}_{i,k} \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the set of feature vectors. The relevance of items is given by their mean rewards. The mean reward of item k in list i is $\mathbf{x}_{i,k}^\top \boldsymbol{\theta}_*$, where $\boldsymbol{\theta}_* \in \mathbb{R}^d$ is an unknown parameter. Without loss of generality, we assume that the original order of the items is optimal, $\mathbf{x}_{i,j}^\top \boldsymbol{\theta}_* > \mathbf{x}_{i,k}^\top \boldsymbol{\theta}_*$ for any $j < k$ and list i. The agent does not know it. The agent interacts with humans for n rounds. At round t, it selects a list I_t and the human provides stochastic feedback on it. Our goal is to design a policy for selecting the lists such that the agent learns the optimal order of all items in all lists after n rounds.

Feedback model: We study two models of human feedback, absolute and ranking:

(1) In the *absolute feedback model*, the human provides a reward for each item in list I_t chosen by the agent. Specifically, the agent observes noisy rewards

$$y_{t,k} = \mathbf{x}_{I_t,k}^\top \boldsymbol{\theta}_* + \eta_{t,k}\,, \tag{7.1}$$

for all $k \in [K]$ in list I_t, where $\eta_{t,k}$ is independent zero-mean 1-sub-

Gaussian noise. This feedback is stochastic and similar to that in the document-based click model (Chuklin et al., 2022).

(2) In the *ranking feedback model*, the human orders all K items in list I_t selected by the agent. The feedback is a permutation $\sigma_t : [K] \to [K]$, where $\sigma_t(k)$ is the index of the k-th ranked item. The probability that this permutation is generated is

$$p(\sigma_t) = \prod_{k=1}^{K} \frac{\exp[\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}_*]}{\sum_{j=k}^{K} \exp[\mathbf{x}_{I_t,\sigma_t(j)}^\top \boldsymbol{\theta}_*]} \,. \tag{7.2}$$

Simply put, items with higher mean rewards are more preferred by humans and hence more likely to be ranked higher. This feedback model is known as the *Plackett-Luce* (*PL*) model (Plackett, 1975; Luce, 2005; Zhu et al., 2023b), and it is a standard assumption when learning values of individual choices from relative feedback. Since the feedback at round t is with independent noise, in both (7.1) and (7.2), any list can be observed multiple times and we do need to assume that $n \leqslant L$.

Objective: At the end of round n, the agent outputs a permutation $\hat{\sigma}_{n,i} : [K] \to [K]$ for all lists $i \in [L]$, where $\hat{\sigma}_{n,i}(k)$ is the item placed at position k in list i. Our evaluation metric is the *ranking loss* after n rounds, which we define as

$$R_n = \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbb{1}\{\hat{\sigma}_{n,i}(j) > \hat{\sigma}_{n,i}(k)\} \,. \tag{7.3}$$

The loss is the number of incorrectly ordered pairs of items in permutation $\hat{\sigma}_{n,i}$, summed over all lists $i \in [L]$. It can also be viewed as the Kendall tau rank distance (Kendall, 1948) between the optimal order of items in all lists and that according to $\hat{\sigma}_{n,i}$. We note that other ranking metrics exist, such as the *normalized discounted cumulative gain* (*NDCG*) (Wang et al., 2013) and *mean reciprocal rank* (*MRR*) (Voorhees, 1999). Our work can be

extended to them and we leave this for future work.

The two closest related works are Mehta et al. (2023) and Das et al. (2024). They proposed algorithms for learning to rank L pairs of items from pairwise feedback. Their optimized metric is the maximum gap over the L pairs. We learn to rank L lists of K items from K-way ranking feedback. We bound the maximum prediction error, which is a similar metric to the prior works, and the ranking loss in (7.3), which is novel. Our setting is related to other bandit settings as follows. Due to the budget n, it is similar to fixed-budget *best arm identification* (*BAI*) (Bubeck et al., 2009; Audibert et al., 2010; Azizi et al., 2022; Yang and Tan, 2022). The main difference is that we do not want to identify the best arm. We want to sort L lists of K items. Online learning to rank has also been studied extensively (Radlinski et al., 2008; Kveton et al., 2015; Zong et al., 2016; Li et al., 2016; Lagree et al., 2016). We do not minimize cumulative regret or try to identify the best arm. A more detailed comparison is in section 7.2.

We introduce optimal designs (Pukelsheim, 2006; Fedorov, 2013) next. This allows us to minimize the expected ranking loss within a budget of n rounds efficiently.

7.2 Related Work

The two closest related works are Mehta et al. (2023) and Das et al. (2024). They proposed algorithms for learning to rank L pairs of items from pairwise feedback. Their optimized metric is the maximum gap over L pairs. We learn to rank L lists of K items from K-way ranking feedback. We bound the maximum prediction error, which is a similar metric to these related works, and the ranking loss in (7.3), which is novel. Algorithm APO in Das et al. (2024) is the closest related algorithmic design. APO greedily minimizes the maximum error in pairwise ranking of L lists of length $K = 2$. Therefore, Dope with ranking feedback (section 7.5) can be

viewed as a generalization of Das et al. (2024) to lists of length $K \geqslant 2$. APO can be compared to Dope by applying it to all possible $\binom{K}{2}L$ lists of length 2 created from our lists of length K. We do that in APO in section 7.6. The last difference from Das et al. (2024) is that they proposed two variants of APO: analyzed and practical. We propose a single algorithm, which is both analyzable and practical.

Our problem can be generally viewed as learning preferences from human feedback (Fürnkranz and Hüllermeier, 2003; Glass, 2006; Houlsby et al., 2011). The two most common forms of feedback are pairwise, where the agent observes a preference over two items (Bradley and Terry, 1952); and ranking, where the agent observes a ranking of the items (Plackett, 1975; Luce, 2005). Online learning from human feedback has been studied extensively. In online learning to rank Radlinski et al. (2008); Zoghi et al. (2017), the agent selects a list of K items and the human provides absolute feedback, in the form of clicks, on all recommended items or their subset. Two popular feedback models are cascading (Kveton et al., 2015; Zong et al., 2016; Li et al., 2016) and position-based (Lagree et al., 2016; Ermis et al., 2020; Zhou et al., 2023) models. The main difference in section 7.4 is that we do not minimize cumulative regret or try to identify the best list of K items. We learn to rank L lists of K items within a budget of n observations. Due to the budget, our work is related to fixed-budget BAI (Bubeck et al., 2009; Audibert et al., 2010; Azizi et al., 2022; Yang and Tan, 2022). The main difference is that we do not aim to identify the best arm.

Online learning from preferential feedback has been studied extensively (Bengs et al., 2021) and is often formulated as a dueling bandit (Yue et al., 2012; Lekang and Lamperski, 2019; Xu et al., 2020a; Kirschner and Krause, 2021; Pasztor et al., 2024; Saha, 2021; Saha and Krishnamurthy, 2022; Saha and Gaillard, 2022; Saha et al., 2023; Takeno et al., 2023; Xu et al., 2024b). Our work on ranking feedback (section 7.5) differs from these works in three main aspects. First, most dueling bandit papers consider

pairwise feedback ($K = 2$) while we study a more general setting of $K \geqslant 2$. Second, a classic objective in dueling bandits is to minimize regret with respect to the best arm, sometimes in context; either in a cumulative or simple regret setting. We do not minimize cumulative or simple regret. We learn to rank L lists of K items. Finally, the acquisition function in dueling bandits is adaptive and updated in each round. Dope is a static design where the exploration policy is precomputed.

Preference-based learning has also been studied in a more general setting of reinforcement learning (Wirth et al., 2017; Novoseller et al., 2020; Xu et al., 2020b; Hejna and Sadigh, 2023). Preference-based RL differs from classic RL by learning human preferences through non-numerical rewards (Christiano et al., 2017; Lee et al., 2021; Chen et al., 2022b). Our work can be also viewed as collecting human feedback for learning policies offline (Jin et al., 2021; Rashidinejad et al., 2021; Zanette et al., 2021; Sekhari et al., 2024). One of the main challenges of offline learning is potentially insufficient data coverage. We address this by collecting diverse data, using optimal designs (Pukelsheim, 2006; Fedorov, 2013).

Finally, we wanted to compare the ranking loss in (7.3) to other objectives. There is no reduction to dueling bandits. A classic objective in dueling bandits is to *minimize regret with respect to the best arm* from dueling feedback. Our goal is to *rank* L *lists*. One could think that our problem can be solved as a contextual dueling bandit, where each list is represented as a context. This is not possible because the context is controlled by the environment. In our setting, the agent controls the chosen list, similarly to APO in Das et al. (2024). Our objective also cannot be reduced to fixed-budget BAI. Our comparisons to Azizi et al. (2022) (sections 7.4 and 7.5) focus on similarities in high-probability bounds. The dependence on n and d is expected to be similar because the probability of making a mistake in Azizi et al. (2022) and a ranking error in our work depend on how well the generalization model is estimated, which is the same in both works.

7.3 Optimal Design and Matrix Kiefer-Wolfowitz

This section introduces a unified approach to human preference elicitation from both absolute and ranking feedback. First, we note that to learn the optimal order of items in all lists, the agent has to estimate the unknown model parameter θ_* well. In this work, the agent uses a *maximum-likelihood estimator* (*MLE*) to obtain an estimate $\widehat{\theta}_n$ of θ_*. After that, it orders the items in all lists according to their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\theta}_n$ in descending order to get the permutation $\hat{\sigma}_{n,i}$. If $\widehat{\theta}_n$ would minimize the prediction error $(\mathbf{x}_{i,k}^\top(\widehat{\theta}_n - \theta_*))^2$ over all items $k \in [K]$ in list i, the permutation $\hat{\sigma}_{n,i}$ would be closer to the optimal order. Moreover, if $\widehat{\theta}_n$ minimized the maximum error over all lists, all permutations would be closer and the ranking loss in (7.3) would be minimized. This is why we focus on minimizing the *maximum prediction error*

$$\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} (\mathbf{a}^\top(\widehat{\theta}_n - \theta_*))^2 = \max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top(\widehat{\theta}_n - \theta_*)(\widehat{\theta}_n - \theta_*)^\top \mathbf{A}_i)\,, \quad (7.4)$$

where $\mathbf{A}_i$ is a matrix representing list i and $\mathbf{a} \in \mathbf{A}_i$ is a column in it. In the absolute feedback model, the columns of $\mathbf{A}_i$ are feature vectors of items in list i (section 7.4). In the ranking feedback model, the columns of $\mathbf{A}_i$ are the differences of feature vectors of items in list i (section 7.5). Therefore, $\mathbf{A}_i$ depends on the type of human feedback. In fact, as we show later, it is dictated by the covariance of $\widehat{\theta}_n$ in the corresponding human feedback model. We note that the objective in (7.4) is worst-case over lists and that other alternatives, such as $\frac{1}{L}\sum_{i=1}^{L}\sum_{\mathbf{a} \in \mathbf{A}_i}(\mathbf{a}^\top(\widehat{\theta}_n - \theta_*))^2$, may be possible. We leave this for future work.

We prove in sections 7.4 and 7.5 that the agent can minimize the maximum prediction error in (7.4) and the ranking loss in (7.3) by sampling from a fixed distribution $\pi_* \in \Delta^L$. That is, the probability of selecting list i

at round t is $\mathbb{P}(I_t = i) = \pi_*(i)$. The distribution π_* is a minimizer of

$$g(\pi) = \max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)\,, \tag{7.5}$$

where $\mathbf{V}_\pi = \sum_{i=1}^{L} \pi(i)\mathbf{A}_i\mathbf{A}_i^\top$ is a *design matrix*. The *optimal design* aims to find the distribution π_*. Since (7.5) does not depend on the received feedback, our algorithms are not adaptive.

The problem of finding π_* that minimizes (7.5) is called the *G-optimal design* Lattimore and Szepesvári (2020a). The minimum of (7.5) and the support of π_* are characterized by the Kiefer-Wolfowitz theorem Kiefer and Wolfowitz (1960); Lattimore and Szepesvári (2020a). The original theorem is for least-squares regression, where $\mathbf{A}_i$ are feature vectors. At a high level, it says that the smallest ellipsoid that covers all feature vectors has the minimum volume, and in this way relates the minimization of (7.5) to maximizing $\log\det(\mathbf{V}_\pi)$. We generalize this claim to lists, where $\mathbf{A}_i$ is a matrix of feature vectors representing list i. This generalization allows us to go from a design over feature vectors to a design over lists represented by matrices.

Theorem 7.1 (Matrix Kiefer-Wolfowitz)**.** *Let $M \geqslant 1$ be an integer and $\mathbf{A}_1, \ldots, \mathbf{A}_L \in \mathbb{R}^{d \times M}$ be L matrices whose column space spans $\mathbb{R}^d$. Then the following claims are equivalent:*

(a) π_* is a minimizer of $g(\pi)$ defined in (7.5).

(b) π_* is a maximizer of $f(\pi) = \log\det(\mathbf{V}_\pi)$.

(c) $g(\pi_*) = d$.

Furthermore, there exists a minimizer π_* of $g(\pi)$ such that $|\mathrm{supp}\,(\pi_*)| \leqslant d(d+1)/2$.

Proof. We generalize the proof of the Kiefer-Wolfowitz theorem in Lattimore and Szepesvári (2020a). The key observation is that even if $\mathbf{A}_i$ is a

matrix and not a vector, the design matrix $\mathbf{V}_\pi$ is positive definite. Using this, we establish three key facts that are used in the original proof. First, we show that $f(\pi)$ is concave in π and that $(\nabla f(\pi))_i = \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)$ is its gradient with respect to $\pi(i)$. Second, $\sum_{i=1}^{L} \pi(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i) = d$. Finally, we prove that $g(\pi) \geqslant \sum_{i=1}^{L} \pi(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)$. The complete proof is in section F.1. □

From the equivalence in theorem 7.1, it follows that the agent should solve the optimal design

$$\pi_* = \underset{\pi \in \Delta^L}{\arg\max}\, f(\pi) = \underset{\pi \in \Delta^L}{\arg\max} \log\det(\mathbf{V}_\pi) \tag{7.6}$$

and sample according to π_* to minimize the maximum prediction error in (7.4). Note that the optimal design over lists in (7.6) is different from the one over features (Lattimore and Szepesvári, 2020a). As an example, suppose that we have 4 feature vectors $\{\mathbf{x}_i\}_{i\in[4]}$ and two lists: $\mathbf{A}_1 = (\mathbf{x}_1, \mathbf{x}_2)$ and $\mathbf{A}_2 = (\mathbf{x}_3, \mathbf{x}_4)$. The list design is over 2 variables (lists) while the feature-vector design is over 4 variables (feature vectors). The list design can also be viewed as a constrained feature-vector design, where $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_3, \mathbf{x}_4)$ are observed together with the same probability.

The optimization problem in (7.6) is convex and thus easy to solve. When the number of lists is large, the Frank-Wolfe algorithm (Nocedal and Wright, 1999; Jaggi, 2013) can be applied, which solves convex optimization problems with linear constraints as a sequence of linear programs. We use CVXPY (Diamond and Boyd, 2016) to compute the optimal design. We report its computation time, as a function of the number of lists L, in section F.4. The computation time scales roughly linearly with the number of lists L. In the following sections, we utilize theorem 7.1 to bound the maximum prediction error and ranking loss for both absolute and ranking feedback.

7.4 Learning with Absolute Feedback

This section is organized as follows. In section 7.4, we present an algorithm for human preference elicitation under absolute feedback. We bound its prediction error in section 7.4 and its ranking loss in section 7.4.

Algorithm Dope

Now we present our algorithm for absolute feedback called **D**-**o**ptimal **p**reference **e**licitation (Dope). The algorithm has four main parts. First, we solve the optimal design problem in (7.6) to get a data logging policy π_*. The matrix for list i is $\mathbf{A}_i = [\mathbf{x}_{i,k}]_{k \in [K]} \in \mathbb{R}^{d \times K}$, where $\mathbf{x}_{i,k}$ is the feature vector of item k in list i. Second, we collect human feedback for n rounds. At round $t \in [n]$, we sample a list $I_t \sim \pi_*$ and then observe $y_{t,k}$ for all $k \in [K]$, as defined in (7.1). Third, we estimate the model parameter as

$$\widehat{\theta}_n = \overline{\Sigma}_n^{-1} \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} y_{t,k} \,. \tag{7.7}$$

The normalized and unnormalized covariance matrices corresponding to the estimate are

$$\Sigma_n = \frac{1}{n} \overline{\Sigma}_n \,, \quad \overline{\Sigma}_n = \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} \mathbf{x}_{I_t,k}^\top \,, \tag{7.8}$$

respectively. Finally, we sort the items in all lists i according to their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\theta}_n$ in descending order, to obtain the permutation $\hat{\sigma}_{n,i}$. The pseudo-code of Dope is in algorithm 8.

The estimator (7.7) is the same as in *ordinary least squares* (*OLS*), because each observed list can be treated as K independent observations. The matrix for list i, $\mathbf{A}_i$, can be related to the inner sum in (7.8) through $\mathbf{Tr}(\mathbf{A}_i \mathbf{A}_i^\top) = \sum_{k=1}^{K} \mathbf{x}_{i,k} \mathbf{x}_{i,k}^\top$. Therefore, our optimal design for absolute feed-

Algorithm 8 Dope for absolute feedback.

for $i = 1, \dots, L$ **do**
  $\mathbf{A}_i \leftarrow [\mathbf{x}_{i,k}]_{k \in [K]}$
$\mathbf{V}_\pi \leftarrow \sum_{i=1}^{L} \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$
$\pi_* \leftarrow \arg\max_{\pi \in \Delta^L} \log\det(\mathbf{V}_\pi)$
for $t = 1, \dots, n$ **do**
  Sample $I_t \sim \pi_*$
  for $k = 1, \dots, K$ **do**
    Observe $y_{t,k}$ in (7.1)
Compute $\widehat{\theta}_n$ in (7.7)
for $i = 1, \dots, L$ **do**
  Set $\hat{\sigma}_{n,i}(k)$ to item with the k-th highest mean reward in list i, $\mathbf{x}_{i,k}^\top \widehat{\theta}_n$
Output: Permutation $\hat{\sigma}_{n,i}$ for all $i \in [L]$

Algorithm 9 Dope for ranking feedback.

for $i = 1, \dots, L$ **do**
  for $(j, k) \in \Pi_2(K)$ **do**
    $\mathbf{z}_{i,j,k} \leftarrow \mathbf{x}_{i,j} - \mathbf{x}_{i,k}$
  $\mathbf{A}_i \leftarrow [\mathbf{z}_{i,j,k}]_{(j,k) \in \Pi_2(K)}$
$\mathbf{V}_\pi \leftarrow \sum_{i=1}^{L} \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$
$\pi_* \leftarrow \arg\max_{\pi \in \Delta^L} \log\det(\mathbf{V}_\pi)$
for $t = 1, \dots, n$ **do**
  Sample $I_t \sim \pi_*$
  Observe σ_t in (7.2)
Compute $\widehat{\theta}_n$ in (7.10)
for $i = 1, \dots, L$ **do**
  Set $\hat{\sigma}_{n,i}(k)$ to item with the k-th highest mean reward in list i, $\mathbf{x}_{i,k}^\top \widehat{\theta}_n$
Output: Permutation $\hat{\sigma}_{n,i}$ for all $i \in [L]$

back logs data for a least-squares estimator by optimizing its covariance Lattimore and Szepesvári (2020a); Jamieson and Jain (2022).

Maximum Prediction Error Under Absolute Feedback

In this section, we bound the maximum prediction error of Dope under absolute feedback. We start with a lemma that uses the optimal design π_* to bound $\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}}$.

Lemma 7.2. *Let π_* be the optimal design in (7.6). Fix budget n and let each allocation $n\pi_*(i)$ be an integer. Then* $\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}} = d/n$.

The lemma is proved in section F.1. Since all $n\pi_*(i)$ are integers, we note that $\overline{\mathbf{\Sigma}}_n$ is full rank and thus invertible. The condition of the lemma,

that each $n\pi_*(i)$ is an integer, does not require $n \geqslant L$. This is because $\pi_*(i)$ has at most $d(d+1)/2$ non-zero entries (theorem 7.1). This is independent of the number of lists L, which could also be infinite (Chapter 21.1 in Lattimore and Szepesvári (2020a)). The integer condition can be also relaxed by rounding non-zero entries of $n\pi_*(i)$ up to the closest integer. This clearly yields an integer allocation of size at most $n + d(d+1)/2$. All claims in our work would hold for any π_* and this allocation. With theorem 7.2 in hand, the maximum prediction error is bounded as follows.

Theorem 7.3 (Maximum prediction error)**.** *With probability at least* $1-\delta$*, the maximum prediction error after* n *rounds is*

$$\max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top (\widehat{\theta}_n - \theta_*)(\widehat{\theta}_n - \theta_*)^\top \mathbf{A}_i) = O\left(\frac{d^2 + d\log(1/\delta)}{n}\right).$$

The theorem is proved in section F.1. As in theorem 7.2, we assume that each allocation $n\pi_*(i)$ is an integer. If the allocations were not integers, rounding errors would arise and need to be bounded (Pukelsheim, 2006; Fiez et al., 2019; Katz-Samuels et al., 2020). At a high level, our bound would be multiplied by $1+\beta$ for some $\beta > 0$ (Chapter 21 in Lattimore and Szepesvári (2020a)). We omit this factor in our proofs to simplify them.

theorem 7.3 says that the maximum prediction error is $\tilde{O}(d^2/n)$. Note that this rate cannot be attained trivially, for instance by uniform sampling. To see this, consider the following example. Take $K = 2$. Let $\mathbf{x}_{i,1} = (1,0,0)$ for $i \in [L-1]$ and $\mathbf{x}_{L,1} = (0,1,0)$, and $\mathbf{x}_{i,2} = (0,0,1)$ for all $i \in [L]$. In this case, the minimum eigenvalue of $\overline{\Sigma}_n$ is n/L in expectation, because only one item in list L provides information about the second feature, $\mathbf{x}_{L,1} = (0,1,0)$. Following the same steps as in theorem 7.3, we would get a rate of $\tilde{O}(dL/n)$. Prior works on optimal designs also made similar observations (Soare et al., 2014).

The rate in theorem 7.3 is the same as in linear models. More specifically,

by the Cauchy-Schwarz inequality, we would get

$$(\mathbf{x}^\top(\hat{\theta}_n - \theta_*))^2 \leqslant \|\hat{\theta}_n - \theta_*\|^2_{\overline{\Sigma}_n} \|\mathbf{x}\|^2_{\overline{\Sigma}_n^{-1}} = \tilde{O}(d)\,\tilde{O}(d/n) = \tilde{O}(d^2/n)$$

with a high probability, where θ_*, $\hat{\theta}_n$, and $\overline{\Sigma}_n$ are the analogous linear model quantities. This bound holds for infinitely many feature vectors. It can be tightened to $\tilde{O}(d/n)$ for a finite number of feature vectors, where $\tilde{O}$ hides the logarithm of the number of feature vectors. This can be proved using a union bound over (20.3) in Chapter 20 of Lattimore and Szepesvári (2020a).

Ranking Loss Under Absolute Feedback

In this section, we bound the expected ranking loss under absolute feedback. Recall from section 7.1 that the original order of items in each list is optimal. With this in mind, the *gap* between the mean rewards of items j and k in list i is $\Delta_{i,j,k} = (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \theta_*$, for any $i \in [L]$ and $(j,k) \in \Pi_2(K)$.

Theorem 7.4 (Ranking loss)**.** *The expected ranking loss after n rounds is bounded as*

$$\mathbb{E}[R_n] \leqslant 2 \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \exp\left[-\frac{\Delta_{i,j,k}^2 n}{4d}\right].$$

Proof. From the definition of the ranking loss, we have

$$\mathbb{E}[R_n] = \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbb{E}[\mathbb{1}\{\hat{\sigma}_{n,i}(j) > \hat{\sigma}_{n,i}(k)\}] = \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\theta}_n < \mathbf{x}_{i,k}^\top \widehat{\theta}_n\right).$$

where $\mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\theta}_n < \mathbf{x}_{i,k}^\top \widehat{\theta}_n\right)$ is the probability of predicting a sub-optimal item k above item j in list i. We bound this probability from above by bounding the sum of $\mathbb{P}\left(\mathbf{x}_{i,k}^\top(\widehat{\theta}_n - \theta_*) > \frac{\Delta_{i,j,k}}{2}\right)$ and $\mathbb{P}\left(\mathbf{x}_{i,j}^\top(\theta_* - \widehat{\theta}_n) > \frac{\Delta_{i,j,k}}{2}\right)$.

Each of these probabilities is bounded from above by $\exp\left[-\frac{\Delta_{i,j,k}^2 n}{4d}\right]$, using a concentration inequality in theorem F.2. The full proof is in section F.1. □

Each term in theorem 7.4 can be bounded from above by $\exp\left[-\frac{\Delta_{\min}^2 n}{4d}\right]$, where n is the sample size, d is the number of features, and $\Delta_{\min}$ denotes the minimum gap. Therefore, the bound decreases exponentially with budget n and gaps, and increases with d. This dependence is similar to that in Theorem 1 of Azizi et al. (2022) for fixed-budget best-arm identification in linear models. Yang and Tan (2022) derived a similar bound and a matching lower bound. The gaps $\Delta_{i,j,k}$ reflect the hardness of sorting list i, which depends on the differences of the mean rewards of items j and k in it.

Finally, we wanted to note that our optimal designs may not be optimal for ranking. We have not focused solely on ranking because we see value in both prediction error (theorem 7.3) and ranking loss (theorem 7.4) bounds. The fact that we provide both shows the versatility of our approach.

7.5 Learning with Ranking Feedback

This section is organized similarly to section 7.4. In section 7.5, we present an algorithm for human preference elicitation under ranking feedback. We bound its prediction error in section 7.5 and its ranking loss in section 7.5. Our algorithm design and analysis are under the following assumption, which we borrow from Zhu et al. (2023b).

Assumption 10. *We assume that the model parameter satisfies* $\theta_* \in \mathbf{\Theta}$, *where*

$$\mathbf{\Theta} = \{\theta \in \mathbb{R}^d : \theta^\top \mathbf{1}_d = 0, \|\theta\|_2 \leqslant 1\}. \tag{7.9}$$

We also assume that $\max_{i\in[L],\, k\in[K]} \|\mathbf{x}_{i,k}\|_2 \leqslant 1$.

The assumption of bounded model parameter and feature vectors is common in bandits (Abbasi-Yadkori et al., 2011; Lattimore and Szepesvári, 2020a). The additional assumption of $\theta^\top \mathbf{1}_d = 0$ is from Zhu et al. (2023b), from which we borrow the estimator and concentration bound.

Algorithm Dope

We present Dope for ranking feedback next. The algorithm is similar to Dope in section 7.4 and has four main parts. First, we solve the optimal design problem in (7.6) to get a data logging policy π_*. The matrix for list i is $\mathbf{A}_i = [\mathbf{z}_{i,j,k}]_{(j,k)\in\Pi_2(K)} \in \mathbb{R}^{d\times K(K-1)/2}$, where $\mathbf{z}_{i,j,k} = \mathbf{x}_{i,j} - \mathbf{x}_{i,k}$ denotes the difference of feature vectors of items j and k in list i. Second, we collect human feedback for n rounds. At round $t \in [n]$, we sample a list $I_t \sim \pi_*$ and then observe σ_t drawn from the PL model, as defined in (7.2). Third, we estimate the model parameter as

$$\widehat{\theta}_n = \operatorname*{arg\,min}_{\theta\in\Theta} \ell_n(\theta), \quad \ell_n(\theta) = -\frac{1}{n}\sum_{t=1}^{n}\sum_{k=1}^{K} \log\left(\frac{\exp[\mathbf{x}_{I_t,\sigma_t(k)}^\top \theta]}{\sum_{j=k}^{K}\exp[\mathbf{x}_{I_t,\sigma_t(j)}^\top \theta]}\right), \tag{7.10}$$

where Θ is defined in assumption 10. We solve this estimation problem using *iteratively reweighted least squares* (*IRLS*) (Wolke and Schwetlick, 1988), a popular method for fitting the parameters of *generalized linear models* (*GLMs*). Finally, we sort the items in all lists i according to their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\theta}_n$ in descending order, to obtain the permutation $\hat{\sigma}_{n,i}$. The pseudo-code of Dope is in algorithm 9.

The optimal design for (7.10) is derived as follows. First, we derive the Hessian of $\ell_n(\theta)$, $\nabla^2\ell_n(\theta)$, in theorem F.3. The optimal design with $\nabla^2\ell_n(\theta)$ cannot be solved exactly because $\nabla^2\ell_n(\theta)$ depends on an unknown model parameter θ. To get around this, we eliminate θ-dependent terms by bounding them from below. Many prior works on decision mak-

ing under uncertainty with GLMs took this approach (Filippi et al., 2010b; Li et al., 2017a; Zhu et al., 2023b; Das et al., 2024; Zhan et al., 2024). We derive normalized and unnormalized covariance matrices

$$\boldsymbol{\Sigma}_n = \frac{2}{K(K-1)n}\overline{\boldsymbol{\Sigma}}_n\,, \quad \overline{\boldsymbol{\Sigma}}_n = \sum_{t=1}^{n}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbf{z}_{I_t,j,k}\mathbf{z}_{I_t,j,k}^\top\,, \tag{7.11}$$

and prove that $\nabla^2\ell_n(\theta) \succeq \gamma\boldsymbol{\Sigma}_n$ for some $\gamma > 0$. Therefore, we can maximize $\log\det(\nabla^2\ell_n(\theta))$, for any $\theta \in \boldsymbol{\Theta}$, by maximizing $\log\det(\boldsymbol{\Sigma}_n)$. The matrix for list i, $\mathbf{A}_i$, can be related to the inner sum in (7.11) through $\mathbf{Tr}(\mathbf{A}_i\mathbf{A}_i^\top) = \sum_{j=1}^{K}\sum_{k=j+1}^{K}\mathbf{z}_{i,j,k}\mathbf{z}_{i,j,k}^\top$.

The cost for our approximation is a constant factor of $C > 0$ in our bounds (theorems 7.5 and 7.6). In section F.3, we discuss a more adaptive design and also compare to it empirically. We conclude that it would be harder to implement and analyze, and we do not observe empirical benefits at $K = 2$.

Maximum Prediction Error Under Ranking Feedback

In this section, we bound the maximum prediction error of Dope under ranking feedback. Similarly to the proof of theorem 7.3, we decompose the error into two parts, which capture the efficiency of the optimal design and the uncertainty in the MLE $\widehat{\theta}_n$.

Theorem 7.5 (Maximum prediction error)**.** *With probability at least* $1-\delta$*, the maximum prediction error after* n *rounds is*

$$\max_{i\in[L]} \mathbf{Tr}(\mathbf{A}_i^\top(\widehat{\theta}_n - \theta_*)(\widehat{\theta}_n - \theta_*)^\top\mathbf{A}_i) = O\left(\frac{K^6(d^2 + d\log(1/\delta))}{n}\right).$$

This theorem is proved in section F.1. We build on a self-normalizing bound of Zhu et al. (2023b), $\|\widehat{\theta}_n - \theta_*\|^2_{\boldsymbol{\Sigma}_n} \leqslant O\left(\frac{K^4(d+\log(1/\delta))}{n}\right)$, which may not be tight in K. If the bound could be improved by a multiplicative

$c > 0$, we would get a multiplicative c improvement in theorem 7.5. We remind the reader that if the allocations $n\pi_*(i)$ are not integers, a rounding procedure is needed (Pukelsheim, 2006; Fiez et al., 2019; Katz-Samuels et al., 2020). This would result in a multiplicative $1+\beta$ factor in our bound, for some $\beta > 0$. For simplicity, we omit this factor in our derivations.

Ranking Loss Under Ranking Feedback

In this section, we bound the expected ranking loss under ranking feedback. Similarly to section 7.4, we define the *gap* between the mean rewards of items j and k in list i as $\Delta_{i,j,k} = \mathbf{z}_{i,j,k}^\top \theta_*$, where $\mathbf{z}_{i,j,k} = \mathbf{x}_{i,j} - \mathbf{x}_{i,k}$ is the difference of feature vectors of items j and k in list i.

Theorem 7.6 (Ranking loss)**.** *The expected ranking loss after n rounds is bounded as*

$$\mathbb{E}[R_n] \leqslant \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \exp\left[-\frac{\Delta_{i,j,k}^2 n}{CK^4 d} + d\right],$$

where $C > 0$ is a constant.

Proof. The proof is similar to theorem 7.4. At the end of round n, we bound the probability that a sub-optimal item k is ranked above item j. The proof has two parts. First, for any list $i \in [L]$ and items $(j,k) \in \Pi_2(K)$, we show that $\mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\theta}_n < \mathbf{x}_{i,k}^\top \widehat{\theta}_n\right) = \mathbb{P}\left(\mathbf{z}_{i,j,k}^\top (\theta_* - \widehat{\theta}_n) > \Delta_{i,j,k}\right)$. Then we bound this quantity by $\exp\left[-\frac{\Delta_{i,j,k}^2 n}{CK^4 d} + d\right]$. The full proof is in section F.1. □

The bound in theorem 7.6 is similar to that in theorem 7.4, with the exception of multiplicative K^{-4} and additive d. The leading term inside the sum can be bounded by $\exp\left[-\frac{\Delta_{\min}^2 n}{CK^4 d}\right]$, where n is the sample size, d is the number of features, and $\Delta_{\min}$ is the minimum gap. Therefore, similarly to theorem 7.4, the bound decreases exponentially with budget n and gaps; and increases with d. This dependence is similar to Theorem 2 of

Azizi et al. (2022) for fixed-budget best-arm identification in GLMs. Our bound does not involve the extra factor of $\kappa > 0$ because we assume that all vectors lie in a unit ball (assumption 10).

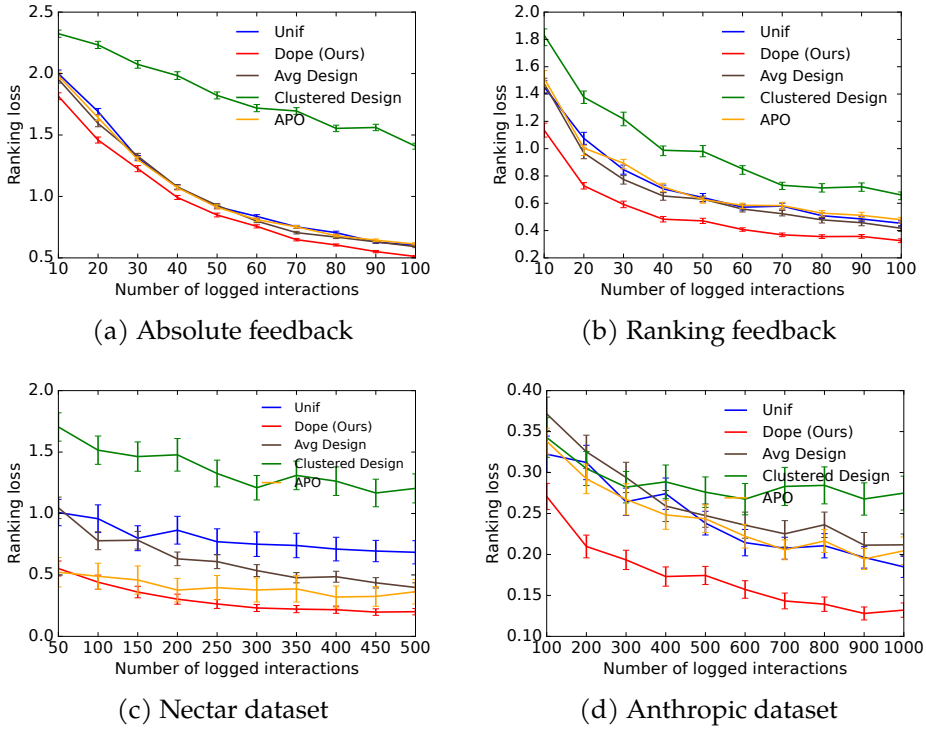
7.6 Experiments

The goal of our experiments is to evaluate Dope empirically and compare it to baselines. All methods estimate $\widehat{\theta}_n$ using (7.7) or (7.10), depending on the feedback. To guarantee that these problems are well defined, even when the sample covariance matrix is not full rank, we regularize both objectives with $\gamma\|\theta\|_2^2$, for a small $\gamma > 0$. This mostly impacts small sample sizes. Specifically, since the optimal design leads to policies that collect diverse feature vectors, the sample covariance matrix is likely to be full rank when the sample size is large. After $\widehat{\theta}_n$ is estimated, each method ranks items in all lists based on their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\theta}_n$. The performance of all methods is measured by their ranking loss in (7.3) divided by L. All experiments are averaged over 100 independent runs, and we report results in fig. 7.1. We compare the following algorithms:

(1) Dope: This is our method. We solve the optimal design problem in (7.6) and then sample lists I_t according to π_*.

(2) Uniform: This baseline chooses lists I_t uniformly at random from $[L]$. While simple, it is known to be competitive in real-world problems where feature vectors may cover the feature space close to uniformly (Ash et al., 2019; Yuan et al., 2020; Ash et al., 2021; Ren et al., 2021).

(3) Avg-Design: The exploration policy is an optimal design over feature vectors. The feature vector of list i is the mean of the feature vectors of all items in it, $\bar{\mathbf{x}}_i = \frac{1}{K}\sum_{k=1}^K \mathbf{x}_{i,k}$. After the design is computed, we sample lists I_t according to it. The rest is the same as in Dope. This baseline shows that our list representation with multiple feature vectors can outperform more naive choices.



(a) Absolute feedback

(b) Ranking feedback

(c) Nectar dataset

(d) Anthropic dataset

Figure 7.1: Ranking loss of all compared methods plotted as a function of the number of rounds. The error bars are one standard error of the estimates.

(4) Clustered-Design: This approach uses the same representation as Avg-Design. The difference is that we cluster the lists using k-medoids. Then we sample lists I_t uniformly at random from the cluster centroids. The rest is the same as in Avg-Design. This baseline shows that Dope outperforms other notions of diversity, such as obtained by clustering. We tune k ($k = 10$ in the Nectar dataset and $k = 6$ otherwise) and report only the best results.

(5) APO: This method was proposed in Das et al. (2024) and is the closest related work. APO greedily minimizes the maximum error in pairwise ranking of L lists of length $K = 2$. We extend it to $K > 2$ as follows. First, we turn L lists of length K into $\binom{K}{2}$L lists of length 2, one for

each pair of items in the original lists. Then we apply APO to these $\binom{K}{2}L$ lists of length 2.

Pure exploration algorithms are often compared to cumulative regret baselines (Bubeck et al., 2009; Audibert et al., 2010). Since our problem is a form of learning to rank, *online learning to rank* (*OLTR*) baselines (Radlinski et al., 2008; Kveton et al., 2015; Zong et al., 2016) seem natural. We do not compare to them for the following reason. The problem of an optimal design over lists is to design a distribution over queries. All OLTR algorithms solve a different problem, return a ranked list of items conditioned on a query chosen by the environment. Since they do not choose the queries, they cannot solve our problem.

Synthetic experiment 1 (absolute feedback): We have $L = 400$ questions and represent them by random vectors $\mathbf{q}_i \in [-1, 1]^6$. Each question has $K = 4$ answers. For each question, we generate K random answers $\mathbf{a}_{i,k} \in [-1, 1]^6$. Both the question and answer vectors are normalized to unit length. For each question-answer pair (i, k), the feature vector is $\mathbf{x}_{i,k} = \mathrm{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$ and has length $d = 36$. The outer product captures cross-interaction terms of the question and answer representations. A similar technique has been used for feature preprocessing of the Yahoo! Front Page Today Module User Click Log Dataset (Li et al., 2010, 2011; Zhu et al., 2021; Baek and Farias, 2023). We choose a random $\theta_* \in [0, 1]^d$. The absolute feedback is generated as in (7.1). Our results are reported in fig. 7.1a. We note that the ranking loss of Dope decreases the fastest among all methods, with Uniform, Avg-Design, and APO being close second.

Synthetic experiment 2 (ranking feedback): This experiment is similar to the first experiment, except that the feedback is generated by the PL model in (7.2). Our results are reported in fig. 7.1b and we observe again that the ranking loss of Dope decreases the fastest. The closest baselines are Uniform, Avg-Design, and APO. Their lowest ranking loss ($n = 100$) is attained by Dope at $n = 60$, which is nearly a two-fold reduction in

sample size. In section F.4, we conduct additional studies on this problem. We vary the number of lists L and items K, and report the computation time and ranking loss.

Experiment 3 (Nectar dataset): The Nectar dataset (Zhu et al., 2023a) is a dataset of 183k questions, each with 7 answers. We take a subset of this dataset: $L = 2\,000$ questions and $K = 5$ answers. The answers are generated by GPT-4, GPT-4-0613, GPT-3.5-turbo, GPT-3.5-turbo-instruct, and Anthropic models. We embed the questions and answers in 768 dimensions using Instructor embeddings (Su et al., 2022). Then we project them to $\mathbb{R}^{10}$ using a random projection matrix. The feature vector for answer k to question i is $\mathbf{x}_{i,k} = \text{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$, where $\mathbf{q}_i$ and $\mathbf{a}_{i,k}$ are the projected embeddings of question i and answer k, respectively. Hence $d = 100$. The ranking feedback is simulated using the PL model in (7.2). We estimate its parameter $\theta_* \in \mathbb{R}^d$ from the ranking feedback in the dataset using the MLE in (7.10). Our results are reported in fig. 7.1c. We observe that the ranking loss of Dope is the lowest. The closest baseline is APO. Its lowest ranking loss ($n = 500$) is attained by Dope at $n = 150$, which is more than a three-fold reduction in sample size.

Experiment 4 (Anthropic dataset): The Anthropic dataset (Bai et al., 2022) is a dataset of 161k questions with two answers per question. We take a subset of $L = 2\,000$ questions. We embed the questions and answers in 768 dimensions using Instructor embeddings (Su et al., 2022). Then we project them to $\mathbb{R}^6$ using a random projection matrix. The feature vector for answer k to question i is $\mathbf{x}_{i,k} = \text{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$, where $\mathbf{q}_i$ and $\mathbf{a}_{i,k}$ are the projected embeddings of question i and answer k, respectively. Hence $d = 36$. The ranking feedback is simulated using the PL model in (7.2). We estimate its parameter $\theta_* \in \mathbb{R}^d$ from the ranking preference feedback in the dataset using the MLE in (7.10). Our results are reported in fig. 7.1d. We note again that the ranking loss of Dope is the lowest. The closest baselines are Uniform, Avg-Design, and APO. Their lowest ranking loss

($n = 1\,000$) is attained by Dope at $n = 300$, which is more than a three-fold reduction in sample size.

7.7 Conclusions

We study the problem of optimal human preference elicitation for learning preference models. The problem is formalized as learning to rank K answers to L questions under a budget on the number of asked questions. We consider two feedback models: absolute and ranking. The absolute feedback is motivated by how humans assign relevance judgments in search (Hofmann et al., 2013; MS MARCO, 2016). The ranking feedback is motivated by learning reward models in RLHF (Kaufmann et al., 2024; Rafailov et al., 2023; Kang et al., 2023; Casper et al., 2023; Shen et al., 2023b; Chen et al., 2023). We address both settings in a unified way. The key idea in our work is to generalize optimal designs (Kiefer and Wolfowitz, 1960; Lattimore and Szepesvári, 2020a), a methodology for computing optimal information-gathering policies, to ranked lists. After the human feedback is collected, we learn preference models using existing estimators. Our method is statistically efficient, computationally efficient, and can be analyzed. We bound its prediction errors and ranking losses, in both absolute and ranking feedback models, and evaluate it empirically to show that it is practical.

Our work can be extended in several directions. First, we study only two models of human feedback: absolute and ranking. However, many feedback models exist (Jeon et al., 2020). One common property of these models is that learning of human preferences can be formulated as likelihood maximization. In such cases, an optimal design exists and can be used for human preference elicitation, exactly as in our work. Second, while we bound the prediction errors and ranking losses of Dope, we do not derive matching lower bounds. Therefore, although we believe that

Dope is near optimal, we do not prove it. Third, we want to extend our methodology to the fixed-confidence setting. Finally, we want to apply our approach to learning a reward model in the LLM and evaluate it.

8 OPTIMAL DESIGN FOR ADAPTIVE IN-CONTEXT PROMPT DESIGN IN LARGE LANGUAGE MODELS

Large language models (LLMs), such as Vicuna (Chiang et al., 2023), Falcon-40B (Penedo et al., 2023), and OpenLLaMA (Touvron et al., 2023) are applied in mainly two ways: fine-tuning and prompt designing. In fine-tuning, the LLM weights are adapted to a downstream task (Devlin et al., 2018). Fine-tuning can easily incorporate domain knowledge that a pre-trained model may not possess and resembles classic inductive inference. Fine-tuned models often do not need carefully designed prompts, which makes them easier to deploy. The main drawback of fine-tuning is that it can be costly, because tens of thousands of training examples may be needed to fine-tune billions of parameters of the LLM (Ding et al., 2023). In prompt designing, the LLM weights are fixed and the LLM is given query-specific examples at inference time that affect its output (Lester et al., 2021). This ability to conduct in-context inference is one of the emergent abilities of LLMs. Prompt designing does not require large training sets. It is also preferred when query-specific examples are private or change over time, and thus can only be utilized at inference time.

Prior works on prompt designing mainly focus on hard prompts, which are carefully handcrafted to get the desired output. This can be time-consuming and fragile, as minor prompt modifications can lead to a significant performance drop on the downstream task (Suzgun et al., 2022). In contrast, Zhang et al. (2022a,b) and Diao et al. (2023) explored adaptive prompt design using clustering-based and uncertainty-reducing approaches. While these approaches offer some benefits, we argue that optimal designs (Pukelsheim, 2006; Fedorov, 2013) can outperform them by effectively balancing uncertainty and diversity. Similarly to Zhang et al. (2022a,b) and Diao et al. (2023), we propose a framework for adaptive prompt design called *active in-context prompt design* (*AIPD*). The key idea is

to design the LLM prompt by adaptively choosing few-shot examples for a set of test examples at inference time. The examples are initially unlabeled and we obtain labels for the most informative ones, which maximally reduce the uncertainty in the LLM prediction for all test examples. We assume that the observed labels are collected from experts (human-in-the-loop) or revealed by an oracle (Dasgupta, 2005; Hanneke et al., 2014). The focus on informativeness and diversity ensures efficient label acquisition by selecting the best examples. This reduces reliance on limited and costly resources such as expert labeling.

One motivating example for our work is *theme recognition*, where the goal is to identify a unifying theme for a set of items (e.g., movies, grocery items, or books) provided by the user. For example, let the test query be a triplet of movie titles "Lion King", "Jungle Book", and "Tarzan", and the goal is that the LLM should infer a plausible common theme such as "Disney animated movies", "Children's movies", or "Movies with deep connections with nature". This task is challenging due to the inherent ambiguity and many plausible themes. To address this, we can give the LLM a few informative examples of triplets of movies and their common themes as training examples in context that can guide it towards the correct theme for the test query. This inherently requires a human-in-the-loop who can go over the set of triplets of movies and label their common theme for each example which can be costly. Hence, it is critical to narrow down to a few informative examples from exponentially many training examples possible for vast amounts of data like movies. Finally note that by exposing the LLM to these training examples, we refine its understanding of the task, improve handling of ambiguity, and thus improve its ability to identify the common theme for unseen test examples. To address the above challenges, we propose a framework for adaptive prompt design called active in-context prompt design (AIPD). Our framework is general and can be easily extended to any active supervised-learning task, like active

regression (Gao and Koller, 2011) and active classification (Gao and Koller, 2011). At a high level, we treat the LLM as a general inference machine (Brown et al., 2020; Mirchandani et al., 2023) that is shown adaptively-chosen examples with labels at inference time. The LLM then utilizes them to answer any set of related test examples. The key idea is to choose the next example to label such that we maximally reduce the estimated uncertainty of the answer to the test examples. We focus on designing algorithms with the following two properties: **(1)** Implementable in any LLM that can be queried efficiently. The parameters of the LLM do not change or have to be observed. **(2)** Analyzable in simple models. In this work, we use linear models to motivate and analyze our algorithms.

We now state the main contributions of our work:

(1) We propose a **G**-**O**ptimal design algorithm (GO). The key idea in GO is to retrieve the examples to label that are closest to the set of test examples in the inference task. Our main contribution is the right notion of closeness, based on posterior covariance in a simpler model. GO is implementable with any LLM that can be sampled from, and does not require access to model parameters, feature embeddings of the LLM, or its gradients.

(2) We propose a **S**imulation-Based **A**ctive **L**earning algorithm (SAL). SAL uses simulation to estimate the impact of labeling unlabeled examples on uncertainty of the example in the inference task. SAL is also implementable with any LLM that can be sampled from.

(3) GO is motivated by optimal designs in linear models (Kiefer and Wolfowitz, 1960; Pukelsheim, 2006). This allows us to analyze GO in linear models (Theorem 8.1). Our proof is a major departure from similar analyses in fixed-budget best-arm identification in bandits (Azizi et al., 2022; Yang and Tan, 2022), for instance because we directly analyze the discrete allocation problem and each unlabeled example can be labeled at most once. We discuss this in detail right after Theorem 8.1. SAL is more

general than GO because it does not make any linear model assumption in its design. We show that SAL and GO are equivalent in linear models in Theorem 8.2.

(4) We evaluate GO and SAL on UCI (Markelle Kelly, 1988) and OpenML (Vanschoren et al., 2013) regression and classification tasks, custom NLP datasets, abstract reasoning corpus (ARC) tasks (Alford, 2021; Mirchandani et al., 2023), and Probabilistic Context Free Grammar (PCFG) tasks (Hupkes et al., 2020). GO and SAL consistently outperform other active prompt designing methods (Zhang et al., 2022a,b; Diao et al., 2023) for choosing few-shot examples in majority of the tasks.

We advance the understanding of active in-context prompt design in LLMs and develop a practical methodology for adaptive prompt design. To our knowledge, this is the first paper that analyzes optimal design based prompting approaches that correctly balance uncertainty and diversity-based sampling as opposed to other existing adaptive prompting-based approaches (Zhang et al., 2022a,b; Diao et al., 2023).

This chapter is organized as follows. section 8.1 introduces the problem setting. section 8.3 presents our methods and discusses their properties. section 8.4 is devoted to analyzing our methods. section 8.5 validates our approach empirically. We review related work in detail in section 8.2. Finally, section 8.6 summarizes our contributions and suggests avenues for future work.

8.1 Setting

We pose the problem of adaptive prompt design as active learning. We adopt the following standard active learning terminology (Lewis, 1995; Tong and Koller, 2001; Dasgupta, 2005; Dasgupta et al., 2007; Hanneke et al., 2014). We have a d-dimensional *feature space* $\mathcal{X} \subset \mathbb{R}^d$ and a d_y-dimensional *label space* $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$. A labeled example is a pair $(\mathbf{x}, Y) \in \mathcal{X} \times \mathcal{Y}$.

The feature vectors and labels are related as $Y = f(\mathbf{x}, \theta_*) + \varepsilon$, where f is an underlying model, θ_* is its parameter, and ε is an *independent zero-mean noise vector*. Our goal is to learn to estimate f on test examples by labeling training examples. We have a budget T on the maximum number of training examples that can be labeled. This constraint can arise due to multiple reasons. For instance, human labels may be necessary and they are naturally costly. Another reason may be that the machine learning model has a limited capacity for including labeled examples, such as the length of prompts in LLMs (Zhang et al., 2022a,b; Diao et al., 2023).

Now we introduce our notation in detail. Denote $[m] = \{1, 2, \ldots, m\}$. We have n *training examples* $\mathcal{X}_{\text{examples}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and K *test examples* $\mathcal{X}_* = \{\mathbf{x}_{*,1}, \ldots, \mathbf{x}_{*,K}\}$. We assume that both sets are related, such as being sampled from the same distribution. The label of the test example $\mathbf{x}_{*,k}$ is $Y_{*,k}$. In our motivating theme recognition example the training examples $\mathbf{x}_i$ and test example $\mathbf{x}_{*,k}$ is a concatenation of triplets of movies, and the label Y_i or $Y_{*,k}$ is the common theme amongst the triplets respectively. We want to infer $Y_{*,k}$ for all $k \in [K]$ without explicitly modeling the complex function f. We model the function using an LLM which we treat as a general inference machine because of its large representation capacity (Brown et al., 2020; Mirchandani et al., 2023). Specifically, let $H_t = \{(X_\ell, Y_\ell)\}_{\ell \in [t-1]}$ be a set of $t-1$ previously labeled examples, where $X_\ell \in \mathcal{X}_{\text{examples}}$ is the ℓ-th labeled example and Y_ℓ is its label. Then we denote by $p(\cdot \mid \mathbf{x}, H_t)$ the distribution over labels of an LLM for a queried example $\mathbf{x}$ when H_t is used as few-shot in-context examples. To implement this in the LLM, we simply concatenate $\mathbf{x}$ and H_t in context (Zhang et al., 2022a,b; Diao et al., 2023). We know that in-context examples affect the distribution of responses of an LLM (Xie et al., 2021; Suzgun et al., 2022; Deng et al., 2023; Lee et al., 2023). So, the problem of learning f under a budget T can be viewed as selecting H_{T+1} such that $p(Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1})$ is high for all test examples $k \in [K]$. This problem is challenging, especially when the

training examples need to be labeled.

To effectively reduce the uncertainty of $Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}$, we need to quantify it. One possibility it to use the entropy $-\mathbb{E}_{Y_{*,k} \sim p(\cdot \mid \mathbf{x}_{*,k}, H_{T+1})}[\log p(Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1})]$. This is problematic because the entropy is hard to estimate for high-dimensional random variables (Vershynin, 2020), especially without having access to $p(\cdot \mid \mathbf{x}_{*,k}, H_{T+1})$ beyond sampling from it. This is a shortcoming of recent adaptive prompting techniques (Zhang et al., 2022a,b; Diao et al., 2023). Therefore, we propose using the covariance of $Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}$ as the uncertainty measure. Specifically, we measure the uncertainty of the k-th test example by $\mathrm{tr}(\mathrm{cov}[Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}])$ and the uncertainty over all test examples by $\max_{k \in [K]} \mathrm{tr}(\mathrm{cov}[Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}])$. Since the trace of the covariance is the sum of the variances in individual dimensions, our objective can be interpreted as minimizing the maximum variance over the predicted labels of all test examples. This is a natural measure of uncertainty in linear models and corresponding optimal designs (Pukelsheim, 2006; Fedorov, 2013).

Before we present our algorithms, we wanted to outline their general design. Given a budget T, we design sequential adaptive algorithms over T rounds, where the example $X_t \in \mathcal{X}_{\text{examples}}$ in round $t \in [T]$ is chosen as a function of $H_t = \{(X_\ell, Y_\ell)\}_{\ell \in [t-1]}$ up to that round. Since H_t summarizes past actions of the algorithm, we call it a *history*. The label of example X_t is $Y_t = f(X_t, \theta_*) + \varepsilon_t$, where ε_t is an independent zero-mean noise vector in round t. Our objective is to minimize the maximum uncertainty over all test examples, $\max_{k \in [K]} \mathrm{tr}(\mathrm{cov}[Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}])$.

8.2 Related Work

We study the problem of choosing human demonstrations adaptively to get the desired output from the LLM as quickly as statistically possible. We use active learning to choose them and then ask a human to label them.

Finally, the human demonstrations are fed as in-context input to the LLM together with the main user query, to obtain the desired output. Thus the name is active transductive inference. Prior works on prompt-tuning and transductive inference (Lester et al., 2021; Dong et al., 2022; Zhang et al., 2022a; Min et al., 2022; Wu et al., 2022; Yu et al., 2022; Suzgun et al., 2022; Liu et al., 2023a; Yu et al., 2023; Liu et al., 2023d) focus on hard prompt-tuning where the user must carefully handcraft the prompt to get the desired output for tasks like movie recommendation, disambiguation QA, navigation, etc. Such examples of carefully handcrafting hard prompts with demonstrations can be found in Suzgun et al. (2022); Srivastava et al. (2022). These papers also show how failing to design such prompts with demonstrations can lead the LLM to predict wrong outputs. Note that we adaptively design the prompt through carefully chosen demonstrations. In our experiments, we show significant improvement over randomly chosen demonstrations.

The problem of active learning is also related to *dataset augmentation* (Dukler et al., 2021). In this work, the most informative unlabeled examples are chosen by optimizing the error of the model on the validation set. The gradient of the validation set error with respect to the weights of unlabeled examples has a closed form when the original model is linearized. The main difference in active learning, including in our work, is that labeling all unlabeled examples would be costly. Therefore, the labels are not available in advance and are queried adaptively. We also learn in context and do not assume that the gradient information of the LLM is available. Prompt composition has also been an active area of research. Bowman et al. (2023) proposed a-la-carte prompt tuning, where prompts are tuned on individual datasets and composed at inference time to mimic the performance of the model that would have been trained on the union of the corresponding datasets. This idea has been further extended by Perera et al. (2023), where prompts for previously unseen tasks

are obtained by linearly combining prompts from known tasks. To do this, they use spectral decomposition and project prompts from known tasks to a lower dimensional space. In our work, we do not tune prompts or compose simpler models. We actively probe an LLM, treated as a black box without any extra side information, to answer a test example as accurately as possible, with as little variance in the answer as possible.

Recently there has been a lot of progress in prompt tuning (or aligning). Hassan et al. (2023) studies prompt aligning for a single test sample to adapt to multi-modal test prompts at test time by minimizing the feature distribution shift to the test domain. In contrast in this paper, we study adapting prompts for many test samples without the feature distribution shift assumption. The Wang et al. (2023a) trains a smaller LLMto select demonstrations for a larger LLM. However, we rely on active learning to select the smallest number of informative prompts to be labeled by human labelers. This avoids finetuning a smaller LLM for individual tasks. Wang et al. (2023d) studies transductive inference for diffusion models for a different setting where given a pair of task-specific example images, such as depth from/to image and scribble from/to image, and text guidance, the model automatically understands the underlying task and performs the same task on a new query image following the text guidance. However, in our setting, we do not explicitly encode any guidance text. The Wen et al. (2023) proposes to mix the nearest neighbor method with gradient optimization to select prompts during test time. Similarly, Zhang et al. (2023) proposes a nearest neighbor approach to select in-context examples for computer vision tasks. We compare our approach against such nearest-neighbor selection algorithms. Finally, Bai et al. (2023); Wang et al. (2023b) analyze transductive inference theoretically to understand its universality, generalization capability, and limitations. In contrast, we only do a theoretical analysis of AIPD to show it maximally reduces the estimated variance of the answer to the user's query. The Zhang et al.

(2022b) study the chain-of-thought prompting where they automatically select the prompt using a clustering-based approach.

There are some related works in the area of medical diagnosis chatbot examples that we shared in the introduction. One notable study Caruccio et al. (2024), although not utilizing machine learning techniques, provides valuable insights with its implementation of three hardcoded prompt designs for accurate diagnosis. These designs, however, do not engage in active learning, as they lack the capability to adapt based on user input. Similarly, Kuroiwa et al. (2023) gives more insights into self-diagnostics of orthopedic diseases using ChatGPT. In contrast, web applications such as Buoy Health BuoyHealth and Live Healthily livehealthily employ a more dynamic approach, actively tailoring subsequent questions based on users' responses. This aligns with active learning principles but it is not clear what techniques they apply and is notably underexplored in academic literature, indicating a potential area for further research. Our setting also goes beyond the single shot active prompt tuning studied in Margatina et al. (2023). Note that this work studies prompt tuning only for one iteration, and does not take into account the historical context. So it has limited ability for complex tasks like ARC (Alford, 2021) and PCFG (Hupkes et al., 2020) as well as handling vector labels like GO and SAL.

Active Learning (AL): Recently, there has been a lot of focus on using deep AL to finetune LLMs. All AL algorithms tend to balance uncertainty and diversity in the selection of unlabeled examples. We briefly discuss them below and also highlight the main difference of these approaches with prompt aligning with GO and SAL.

(1) Coreset: This is a pure diversity-based approach using a coreset selection. In every iteration, first, the embedding of each unlabeled example is computed from the network's penultimate layer, and then unlabeled examples are selected using a greedy furthest-first traversal conditioned on all labeled examples (Sener and Savarese, 2017; Geifman and El-Yaniv,

2017; Citovsky et al., 2021). Observe that in our setting we do not have access to the penultimate layer of the LLM.

(2) Least: This is an uncertainty-based active learning algorithm. Here, the uncertainty score of an unlabeled example is its predicted class probability. At every iteration, this algorithm then samples unlabeled examples with the smallest uncertainty scores (Settles, 2009, 2011; Wang and Shang, 2014).

(3) Margin: This is also an uncertainty-based active learning algorithm (Tong and Koller, 2001; Balcan et al., 2009; Settles, 2009). At every iteration t it selects unlabeled examples that are sorted according to their multiclass margin score and then selects unlabeled examples that are the hardest to discriminate and can be thought of as examples closest to their class margin. However, in our setting, we do not have any information on the hypothesis space of the LLM and hence cannot implement such a baseline.

(4) Entropy: This is also an uncertainty-based active learning algorithm Wang and Shang (2014); Kremer et al. (2014); Diao et al. (2023). At every iteration t it selects unlabeled examples according to the entropy of the example's predictive class probability distribution. We show that Greedy-NN-max-mean is outperformed significantly by GO and SAL in the prediction, pcfg, or arc tasks.

(5) Badge: This is an algorithm that combines both uncertainty and diversity sampling (Ash et al., 2019, 2021). For each unlabeled example $\mathbf{x}$ its gradient embedding $g_{\mathbf{x}}$ is computed with respect to the parameters of the model's penultimate layer. Finally, Badge chooses a batch of samples to sample by applying k-Means++ (Arthur and Vassilvitskii, 2006) on the gradient embeddings. Again recall that we cannot implement such a baseline as we do not have access to the LLMs last layer.

(6) Badge-KM: This algorithm is similar to Badge but in the final step instead of k-Means++ it uses k-Means on the gradient embeddings. In Yuan et al. (2020) it is observed that applying k-Means on the embeddings

results in an increase in accuracy over baselines in some datasets. Further Yuan et al. (2020) observed from the t-SNE plots that k-Means select centers that are further apart compared to the ones chosen by k-Means++ which leads to more diverse sampling in batches.

(7) Bald: Bayesian Active Learning by Disagreements (Kirsch et al., 2019; Gal et al., 2017) chooses unlabeled examples that are expected to maximize the information gained from the model parameters θ_t, i.e. the mutual information between predictions and model posterior.

A more comprehensive survey on how AL is used for finetuning deep models can be found in Ren et al. (2021); Zhan et al. (2022). The Bhatt et al. (2024) study how experimental design can be used to select prompts for finetuning a pre-trained LLM. Some recent works have also focused on selecting unlabeled examples only within a task Wei et al. (2021); Chen et al. (2023); Fifty et al. (2021). However, these works are geared towards selecting prompts within a task for finetuning, whereas we focus on adaptive prompt design using experimental design. The work of Perlitz et al. (2023) also uses AL for finetuning prompts for LLMsto improve label efficiency. The Kung et al. (2023) proposes an AL framework for instruction tuning. However, their approach again focuses on selecting unlabeled examples inside each task and discriminating one task from another. However, they make the simplifying assumption that all unlabeled examples inside the tasks are equally informative which may inhibit the quality of the selected subset.

8.3 Algorithms

In this section, we introduce our active learning algorithms for selecting most informative training examples from $\mathcal{X}_{\text{examples}}$. To simplify notation, we assume scalar labels and then discuss an extension to vector labels at the end of the section. We also let $\mathcal{L}_t \subseteq [n]$ and $\mathcal{U}_t \subseteq [n]$ be the indices of

Algorithm 10 G-optimal design (GO)

Input: Training set $\mathcal{X}_{\text{examples}} = \{\mathbf{x}_i\}_{i=1}^n$, test set $\mathcal{X}_* = \{\mathbf{x}_{*,k}\}_{k=1}^K$, budget T
$\mathcal{L}_1 \leftarrow \emptyset$, $\mathcal{U}_1 \leftarrow [n]$, $H_1 \leftarrow \{\}$
for $t = 1, \ldots, T$ **do**
    $I_t = \arg\min_{i \in \mathcal{U}_t} \max_{k \in [K]} \mathbf{x}_{*,k}^\mathsf{T} \left(\widehat{\mathbf{\Sigma}}_t^{-1} + \mathbf{x}_i \mathbf{x}_i^\mathsf{T} \right)^{-1} \mathbf{x}_{*,k}$
    $X_t \leftarrow \mathbf{x}_{I_t} \in \mathcal{X}_{\text{examples}}$
    Observe label Y_t of example X_t
    $\mathcal{L}_{t+1} \leftarrow \mathcal{L}_t \cup \{I_t\}$, $\mathcal{U}_{t+1} \leftarrow \mathcal{U}_t \setminus \{I_t\}$
    $H_{t+1} \leftarrow H_t \cup \{(X_t, Y_t)\}$
Output: Sample $Y_{*,k} \sim p(\cdot \mid \mathbf{x}_{*,k}, H_{T+1})$ for $k \in [K]$

all labeled and unlabeled training examples up to round t, respectively. Note that $\mathcal{L}_t \cup \mathcal{U}_t = [n]$.

Optimal Design Algorithm

The key idea is to label examples in $\mathcal{X}_{\text{examples}}$ that minimize the maximum uncertainty of predictions over all test examples $\mathbf{x}_{*,k}$. Our computation of uncertainty is borrowed from linear models. Specifically, take a linear model $Y = \mathbf{x}^\mathsf{T}\theta_* + \varepsilon$, where $\theta_* \in \mathbb{R}^d$ is its parameter and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is independent noise. Suppose that $\theta_* \sim \mathcal{N}(\theta_0, \mathbf{\Sigma}_0)$. Then a well-known result in Bayesian statistics (Bishop, 2006b) is that the posterior variance of the model estimate at an example $\mathbf{x}_{*,k}$ given labeled examples H_t is $\mathbf{x}_{*,k}^\mathsf{T} \widehat{\mathbf{\Sigma}}_t \mathbf{x}_{*,k}$, where $\widehat{\mathbf{\Sigma}}_t = (\mathbf{\Sigma}_0^{-1} + \sigma^{-2} \sum_{\ell=1}^{t-1} X_\ell X_\ell^\mathsf{T})^{-1}$ is the posterior covariance of $\theta_* \mid H_t$. Therefore, the maximum uncertainty over test examples is $\max_{k \in [K]} \mathbf{x}_{*,k}^\mathsf{T} \widehat{\mathbf{\Sigma}}_t \mathbf{x}_{*,k}$. The key observation is that this quantity does not depend on labels. Therefore, it can be optimized greedily by choosing the training example that minimizes it the most,

$$I_t = \arg\min_{i \in \mathcal{U}_t} \max_{k \in [K]} \mathbf{x}_{*,k}^\mathsf{T} \left(\widehat{\mathbf{\Sigma}}_t^{-1} + \mathbf{x}_i \mathbf{x}_i^\mathsf{T} \right)^{-1} \mathbf{x}_{*,k}\,, \tag{8.1}$$

where $\mathcal{U}_t$ are indices of all unlabeled training examples up to round t. After the index I_t is chosen, the example $\mathbf{x}_{I_t}$ and its label Y_t are added to the history to get H_{t+1} for the next iteration $t+1$.

This algorithm is a greedy solution to the G-optimal design (Pukelsheim, 2006; Katz-Samuels et al., 2021). We call it **G-O**ptimal design and abbreviate it as GO. The pseudocode of GO is in algorithm 10. Note that GO does not depend on observed Y_t. Similar optimal designs have been effectively applied in active learning (Chaudhuri et al., 2015; Mukherjee et al., 2022b), bandits (Fontaine et al., 2021; Mason et al., 2021), and reinforcement learning (Wagenmaker et al., 2022a). However, this is the first paper that studies optimal design for adaptively designing prompts (Zhang et al., 2022a; Diao et al., 2023). (8.1) can be viewed as choosing that training example $\mathbf{x}_i \in \mathcal{U}_t$ that minimizes the maximum eigenvalue of the posterior covariance $\widehat{\mathbf{\Sigma}}_t$. Therefore this leads to reducing the uncertainty over the model parameter θ_* as the confidence ellipsoid around $\theta_\star$ shrinks (Lattimore and Szepesvári, 2020a). Note that maximum eigenvalue reduction also ensures diversity as it leads to choosing training examples along all directions in $\mathbb{R}^d$.

The time complexity of GO is $O(Kd^2nT)$. This is because, for T rounds, the algorithm searches for the best training example out of at most n and evaluates it on all test examples $\mathbf{x}_{*,k} \in \mathcal{X}_*$. The evaluation of each test example in round t, $\mathbf{x}_{*,k}\mathbf{T}(\widehat{\mathbf{\Sigma}}_t^{-1} + \mathbf{x}_i\mathbf{x}_i\mathbf{T})^{-1}\mathbf{x}_{*,k}$, takes $O(d^2)$ time, because $(\widehat{\mathbf{\Sigma}}_t^{-1} + \mathbf{x}_i\mathbf{x}_i\mathbf{T})^{-1}$ can be computed in $O(d^2)$ time using the Sherman-Morrison formula. In the last step, the LLM is queried K times to return $\{Y_{*,k}\}_{k=1}^K$.

Simulation-Based Algorithm

While GO reduces uncertainty in label predictions, it has a major limitation. The chosen example X_t at round t is not affected by observed labels $(Y_\ell)_{\ell \in [t-1]}$. This is because (8.1) does not depend on $(Y_\ell)_{\ell \in [t-1]}$. While this

Algorithm 11 Simulation-based active learning (SAL)

Input: Training set $\mathcal{X}_{\text{examples}} = \{\mathbf{x}_i\}_{i=1}^{n}$, test set $\mathcal{X}_* = \{\mathbf{x}_{*,k}\}_{k=1}^{K}$, budget T
$\mathcal{L}_1 \leftarrow \emptyset$, $\mathcal{U}_1 \leftarrow [n]$, $H_1 \leftarrow \{\}$
for $t = 1, \ldots, T$ **do**
  for all $i \in \mathcal{U}_t$ **do**
    for all $\mathbf{x}_{*,k} \in \mathcal{X}_*$ **do**
      for $j = 1, 2, \ldots, m$ **do**
        Sample $Y_{t,i}^{(j)} \sim p(\cdot \mid \mathbf{x}_i, H_t)$
        $H_{t,i,j} \leftarrow H_t \cup \{(\mathbf{x}_i, Y_{t,i}^{(j)})\}$
        Sample $\tilde{Y}_{t,i,k}^{(j,1)}, \tilde{Y}_{t,i,k}^{(j,2)} \sim p(\cdot \mid \mathbf{x}_{*,k}, H_{t,i,j})$
  $I_t \leftarrow \arg\min_{i \in \mathcal{U}_t} \max_{k \in [K]} \frac{1}{m} \sum_{j=1}^{m} \left(\tilde{Y}_{t,i,k}^{(j,1)} - \tilde{Y}_{t,i,k}^{(j,2)} \right)^2$
  $X_t \leftarrow \mathbf{x}_{I_t} \in \mathcal{X}_{\text{examples}}$
  Observe label Y_t of example X_t
  $\mathcal{L}_{t+1} \leftarrow \mathcal{L}_t \cup \{I_t\}$, $\mathcal{U}_{t+1} \leftarrow \mathcal{U}_t \setminus \{I_t\}$
  $H_{t+1} \leftarrow H_t \cup \{(X_t, Y_t)\}$
Output: Sample $Y_{*,k} \sim p(\cdot \mid \mathbf{x}_{*,k}, H_{T+1})$ for $k \in [K]$

is a property of linear models, it is undesirable in non-linear models, such as LLMs. To address this limitation, we propose a new algorithm that simulates the impact of labeling examples on the uncertainty of predicted labels. We call it **S**imulation-Based **A**ctive **L**earning and abbreviated it as SAL. The pseudocode of SAL is provided in algorithm 11.

The key idea in SAL is to replace the closed-form formula in (8.1) by a simulation. We detail the algorithm next. Fix round t, history H_t, and candidate example $\mathbf{x}_i$. To estimate the impact of labeling $\mathbf{x}_i$, we simulate its labels m times. For each simulation $j \in [m]$, we sample $Y_{t,i}^{(j)}$ from the conditional distribution $p(\cdot \mid \mathbf{x}_i, H_t)$ using the LLM. Then we extend the history H_t by $\mathbf{x}_i$ and its simulated label $Y_{t,i}^{(j)}$, $H_{t,i,j} = H_t \cup \{(\mathbf{x}_i, Y_{t,i}^{(j)})\}$. This process results in m copies of augmented histories, each reflecting a potential outcome of labeling of $\mathbf{x}_i$. Finally, we take two independent samples for each $j \in [m]$ as $\tilde{Y}_{t,i,k}^{(j,1)}, \tilde{Y}_{t,i,k}^{(j,2)} \sim p(\cdot \mid \mathbf{x}_{*,k}, H_{t,i,j})$. The maximum

uncertainty over test examples after labeling $\mathbf{x}_i$ is estimated as

$$\max_{k\in[K]} \frac{1}{m} \sum_{j=1}^{m} \left(\tilde{Y}^{(j,1)}_{t,i,k} - \tilde{Y}^{(j,2)}_{t,i,k}\right)^2 . \tag{8.2}$$

The training example with the lowest value is chosen and we denote its index by I_t. Then $\mathbf{x}_{I_t}$ and its observed label Y_t are added to the history to get H_{t+1} for the next iteration $t+1$.

Next we justify SAL. Consider the same setting as in section 8.3. Given a label $Y^{(j)}_{t,i}$ for example $\mathbf{x}_i$, the posterior distribution of $\theta_* \mid H_{t,i,j}$ is $\mathcal{N}(\widehat{\theta}_{t,i,j}, \widehat{\mathbf{\Sigma}}_{t,i})$, where $\widehat{\mathbf{\Sigma}}_{t,i} = (\widehat{\mathbf{\Sigma}}_t^{-1} + \sigma^{-2}\mathbf{x}_i\mathbf{x}_i\mathbf{T})^{-1}$ is the simulated posterior covariance of θ_* and

$$\widehat{\theta}_{t,i,j} = \widehat{\mathbf{\Sigma}}_{t,i}\left(\widehat{\mathbf{\Sigma}}_0^{-1}\theta_0 + \sigma^{-2}\left(\sum_{\ell=1}^{t-1} X_\ell Y_\ell + \mathbf{x}_i Y^{(j)}_{t,i}\right)\right)$$

is the posterior mean. By design, $\tilde{Y}^{(j,1)}_{t,i,k}$ and $\tilde{Y}^{(j,2)}_{t,i,k}$ are independent samples from $\mathcal{N}(\mathbf{x}_{*,k}\mathbf{T}\theta_*, \sigma^2)$, where $\theta_* \sim \mathcal{N}(\widehat{\theta}_{t,i,j}, \widehat{\mathbf{\Sigma}}_{t,i})$. Therefore, $\tilde{Y}^{(j,1)}_{t,i,k} - \tilde{Y}^{(j,2)}_{t,i,k} \sim \mathcal{N}(0, 2(\mathbf{x}_{*,k}\mathbf{T}\widehat{\mathbf{\Sigma}}_{t,i}\mathbf{x}_{*,k} + \sigma^2))$. By definition, $(\tilde{Y}^{(j,1)}_{t,i,k} - \tilde{Y}^{(j,2)}_{t,i,k})^2$ is a single sample estimate of $2(\mathbf{x}_{*,k}\mathbf{T}\widehat{\mathbf{\Sigma}}_{t,i}\mathbf{x}_{*,k} + \sigma^2)$ and the sum in (8.2) estimates this quantity from m samples. Note that this estimate is proportional to $\mathbf{x}_{*,k}\mathbf{T}\widehat{\mathbf{\Sigma}}_{t,i}\mathbf{x}_{*,k}$ that appears in the G-optimal design objective in (8.1). Therefore, in linear models, SAL can be viewed as an inefficient implementation of GO. This inefficiency stems from the need to simulate the LLM.

The time complexity of SAL is $O(nKmT)$. This is because it searches for the best example out of at most n in T rounds for each test example $k \in [K]$. The evaluation of impact on each test example requires $2m$ LLM queries.

Vector labels: GO and SAL are easy to extend to vector labels, $d_y > 1$. GO does not depend on labels at all. The only modification in SAL is that (8.2) is replaced with $\max_{k\in[K]} \frac{1}{m}\sum_{j=1}^{m} \|\tilde{Y}^{(j,1)}_{t,i,k} - \tilde{Y}^{(j,2)}_{t,i,k})\|_2^2$. This is the sum

of the posterior variances of the labels over all dimensions.

8.4 Analysis

In this section, we analyze GO and SAL. The analysis is under the assumption that the labels are scalar and hence, our objective simplifies to minimizing $\max_{k\in[K]} \text{var}[Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}]$. The analysis is organized as follows. First, we prove that our objective is decreasing in history but not supermodular, which precludes a straightforward analysis. This property of our objective function is proved in Section G.1 and Section G.1. Second, we analyze GO using the closed form of the posterior covariance $\widehat{\boldsymbol{\Sigma}}_t$. Finally, we prove the equivalence of GO and SAL, and thereby provide guarantees for SAL. All analyses are under the assumption of a linear model with Gaussian noise. These proofs are in Section G.1 and Section G.1.

Analysis of GO

To address challenge posed due to f not being a supermodular (theorem G.3), we leverage the properties of the rank-1 updates in GO. The proof is under the assumption that at round t, the training examples can be partitioned as $\mathcal{X} = S_k \cup \mathbf{S}_k$. The set S_k represents examples that are close to $\mathbf{x}_{*,k}$. The set S_k is convex such that for a $\alpha_k \geqslant 0$ we have $\mathbf{xTy} \geqslant \alpha_k$ for all $\mathbf{x}, \mathbf{y} \in S_k$. In essence, α_k governs the minimum level of similarity required for examples within S_K to be considered similar to the test example $\mathbf{x}_{*,k}$. This is achieved by setting a lower bound on the inner product between any two examples in the set. The set $\mathbf{S}_k$ represents examples that are not close to $\mathbf{x}_{*,k}$. It is defined $\beta_k \geqslant 0$ such that $\mathbf{xTy} \leqslant \beta_k$ for all $\mathbf{x} \in S_k$ and $\mathbf{y} \in \mathbf{S}_k$. In contrast to α_k, β_k limits the maximum similarity any example in S_k can have with examples outside this set. Define $\alpha_{\min} = \min_k \alpha_k$, and $\beta_{\max} = \beta_{\max}$. Define the set $S = \cap_{k=1}^{K} S_k$ as the set of all examples that are

close to all $\{\mathbf{x}_{*,k}\}_{k=1}^{K}$ and $\overline{S} = \cup_{k=1}^{K}\mathbf{S}_k$ as the set of all examples that are not close to all $\{\mathbf{x}_{*,k}\}_{k=1}^{K}$. Assume $S \neq \{\emptyset\}$ and $|S| > T$. With this in hand, we prove the following claim.

Theorem 8.1. *Let* $\alpha_{min}, \beta_{max} \geqslant 0$ *be set such that* $\beta_{max} \geqslant 1 - \alpha_{min}^2$ *and* $T \leqslant \frac{\alpha_{min}^2}{(\beta_{max}+\sqrt{2})\beta_{max}d}$. *Then for any* $\mathbf{x}_{*,k}$ *we can show that* $\mathbf{x}_{*,k}\mathbf{T}\widehat{\mathbf{\Sigma}}_{T+1}\mathbf{x}_{*,k} \leqslant \frac{1}{\alpha_{max}^2 T+1} + (1-\alpha_{max}^2)$.

The proof is in section G.1. It is a major departure from similar proofs in active learning with a fixed budget (Tong and Koller, 2001; Hanneke et al., 2014; Azizi et al., 2022; Yang and Tan, 2022) in three aspects. First, we analyze the discrete allocation problem in (G.1) instead of its continuous optimal-design relaxation (Pukelsheim, 2006). Second, any unlabeled example in $\mathcal{X}$ is labeled at most once. Finally, (G.1) is asymmetric in the sense that we optimize the uncertainty of a single example $\mathbf{x}_*$ over a larger set. To make the analysis manageable, we impose structure on $\mathcal{X}$. The claim in theorem 8.1 holds for any T if $\beta_{max} = 1/(4dn)$ and $\alpha_{min} = \sqrt{1-\beta_{max}}$. In this case, α_{min} is close to 1, and we get a near-optimal $O(1/T)$ decrease in posterior variance.

Analysis of SAL

For a sufficiently large sample size m in SAL, we can establish the following equivalence of SAL and GO.

Theorem 8.2. *Fix a failure probability* $\delta \in (0,1)$. *Define* $\sigma_{t,i,k}^2 = \mathbb{E}[\frac{1}{m}\sum_{j=1}^{m}(\widetilde{Y}_{t,i,k}^{(j,1)} - \widetilde{Y}_{t,i,k}^{(j,2)})^2] = 2\mathbf{x}_{*,k}\mathbf{T}\widehat{\mathbf{\Sigma}}_{t,i}\mathbf{x}_{*,k} + \sigma^2$, *and define* $\sigma_{t,i,max}^2 = \max_{k\in[K]}\sigma_{t,i,k}^2$. *Then for*

any $t \in [T]$ *and* $i \in \mathcal{U}_t$*, we have that*

$$\sigma^2_{t,i,\max}\left[1 - 2\sqrt{\frac{\log(1/\delta)}{m}}\right] \leqslant \max_{k\in[K]} \frac{1}{m}\sum_{j=1}^{m}\left(\widetilde{Y}^{(j,1)}_{t,i,k} - \widetilde{Y}^{(j,2)}_{t,i,k}\right)^2 \leqslant \sigma^2_{t,i,\max}\left[1 + 2\sqrt{\frac{\log(1/\delta)}{m}} + \frac{2\log(1/\delta)}{m}\right].$$

Moreover, for $m \geqslant 8\log(1/\delta)$ *we have that*

$$2\max_k \mathbf{x}_{*,k}\mathbf{T}\widehat{\boldsymbol{\Sigma}}_{t,i}\mathbf{x}_{*,k} + \frac{\sigma^2}{2} \leqslant \max_k \frac{1}{m}\sum_{j=1}^{m}\left(\widetilde{Y}^{(j,1)}_{t,i,k} - \widetilde{Y}^{(j,2)}_{t,i,k}\right)^2 \leqslant 5\max_k \mathbf{x}_{*,k}\mathbf{T}\widehat{\boldsymbol{\Sigma}}_{t,i}\mathbf{x}_{*,k} + \frac{5\sigma^2}{2}.$$

These claims hold with probability at least $1 - \delta$*.*

The claim is proved in section G.1. The key idea in the proof is that (8.2) multiplied by $m/[2(\mathbf{x}_{*,k}\mathbf{T}\widehat{\boldsymbol{\Sigma}}_{t,i}\mathbf{x}_{*,k} + \sigma^2)]$ is a chi-squared random variable with m degrees of freedom. Then we use concentration inequalities of Laurent and Massart (2000) to get a high-probability confidence interval on distance to the mean m, which in turn allows us to relate the actual variance to its empirical estimate. theorem 8.2 shows that SAL is equivalent to GO for a sufficiently large sample size m. theorem 8.1 can be then adapted to SAL as follows. The only change is in condition on T, which changes to $T \leqslant \alpha^2 \Big/ \left(\beta + \sqrt{2}O\left(\frac{1-\sqrt{1/m}}{1+\sqrt{1/m}}\right)\right)\beta d + O(\sqrt{1/m})$. Therefore, SAL attains a near-optimal $O(1/T)$ decrease in posterior variance as $m \to \infty$.

8.5 Experiments

We evaluate GO and SAL on a variety of prediction tasks. These tasks cover both classification and regression, including natural language features, and help us to evaluate the capabilities of GO and SAL to choose few-shot examples for active in-context prompt design. We also demonstrate that GO and SAL can be used for general pattern recognition. Detailed descriptions of all datasets are in Section G.2. We describe the prompts in detail in Section G.3.

Experimental Setup

We use Mistral-7B (Jiang et al., 2023), Vicuna-13B (Chiang et al., 2023), and Falcon-40B (Penedo et al., 2023) as the LLMs and design prompts following Dinh et al. (2022) and Suzgun et al. (2022). To investigate the impact of LLM model size on performance, we experiment with these three models of varying sizes: 7B, 13B, and 40B. Interestingly, we observe that the smaller models (Mistral-7B and Vicuna-13B) perform very poorly on certain tasks. Examples of the prompts are given in Section G.3. Each experiment is averaged over 10 trials. At the beginning of each trial, we randomly select $K = 20$ test examples. We describe in detail how the training set and n are chosen for each dataset in Section G.2.

Each run is a simulation that proceeds as follows. In round t, each method selects a training example to label X_t and then observes the true label Y_t. All past interactions $H_t = \{(X_\ell, Y_\ell)\}_{\ell \in [t-1]}$ along with the test examples $\mathbf{x}_{*,k}$ are used to craft a prompt for the LLM. The performance at round t is evaluated by the *error* $\mathcal{L}(t) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}(Y_{*,k}, \tilde{Y}_{*,k,t})$, where $Y_{*,k}$ is the true label of test example $\mathbf{x}_{*,k}$, $\tilde{Y}_{*,k,t}$ is its LLM predicted label in round t, and $\mathcal{L}(y_*, y)$ is a task-specific error function. For classification tasks, we choose $\mathcal{L}(y_*, y) = \mathbb{I}\{y_* = y\}$ and call $\mathcal{L}(t)$ a *misclassification error*. For regression tasks, we choose $\mathcal{L}(y_*, y) = (y_* - y)^2$ and call $\mathcal{L}(t)$ the *MSE*.

For pattern recognition tasks, where $Y_{*,k}$ and $\tilde{Y}_{*,k,t}$ are either vectors or matrices, we choose let $\mathcal{L}(y_*, y) = \mathbb{I}\{y_* = y\}$ and $\mathcal{L}(t)$ represents *0-1 error*.

We posit that GO and SAL perform well because they both reduce the uncertainty of test examples based on the right notion of similarity. To show this, we compare to baselines that reduce uncertainty uniformly (like Uniform), or reduce uncertainty informatively (Least or Max-Entropy), or only select examples with similar features to test examples (Greedy-NN). As shown in our extensive experiments, these baselines fail to match the capabilities of GO and SAL to select informative examples in the majority of the tasks. The following methods are compared in our experiments:

(1) Uniform: The example X_t in round t is sampled uniformly at random from the unlabeled set $\mathcal{U}_t$. Uniform is a pure exploration algorithm that does not take into account the similarity to test examples and variance reduction. We chose it as a baseline because it tends to work well in practice. Therefore, it is used frequently in active learning and prompt tuning papers (Zhang et al., 2022b; Diao et al., 2023).

(2) Greedy-NN: The example X_t in round t is chosen to align the most with all test examples $\mathbf{x}_{*,k}$ such that $I_t \leftarrow \arg\max_{i \in \mathcal{U}_t} \max_{k \in [K]} \mathbf{x}_{*,k} \mathbf{T} \mathbf{x}_i$. This baseline shows that our information gathering rule in (8.1) goes beyond pure feature similarity. This baseline is similar to the automatic exemplar construction method by clustering by Zhang et al. (2022b).

(3) Least: This is similar to the disagreement-based method of Diao et al. (2023). The disagreement score of the example $i \in \mathcal{U}_t$ is calculated as $s_i = \sum_{k=1}^{K} Y_{tik}$ where $Y_{tik} \sim p(\cdot \mid \mathbf{x}_{*,k}, \mathbf{x}_i)$ is the number of unique answers by for test example $\mathbf{x}_{*,k}$ using only $\mathbf{x}_i$ as the in-context example by the LLM. Then the example selected at round t is $I_t \leftarrow \arg\max_{i \in \mathcal{U}_t} s_i$. This is the unlabeled example where the LLM, disagrees the most for all test examples and is least confident. We compare against this baseline to show that our information gathering rule in (8.1) goes beyond just uncertainty sampling but also takes into account the diversity of training examples

when choosing to label the next example.

(4) Max-Entropy: This is the uncertainty-based maximum entropy method of Zhang et al. (2022a); Diao et al. (2023). At round t the example with the highest entropy is selected as $I_t \leftarrow \arg\max_{i \in \mathcal{U}_t} -\sum_{k=1}^{K} \bar{Y}_{tik} \ln \bar{Y}_{tik}$ where $\bar{Y}_{tik} \sim p(\cdot \mid \mathbf{x}_{*,k}, \mathbf{x}_i)$ is the frequency of a predicted answer among all predictions for the test example $\mathbf{x}_{*,k}$ using $\mathbf{x}_i$ as the in-context example by the LLM. A larger entropy denotes greater uncertainty and therefore, an unlabeled example with the largest entropy will be selected. Again we compare against this uncertainty-based baseline to show that our information gathering rule in (8.1) goes beyond just uncertainty sampling but also considers the diversity of training examples when choosing to label the next example.

(5) GO (ours): This is algorithm 10 where $\mathcal{X}$ are the original feature vectors.

(6) GO-Inst (ours): This is algorithm 10 where the original feature vectors are used in the prompt but $\mathcal{X}$ are their 768-dimensional Instructor embeddings (Su et al., 2022). We use this for natural language classification tasks.

(7) SAL (ours): This is algorithm 11 where $\mathcal{X}$ are the original feature vectors. To implement SAL efficiently, we combine it with GO as a preprocessing step. Specifically, in round t, GO first chooses 5 most informative examples from $\mathcal{U}_t$ and then we apply SAL. We use $m = 1$ in all experiments. We use these approximations because SAL is computationally expensive (section 8.3). Similarly to GO-Inst, we use Instructor embeddings for natural language classification tasks.

All used datasets and experimental setups are described in Section G.2. This section only summarizes the main results.

Standard classification and regression tasks. We use 4 classification and 2 regression datasets from UCI and OpenML (Section G.2). We set $T = 5$ to simulate the realistic scenario when the test queries provided

	Datasets	Uniform	Greedy-NN	Least	Max-Entropy	GO (ours)	SAL (ours)
M	iris	0.41 ± 0.11	0.60 ± 0.13	0.64 ± 0.15	0.72 ± 0.17	0.38 ± 0.14	$\mathbf{0.34 \pm 0.14}$
	banknote	0.75 ± 0.10	$\mathbf{0.58 \pm 0.04}$	0.59 ± 0.02	0.73 ± 0.16	0.77 ± 0.07	0.75 ± 0.15
	balance-scale	0.61 ± 0.13	0.69 ± 0.22	0.55 ± 0.25	0.57 ± 0.14	$\mathbf{0.48 \pm 0.09}$	0.72 ± 0.04
	thyroid-new	$\mathbf{0.44 \pm 0.12}$	0.70 ± 0.08	0.74 ± 0.12	0.57 ± 0.08	0.55 ± 0.06	0.63 ± 0.14
V	iris	0.22 ± 0.24	0.60 ± 0.37	0.60 ± 0.49	0.40 ± 0.20	0.20 ± 0.24	$\mathbf{0.20 \pm 0.24}$
	banknote	0.40 ± 0.37	0.80 ± 0.24	0.50 ± 0.32	0.50 ± 0.32	0.50 ± 0.32	$\mathbf{0.10 \pm 0.20}$
	balance-scale	0.60 ± 0.20	0.60 ± 0.37	0.50 ± 0.32	0.80 ± 0.24	$\mathbf{0.30 \pm 0.24}$	0.50 ± 0.00
	thyroid-new	0.52 ± 0.45	1.00 ± 0.00	0.70 ± 0.24	0.50 ± 0.00	0.60 ± 0.20	$\mathbf{0.50 \pm 0.32}$
F	iris	$\mathbf{0.20 \pm 0.06}$	0.62 ± 0.14	0.70 ± 0.20	0.65 ± 0.18	0.42 ± 0.10	0.33 ± 0.23
	banknote	0.45 ± 0.23	0.53 ± 0.25	0.60 ± 0.12	0.42 ± 0.17	0.45 ± 0.06	$\mathbf{0.45 \pm 0.1}$
	balance-scale	0.70 ± 0.28	0.68 ± 0.13	0.85 ± 0.12	0.62 ± 0.08	0.47 ± 0.24	$\mathbf{0.45 \pm 0.13}$
	thyroid-new	0.55 ± 0.29	0.57 ± 0.20	0.75 ± 0.19	0.65 ± 0.15	0.55 ± 0.23	$\mathbf{0.53 \pm 0.12}$

Table 8.1: Misclassification error in classification datasets using Mistral-7B (M), Vicuna-13B (V), and Falcon-40B (F) on $K = 20$ test examples at the end of budget $T = 5$.

	Datasets	Uniform	Greedy-NN	Least	Max-Entropy	GO (ours)	SAL (ours)
M	machine(e+04)	11.4 ± 3.34	10.5 ± 2.44	14.3 ± 3.39	11.0 ± 1.93	$\mathbf{10.5 \pm 3.74}$	10.6 ± 3.84
	fifa(e-04)	$1.40 \pm .216$	3.72 ± 1.12	$1.18 \pm .53$	4.15 ± 1.11	$.999 \pm .404$	$\mathbf{.68 \pm .26}$
V	machine(e+04)	5.59 ± 1.35	$5.04 \pm .851$	7.95 ± 1.69	4.98 ± 1.06	5.66 ± 1.54	$\mathbf{4.80 \pm 1.46}$
	fifa(e+03)	5.90 ± 1.59	$4.72 \pm .931$	5.11 ± 1.12	6.76 ± 1.69	$\mathbf{1.44 \pm .258}$	$2.59 \pm .742$
F	machine(e+03)	1.16 ± 1.22	4.28 ± 2.30	2.15 ± 1.08	$3.50 \pm 2.45e+03$	$.32 \pm .209$	$\mathbf{2.96 \pm 1.56}$
	fifa(e+01)	7.78 ± 3.85	6.95 ± 2.93	12.4 ± 12.3	26.3 ± 37.6	7.90 ± 4.64	$\mathbf{4.61 \pm 4.35}$

Table 8.2: MSE in regression datasets using Mistral-7B (M), Vicuna-13B (V), and Falcon-40B (F) on $K = 20$ test examples at the end of budget $T = 5$.

Task	Uniform	Greedy-NN	Least	Max-Entropy	GO (ours)	SAL (ours)
Arc-1	0.45 ± 0.50	0.45 ± 0.50	0.90 ± 0.30	0.60 ± 0.49	0.30 ± 0.46	$\mathbf{0.15 \pm 0.36}$
Arc-2	0.80 ± 0.40	1.00 ± 0.00	0.80 ± 0.40	0.80 ± 0.40	0.80 ± 0.40	$\mathbf{0.01 \pm 0.01}$
PCFG-1	0.60 ± 0.49	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.01	0.20 ± 0.40	$\mathbf{0.02 \pm 0.01}$
PCFG-2	0.20 ± 0.40	1.00 ± 0.00	0.20 ± 0.40	1.00 ± 0.00	0.20 ± 0.40	$\mathbf{0.14 \pm 0.40}$

Table 8.3: 0-1 error using Falcon-40B on $K = 20$ test examples at the end of budget $T = 5$. ARC-1 is the expansion-contraction task, ARC-2 is the rotation task, PCFG-1 is the add-subtract task, and PCFG-2 is the repeat experiment task. Mistral-7B and Vicuna-13B perform very poorly on these tasks and thus are omitted.

by the user need to be inferred quickly. For classification tasks, $K = 20$ test examples are chosen among the different classes of the dataset. We describe in detail how the training set and n are chosen for each dataset in Section G.2. For regression tasks, $K = 20$ random test examples are chosen. Our results on classification tasks are reported in table 8.1 and on regression tasks in table 8.2. We observe that GO and SAL are the

	Datasets	Uniform	Greedy-NN	Least	Max-Entropy	GO (ours)	SAL (ours)
M	movie	0.32 ± 0.17	0.90 ± 0.06	0.87 ± 0.09	0.55 ± 0.18	$\mathbf{0.27 \pm 0.10}$	0.49 ± 0.11
	entity	0.69 ± 0.19	0.86 ± 0.06	0.87 ± 0.09	0.59 ± 0.15	0.65 ± 0.18	$\mathbf{0.39 \pm 0.19}$
	theme	0.74 ± 0.05	0.74 ± 0.09	0.82 ± 0.16	$\mathbf{0.68 \pm 0.09}$	0.80 ± 0.09	0.81 ± 0.08
V	movie	0.10 ± 0.20	0.70 ± 0.24	0.90 ± 0.20	0.30 ± 0.24	$\mathbf{0.02 \pm 0.01}$	0.10 ± 0.20
	entity	0.20 ± 0.24	0.90 ± 0.20	0.70 ± 0.24	0.60 ± 0.20	$\mathbf{0.10 \pm 0.20}$	0.10 ± 0.20
	theme	0.90 ± 0.20	0.70 ± 0.40	1.00 ± 0.00	0.70 ± 0.24	$\mathbf{0.60 \pm 0.37}$	0.80 ± 0.40
F	movie	0.55 ± 0.06	0.62 ± 0.18	0.78 ± 0.17	0.53 ± 0.22	$\mathbf{0.38 \pm 0.18}$	0.47 ± 0.17
	entity	0.55 ± 0.23	0.62 ± 0.08	0.75 ± 0.14	0.65 ± 0.24	0.47 ± 0.23	$\mathbf{0.42 \pm 0.24}$
	theme	0.68 ± 0.20	0.70 ± 0.13	0.85 ± 0.05	0.85 ± 0.09	$\mathbf{0.53 \pm 0.18}$	0.55 ± 0.19

Table 8.4: Misclassification error in natural language classification tasks using Mistral-7B (M), Vicuna-13B (V), and Falcon-40B (F) on $K = 20$ test examples at the end of budget $T = 5$.

best-performing methods in the majority of the datasets. Note that there is no single baseline that consistently outperforms them.

General pattern recognition. We experiment with 4 tasks: ARC expansion and contraction, ARC rotation, PCFG Add-Subtract, and PCFG Repeat. Both inputs and outputs in these tasks are vectors or matrices. We describe examples of ARC and PCFG tasks in detail in Section G.2. Each dataset comprises examples of two patterns: expansion and contraction, clockwise and counter-clockwise rotation, add and subtract, repeat first and second digits. In each trial, we choose $K = 20$ different test examples equally from two patterns and set $T = 5$. Our results are reported in table 8.3. In all tasks, GO and SAL are the best-performing methods. SAL outperforms GO in ARC (Alford, 2021; Mirchandani et al., 2023) and PCFG (Hupkes et al., 2020) consistently.

Natural language classification tasks (NLC). We show that GO and SAL work well on general NLP tasks where no explicit numerical features are available. We create three synthetic datasets based on the following tasks: (i) `movie-names`: predicting a genre from a movie name (e.g., romance, horror), (ii) `movie-theme`: predicting a common theme for a pair of movie names (e.g., coming-of-age, sci-fi), and (iii) `entity-names`: predicting an entity's type from its name (e.g., celebrity, mountain, river). Each dataset comprises 5 classes. Further details regarding the additional

datasets are provided in Section G.2. In each trial, $K = 20$ test examples were randomly chosen across the five classes. The feature vectors in GO and SAL are Instructor embeddings of the original text features. Our results are reported in table 8.4. We observe again that GO and SAL are the best-performing methods in the majority of the datasets. There is no single baseline that consistently outperforms them. This shows that the optimal design-based approach of GO and SAL correctly balances uncertainty and diversity-based sampling.

8.6 Conclusions

In this paper, we studied the framework of active in-context prompt design (AIPD) that uses optimal design to systematically choose the most informative unlabeled examples to label for a set of test examples. These informative labeled examples are then used to minimize the prediction error of the LLM for all the test examples. To our knowledge, this is the first paper that studies optimal design for adaptive prompt design. Inspired by the linear model, we proposed an algorithm GO that strategically chooses the most informative examples that minimize the variance of the posterior covariance for any test example from the test set. We proposed a second algorithm SAL that uses simulations to estimate the impact of how unlabeled examples reduce LLMuncertainty for all test examples. It then chooses to label examples that maximally reduce the uncertainty of the LLM for all test examples from the test set. We theoretically analyze GO and SAL and show their equivalence in linear models. We show that both algorithms guarantee information gain at each iteration. Finally, we show empirically that, when used with LLMs like Mistral-7B, Vicuna-13B, and Falcon-40B, both GO and SAL result in better prediction accuracy than other baselines (Zhang et al., 2022a,b; Diao et al., 2023) on tasks like classification, regression, ARC, PCFG, and natural language generation. Our research opens up exciting new directions for future work such as

extending AIPD framework beyond text to enable informative example selection for tasks involving images, videos, or other modalities using multi-modal LLMs (Yin et al., 2023). Additionally, the integration of active learning with diffusion models, a powerful class of generative models, presents promising directions for future research (Ho et al., 2020).

Part V

Conclusion

9 CONCLUSION

In this thesis, we studied how to adaptively collect data for policy evaluation, multi-task learning, and learning preference models for LLM alignment. The main question we addressed in this thesis is:

> *How to adaptively collect diverse and informative data to balance exploration-exploitation and minimize the metric of error?*

To address this main question we divided the thesis into three parts where each part has one central theme as follows: **1**) Adaptive data collection for policy evaluation which leads to a better evaluation of a learning agent before its deployment. In this part, the metric of error is the mean squared error of the value of the target policy. We showed in the three subsequent chapters how to use optimal design to minimize the mean squared error of the value of the target policy compared to an oracle that has access to the problem parameters. **2**) Adaptive data collection for Multi-task learning which helps the learning agent to minimize the metric of error across the tasks by leveraging the shared structure across the tasks. In this part, the metric of error considered is the prediction error of the best arm for each task and the cumulative regret across the tasks. It also leads the agent to generalize well to new unseen tasks given that this task shares some similarities with the tasks during training time. This section consists of two chapters where the first chapter shows how to adaptively collect data that minimizes prediction error of the best arm for each task. The last chapter in this part shows how to minimize the cumulative regret using a Decision Transformer from data collected by a weak demonstrator. **3**) Finally we show how to adaptively select informative examples to learn preference models for aligning LLMs and adaptively designing prompts for few-shot learning using LLMs. In this last part, the core theme is data

collection for LLMs and consists of two more chapters. In the first chapter, we show how to use optimal design to learn the preference model using human feedback. In the next chapter, we show how to use optimal design to collect and build informative prompts for few-shot learning in LLMs. At the end of each of the chapters, we also talk about how to extend these works for future directions.

REFERENCES

Abbasi-Yadkori, Yasin, Dávid Pál, and Csaba Szepesvári. 2011. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems* 24.

Agarwal, Alekh, Nan Jiang, Sham M Kakade, and Wen Sun. 2019. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*.

Agarwal, Ananye, Ashish Kumar, Jitendra Malik, and Deepak Pathak. 2022. Legged locomotion in challenging terrains using egocentric vision. *CoRL*.

Agarwal, Deepak, Bee-Chung Chen, and Pradheep Elango. 2009. Explore/exploit schemes for web content optimization. In *2009 ninth ieee international conference on data mining*, 1–10. IEEE.

Agrawal, Shipra, and Navin Goyal. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*, 39–1. JMLR Workshop and Conference Proceedings.

Akrour, Riad, Marc Schoenauer, and Michèle Sebag. 2012. April: Active preference learning-based reinforcement learning. In *Machine learning and knowledge discovery in databases: European conference, ecml pkdd 2012, bristol, uk, september 24-28, 2012. proceedings, part ii 23*, 116–131. Springer.

Alford, Simon. 2021. A neurosymbolic approach to abstraction and reasoning. Ph.D. thesis, Massachusetts Institute of Technology.

Altman, Eitan. 2021. *Constrained markov decision processes*. Routledge.

Amani, Sanae, Mahnoosh Alizadeh, and Christos Thrampoulidis. 2019. Linear stochastic bandits under safety constraints. In *Advances in neural*

information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada, ed. Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, 9252–9262.

Amodei, Dario, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*.

Antos, András, Varun Grover, and Csaba Szepesvári. 2008. Active learning in multi-armed bandits. In *International conference on algorithmic learning theory*, 287–302. Springer.

Arthur, David, and Sergei Vassilvitskii. 2006. k-means++: The advantages of careful seeding. Tech. Rep., Stanford.

Asghar, Nabiha. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Ash, Jordan, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. 2021. Gone fishing: Neural active learning with fisher embeddings. *Advances in Neural Information Processing Systems* 34.

Ash, Jordan T, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*.

Audibert, Jean-Yves, Sébastien Bubeck, and Rémi Munos. 2010. Best arm identification in multi-armed bandits. *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010* 41–53.

Audibert, Jean-Yves, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410(19):1876–1902.

Auer, Peter, Nicolò Cesa-Bianchi, and Paul Fischer. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning* 47(2): 235–256.

Auer, Peter, and Ronald Ortner. 2010. Ucb revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica* 61(1-2):55–65.

Azizi, Mohammad Javad, Branislav Kveton, and Mohammad Ghavamzadeh. 2022. Fixed-budget best-arm identification in structured bandits. In *Proceedings of the 31st international joint conference on artificial intelligence*.

Baek, Jackie, and Vivek Farias. 2023. Ts-ucb: Improving on thompson sampling with little to no additional computation. In *International conference on artificial intelligence and statistics*, 11132–11148. PMLR.

Bai, Yu, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. 2023. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*.

Bai, Yuntao, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Balcan, Maria-Florina, Alina Beygelzimer, and John Langford. 2009. Agnostic active learning. *Journal of Computer and System Sciences* 75(1):78–89.

Barto, Andrew G. 2013. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, 17–47. Springer.

Bengio, Yoshua, Samy Bengio, and Jocelyn Cloutier. 1990. *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche ....

Bengio, Yoshua, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.

Bengs, Viktor, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. 2021. Preference-based online learning with dueling bandits: A survey. *The Journal of Machine Learning Research* 22(1):278–385.

Berthet, Quentin, and Vianney Perchet. 2017. Fast rates for bandit optimization with upper-confidence frank-wolfe. *Advances in Neural Information Processing Systems* 30.

Bhatia, Rajendra. 2013. *Matrix analysis*, vol. 169. Springer Science & Business Media.

Bhatt, Gantavya, Yifang Chen, Arnav M Das, Jifan Zhang, Sang T Truong, Stephen Mussmann, Yinglun Zhu, Jeffrey Bilmes, Simon S Du, Kevin Jamieson, et al. 2024. An experimental design framework for label-efficient supervised finetuning of large language models. *arXiv preprint arXiv:2401.06692*.

Bishop, C. 2006a. Pattern recognition and machine learning. *Springer google schola* 2:531–537.

Bishop, Christopher. 2006b. *Pattern recognition and machine learning*. New York, NY: Springer.

Bıyık, Erdem, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. 2020. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*.

Bottou, Léon, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research* 14(11).

Bouchard, Guillaume, Théo Trouillon, Julien Perez, and Adrien Gaidon. 2016. Online learning to sample. *arXiv preprint arXiv:1506.09016*.

Boucheron, Stephane, Gabor Lugosi, and Pascal Massart. 2013. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press.

Bousquet, Olivier, and André Elisseeff. 2002. Stability and generalization. *The Journal of Machine Learning Research* 2:499–526.

Bowman, Benjamin, Alessandro Achille, Luca Zancato, Matthew Trager, Pramuditha Perera, Giovanni Paolini, and Stefano Soatto. 2023. a-la-carte prompt tuning (apt): Combining distinct data via composable prompting. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 14984–14993.

Box, George EP, and George C Tiao. 2011. *Bayesian inference in statistical analysis*. John Wiley & Sons.

Bradley, Ralph Allan, and Milton Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika* 39(3-4): 324–345.

Bragman, Felix JS, Ryutaro Tanno, Zach Eaton-Rosen, Wenqi Li, David J Hawkes, Sebastien Ourselin, Daniel C Alexander, Jamie R McClelland, and M Jorge Cardoso. 2018. Uncertainty in multitask learning: joint representations for probabilistic mr-only radiotherapy planning. In *Medical image computing and computer assisted intervention–miccai 2018: 21st*

international conference, granada, spain, september 16-20, 2018, proceedings, part iv 11, 3–11. Springer.

Brandfonbrener, David, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. 2022. When does return-conditioned supervised learning work for offline reinforcement learning? *Advances in Neural Information Processing Systems* 35:1542–1553.

Brohan, Anthony, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33:1877–1901.

Bubeck, Sébastien, Nicolo Cesa-Bianchi, et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning* 5(1):1–122.

Bubeck, Sebastien, Remi Munos, and Gilles Stoltz. 2009. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th international conference on algorithmic learning theory*, 23–37.

Bubeck, Sébastien, Gilles Stoltz, Csaba Szepesvári, and Rémi Munos. 2008. Online optimization in x-armed bandits. *Advances in Neural Information Processing Systems* 21.

Bubeck, Sébastien, Gilles Stoltz, and Jia Yuan Yu. 2011. Lipschitz bandits without the lipschitz constant. In *Algorithmic learning theory: 22nd international conference, alt 2011, espoo, finland, october 5-7, 2011. proceedings 22*, 144–158. Springer.

BuoyHealth. BuoyHealth. https://www.buoyhealth.com/.

Burda, Yuri, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*.

Burges, Christopher. 2010. From RankNet to LambdaRank to LambdaMART: An overview. Tech. Rep. MSR-TR-2010-82, Microsoft Research.

Cai, Hengrui, Chengchun Shi, Rui Song, and Wenbin Lu. 2021. Deep jump learning for off-policy evaluation in continuous treatment settings. *Advances in Neural Information Processing Systems* 34:15285–15300.

Camilleri, Romain, Andrew Wagenmaker, Jamie H Morgenstern, Lalit Jain, and Kevin G Jamieson. 2022. Active learning with safety constraints. *Advances in Neural Information Processing Systems* 35:33201–33214.

Carlin, Bradley P, and Thomas A Louis. 2008. *Bayesian methods for data analysis*. CRC press.

Carpentier, Alexandra, and Rémi Munos. 2011. Finite-time analysis of stratified sampling for monte carlo. In *Nips-twenty-fifth annual conference on neural information processing systems*.

———. 2012. Minimax number of strata for online stratified sampling given noisy samples. In *International conference on algorithmic learning theory*, 229–244. Springer.

Carpentier, Alexandra, Remi Munos, and András Antos. 2015. Adaptive strategy for stratified monte carlo sampling. *J. Mach. Learn. Res.* 16:2231–2271.

Caruccio, Loredana, Stefano Cirillo, Giuseppe Polese, Giandomenico Solimando, Shanmugam Sundaramurthy, and Genoveffa Tortora. 2024. Can chatgpt provide intelligent diagnoses? a comparative study between

predictive models and chatgpt to define a new medical diagnostic bot. *Expert Systems with Applications* 235:121186.

Casper, Stephen, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Chaudhuri, Kamalika, Prateek Jain, and Nagarajan Natarajan. 2017. Active heteroscedastic regression. In *International conference on machine learning*, 694–702. PMLR.

Chaudhuri, Kamalika, Sham M Kakade, Praneeth Netrapalli, and Sujay Sanghavi. 2015. Convergence rates of active learning for maximum likelihood estimation. In *Advances in neural information processing systems*, 1090–1098.

Chen, Fan, Junyu Zhang, and Zaiwen Wen. 2022a. A near-optimal primal-dual method for off-policy learning in cmdp. *Advances in Neural Information Processing Systems* 35:10521–10532.

Chen, Lili, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021a. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34:15084–15097.

Chen, Xiaoyu, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. 2022b. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International conference on machine learning*, 3773–3793. PMLR.

Chen, Yi, Jing Dong, and Zhaoran Wang. 2021b. A primal-dual approach to constrained markov decision processes. *arXiv preprint arXiv:2101.10895*.

Chen, Zhuotong, Yifei Ma, Branislav Kveton, and Anoop Deoras. 2023. Active learning with crowd sourcing improves information retrieval. In *Icml 2023 workshop on interactive learning with implicit human feedback*.

Chiang, Wei-Lin, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Chowdhury, Sayak Ray, and Aditya Gopalan. 2017. On kernelized multi-armed bandits. In *International conference on machine learning*, 844–853. PMLR.

Chowdhury, Sayak Ray, Aditya Gopalan, and Odalric-Ambrym Maillard. 2021. Reinforcement learning in parametric mdps with exponential families. In *International conference on artificial intelligence and statistics*, 1855–1863. PMLR.

Christiano, Paul F, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30.

Chu, Wei, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 208–214. JMLR Workshop and Conference Proceedings.

Chuklin, Aleksandr, Ilya Markov, and Maarten De Rijke. 2022. *Click models for web search*. Springer Nature.

Ciosek, Kamil, and Shimon Whiteson. 2017. OFFER: Off-environment reinforcement learning. In *Proceedings of the 31st aaai conference on artificial intelligence (aaai)*.

Citovsky, Gui, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale. *Advances in Neural Information Processing Systems* 34.

Corrado, Nicholas, and Josiah P. Hanna. 2023. On-policy policy gradient reinforcement learning without on-policy sampling. In *Arxiv pre-print*.

Cortez, Paulo, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems* 47(4):547–553.

Crawshaw, Michael. 2020. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.

Dai, Zhongxiang, Yao Shu, Arun Verma, Flint Xiaofeng Fan, Bryan Kian Hsiang Low, and Patrick Jaillet. 2022. Federated neural bandit. *arXiv preprint arXiv:2205.14309*.

Dann, Christoph, Lihong Li, Wei Wei, and Emma Brunskill. 2019. Policy certificates: Towards accountable reinforcement learning. In *International conference on machine learning*, 1507–1516. PMLR.

Das, Nirjhar, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. 2024. Active preference optimization for sample efficient RLHF. *CoRR* abs/2402.10500.

Dasgupta, Sanjoy. 2005. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems 17*, 337–344.

Dasgupta, Sanjoy, Daniel J Hsu, and Claire Monteleoni. 2007. A general agnostic active learning algorithm. *Advances in neural information processing systems* 20.

De Vito, Saverio, Ettore Massera, Marco Piga, Luca Martinotto, and Girolamo Di Francia. 2008. On field calibration of an electronic nose for

benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical* 129(2):750–757.

Degenne, Rémy, Pierre Ménard, Xuedong Shang, and Michal Valko. 2020. Gamification of pure exploration for linear bandits. In *International conference on machine learning*, 2432–2442. PMLR.

Deng, Zhijie, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of llm-generated texts with a bayesian surrogate model. *arXiv preprint arXiv:2305.16617*.

Deshpande, Yash, and Andrea Montanari. 2012. Linear bandits in high dimension and recommendation systems. In *2012 50th annual allerton conference on communication, control, and computing (allerton)*, 1750–1754. IEEE.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Diamond, Steven, and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17(83):1–5.

Diao, Shizhe, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.

Ding, Dongsheng, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. 2021. Provably efficient safe exploration via primal-dual policy optimization. In *International conference on artificial intelligence and statistics*, 3304–3312. PMLR.

Ding, Dongsheng, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. 2020. Natural policy gradient primal-dual method for constrained markov

decision processes. *Advances in Neural Information Processing Systems* 33: 8378–8390.

Ding, Ning, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence* 5(3):220–235.

Ding, Shutong, Jingya Wang, Yali Du, and Ye Shi. 2024. Reduced policy optimization for continuous control with hard constraints. *Advances in Neural Information Processing Systems* 36.

Dinh, Tuan, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. 2022. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems* 35: 11763–11784.

Dodge, Jesse, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Dong, Kefan, Jiaqi Yang, and Tengyu Ma. 2021. Provable model-based nonlinear bandit and reinforcement learning: Shelve optimism, embrace virtual curvature. *Advances in neural information processing systems* 34: 26168–26182.

Dong, Qingxiu, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Du, Simon S, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. 2020. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*.

Du, Yihan, Longbo Huang, and Wen Sun. 2023. Multi-task representation learning for pure exploration in linear bandits. *arXiv preprint arXiv:2302.04441*.

Duan, Yan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. 2016. Rl
2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Dudík, Miroslav, Dumitru Erhan, John Langford, and Lihong Li. 2014. Doubly robust policy evaluation and optimization.

Dukler, Yonatan, Alessandro Achille, Giovanni Paolini, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. 2021. Diva: Dataset derivative of a learning task. *arXiv preprint arXiv:2111.09785*.

Efroni, Yonathan, Shie Mannor, and Matteo Pirotta. 2020. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*.

Ermis, Beyza, Patrick Ernst, Yannik Stein, and Giovanni Zappella. 2020. Learning to rank in the position based model with bandit feedback. In *Proceedings of the 29th acm international conference on information & knowledge management*, 2405–2412.

Fang, Yuguang, Kenneth A Loparo, and Xiangbo Feng. 1994. Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control* 39(12):2489–2490.

Fedorov, Valerii. 2010. Optimal experimental design. *Wiley Interdisciplinary Reviews: Computational Statistics* 2(5):581–589.

Fedorov, Valerii Vadimovich. 2013. *Theory of optimal experiments*. Elsevier.

Fiez, Tanner, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. 2019. Sequential experimental design for transductive linear bandits. *Advances in neural information processing systems* 32.

Fifty, Chris, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. 2021. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems* 34:27503–27516.

Filippi, Sarah, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. 2010a. Parametric bandits: The generalized linear case. *Advances in neural information processing systems* 23.

Filippi, Sarah, Olivier Cappe, Aurelien Garivier, and Csaba Szepesvari. 2010b. Parametric bandits: The generalized linear case. In *Advances in neural information processing systems 23*, 586–594.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Fischer, Thomas G. 2018. Reinforcement learning in financial markets-a survey. Tech. Rep., FAU Discussion Papers in Economics.

Fontaine, Xavier, Pierre Perrault, Michal Valko, and Vianney Perchet. 2021. Online a-optimal design and active linear regression. In *International conference on machine learning*, 3374–3383. PMLR.

François-Lavet, Vincent, Peter Henderson, Riashat Islam, Marc G Bellemare, Joelle Pineau, et al. 2018. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning* 11(3-4):219–354.

Friedman, Dan, Alexander Wettig, and Danqi Chen. 2024. Learning transformer programs. *Advances in Neural Information Processing Systems* 36.

Fu, Justin, Sergey Levine, and Pieter Abbeel. 2016. One-shot learning of manipulation skills with online dynamics adaptation and neural network priors. In *2016 ieee/rsj international conference on intelligent robots and systems (iros)*, 4019–4026. IEEE.

Fujimoto, Scott, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, 2052–2062. PMLR.

Fürnkranz, Johannes, and Eyke Hüllermeier. 2003. Pairwise preference learning and ranking. In *European conference on machine learning*, 145–156. Springer.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *Proceedings of the 34th international conference on machine learning*, ed. Doina Precup and Yee Whye Teh, vol. 70 of *Proceedings of Machine Learning Research*, 1183–1192. PMLR.

Gao, Tianshi, and Daphne Koller. 2011. Active classification based on value of classifier. *Advances in neural information processing systems* 24.

Garcelon, Evrard, Mohammad Ghavamzadeh, Alessandro Lazaric, and Matteo Pirotta. 2020. Improved algorithms for conservative exploration in bandits. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, new york, ny, usa, february 7-12, 2020*, 3962–3969. AAAI Press.

Garivier, Aurélien, and Emilie Kaufmann. 2016. Optimal best arm identification with fixed confidence. In *Conference on learning theory*, 998–1027. PMLR.

Ge, Yao, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker. 2022. Few-shot learning for medical text: A systematic review. *arXiv preprint arXiv:2204.14081*.

Geifman, Yonatan, and Ran El-Yaniv. 2017. Deep active learning over the long tail. *arXiv preprint arXiv:1711.00941*.

Ghasemipour, Kamyar, Shixiang Shane Gu, and Ofir Nachum. 2022. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems* 35:18267–18281.

Glass, Alyssa. 2006. Explaining preference learning.

Greene, William H. 2002. 000. econometric analysis.

Gu, Shangding, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, Yaodong Yang, and Alois Knoll. 2022. A review of safe reinforcement learning: Methods, theory and applications. *arXiv preprint arXiv:2205.10330*.

Gupta, Abhishek, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. 2018. Meta-reinforcement learning of structured exploration strategies. *Advances in neural information processing systems* 31.

Gupta, Samarth, Shreyas Chaudhari, Subhojyoti Mukherjee, Gauri Joshi, and Osman Yağan. 2020a. A unified approach to translate classical bandit algorithms to the structured bandit setting. *IEEE Journal on Selected Areas in Information Theory* 1(3):840–853.

Gupta, Samarth, Shreyas Chaudhari, Subhojyoti Mukherjee, Gauri Joshi, and Osman Yağan. 2020b. A unified approach to translate classical bandit algorithms to the structured bandit setting. *IEEE Journal on Selected Areas in Information Theory* 1(3):840–853.

———. 2021. A unified approach to translate classical bandit algorithms to structured bandits. In *Icassp 2021 - 2021 ieee international conference on acoustics, speech and signal processing (icassp)*, 3360–3364.

Gupta, Shourya, Utkarsh Suryaman, Rahul Narava, and Shashi Shekhar Jha. 2024. Model-based safe reinforcement learning using variable horizon rollouts. In *Proceedings of the 7th joint international conference on data science & management of data (11th acm ikdd cods and 29th comad)*, 100–108.

Hambly, Ben, Renyuan Xu, and Huining Yang. 2021. Recent advances in reinforcement learning in finance. *arXiv preprint arXiv:2112.04553*.

Hanna, Josiah P, Philip S Thomas, Peter Stone, and Scott Niekum. 2017a. Data-efficient policy evaluation through behavior policy search. In *International conference on machine learning*, 1394–1403. PMLR.

Hanna, Josiah P., Philip S. Thomas, Peter Stone, and Scott Niekum. 2017b. Data-Efficient Policy Evaluation Through Behavior Policy Search. *arXiv:1706.03469 [cs]*. ArXiv: 1706.03469.

Hanneke, Steve, et al. 2014. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning* 7(2-3):131–309.

Harper, F Maxwell, and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5(4):1–19.

Hassan, Jameel, Hanan Gani, Noor Hussein, Muhammad Uzair Khattak, Muzammal Naseer, Fahad Shahbaz Khan, and Salman Khan. 2023. Align

your prompts: Test-time prompting with distribution alignment for zero-shot generalization. *arXiv preprint arXiv:2311.01459*.

Hejna, Joey, and Dorsa Sadigh. 2023. Inverse preference learning: Preference-based rl without a reward function. *arXiv preprint arXiv:2305.15363*.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33: 6840–6851.

Hofmann, Katja, Shimon Whiteson, and Maarten de Rijke. 2013. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions on Information Systems* 31(4):1–43.

Hong, Joey, Branislav Kveton, Sumeet Katariya, Manzil Zaheer, and Mohammad Ghavamzadeh. 2022a. Deep hierarchy in bandits. In *International conference on machine learning*, 8833–8851. PMLR.

Hong, Joey, Branislav Kveton, Manzil Zaheer, Yinlam Chow, Amr Ahmed, and Craig Boutilier. 2020. Latent bandits revisited. *Advances in Neural Information Processing Systems* 33:13423–13433.

Hong, Joey, Branislav Kveton, Manzil Zaheer, and Mohammad Ghavamzadeh. 2022b. Hierarchical bayesian bandits. In *International conference on artificial intelligence and statistics*, 7724–7741. PMLR.

Hong, Joey, Branislav Kveton, Manzil Zaheer, Sumeet Katariya, and Mohammad Ghavamzadeh. 2023. Multi-task off-policy learning from bandit feedback. In *International conference on machine learning*, 13157–13173. PMLR.

Honorio, Jean, and Tommi Jaakkola. 2014. Tight bounds for the expected risk of linear classifiers and pac-bayes finite-sample guarantees. In *Artificial intelligence and statistics*, 384–392. PMLR.

Houlsby, Neil, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Huang, Ruitong, Mohammad M. Ajallooeian, Csaba Szepesvári, and Martin Müller. 2017. Structured best arm identification with fixed confidence. In *International conference on algorithmic learning theory, ALT 2017, 15-17 october 2017, kyoto university, kyoto, japan*, ed. Steve Hanneke and Lev Reyzin, vol. 76 of *Proceedings of Machine Learning Research*, 593–616. PMLR.

Hupkes, Dieuwke, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research* 67:757–795.

Hutchinson, Spencer, Berkay Turan, and Mahnoosh Alizadeh. 2024. Directional optimism for safe linear bandits. In *International conference on artificial intelligence and statistics*, 658–666. PMLR.

Ibarz, Julian, Jie Tan, Chelsea Finn, Mrinal Kalakrishnan, Peter Pastor, and Sergey Levine. 2021. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research* 40(4-5):698–721.

Jaggi, Martin. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, 427–435. PMLR.

Jamieson, Kevin, and Lalit Jain. 2022. Interactive machine learning.

Janner, Michael, Qiyang Li, and Sergey Levine. 2021. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems* 34:1273–1286.

Jeon, Hong Jun, Smitha Milli, and Anca Dragan. 2020. Reward-rational (implicit) choice: A unifying formalism for reward learning. In *Advances in neural information processing systems 33*.

Jiang, Albert Q, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Jiang, Nan, and Lihong Li. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International conference on machine learning*, 652–661. PMLR.

Jiang, Yiding, Evan Liu, Benjamin Eysenbach, J Zico Kolter, and Chelsea Finn. 2022. Learning options via compression. *Advances in Neural Information Processing Systems* 35:21184–21199.

Jin, Ying, Zhimei Ren, Zhuoran Yang, and Zhaoran Wang. 2022. Policy learning" without"overlap: Pessimism and generalized empirical bernstein's inequality. *arXiv preprint arXiv:2212.09900*.

Jin, Ying, Zhuoran Yang, and Zhaoran Wang. 2021. Is pessimism provably efficient for offline rl? In *International conference on machine learning*, 5084–5096. PMLR.

Johnson, Richard Arnold, Dean W Wichern, et al. 2002. Applied multivariate statistical analysis.

Jun, Kwang-Sung, Rebecca Willett, Stephen Wright, and Robert Nowak. 2019. Bilinear bandits with low-rank structure. In *International conference on machine learning*, 3163–3172. PMLR.

Justus, Daniel, John Brennan, Stephen Bonner, and Andrew Stephen McGough. 2018. Predicting the computational cost of deep learning

models. In *2018 ieee international conference on big data (big data)*, 3873–3882. IEEE.

Kallus, Nathan, Yuta Saito, and Masatoshi Uehara. 2021. Optimal off-policy evaluation from multiple logging policies. In *International conference on machine learning*, 5247–5256. PMLR.

Kang, Yachen, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. 2023. Beyond reward: Offline preference-guided policy optimization. 2305.16217.

Kang, Yue, Cho-Jui Hsieh, and Thomas Chun Man Lee. 2022. Efficient frameworks for generalized low-rank matrix bandit problems. *Advances in Neural Information Processing Systems* 35:19971–19983.

Katz-Samuels, Julian, Lalit Jain, Kevin G Jamieson, et al. 2020. An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits. *Advances in Neural Information Processing Systems* 33:10371–10382.

Katz-Samuels, Julian, Jifan Zhang, Lalit Jain, and Kevin Jamieson. 2021. Improved algorithms for agnostic pool-based active classification. In *International conference on machine learning*, 5334–5344. PMLR.

Kaufmann, Timo, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. 2024. A survey of reinforcement learning from human feedback. 2312.14925.

Kazerouni, Abbas, Mohammad Ghavamzadeh, Yasin Abbasi, and Benjamin Van Roy. 2017. Conservative contextual linear bandits. In *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, december 4-9, 2017, long beach, ca, USA*, ed. Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, 3910–3919.

Keles, Feyza Duman, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. 2023. On the computational complexity of self-attention. In *International conference on algorithmic learning theory*, 597–619. PMLR.

Kendall, Maurice George. 1948. Rank correlation methods.

Kiefer, Jack, and Jacob Wolfowitz. 1960. The equivalence of two extremum problems. *Canadian Journal of Mathematics* 12:363–366.

Kiran, B Ravi, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems* 23(6):4909–4926.

Kirsch, Andreas, Joost Van Amersfoort, and Yarin Gal. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems* 32.

Kirschner, Johannes, and Andreas Krause. 2018. Information directed sampling and bandits with heteroscedastic noise. In *Conference on learning theory*, 358–384. PMLR.

———. 2021. Bias-robust Bayesian optimization via dueling bandits. In *Proceedings of the 38th international conference on machine learning*.

Kohavi, Ron, and Roger Longbotham. 2017. Online controlled experiments and a/b testing. *Encyclopedia of machine learning and data mining* 7(8):922–929.

Kremer, Jan, Kim Steenstrup Pedersen, and Christian Igel. 2014. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4(4):313–326.

Kumar, Aviral, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems* 32.

Kumar, Aviral, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33:1179–1191.

Kung, Po-Nien, Fan Yin, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2023. Active instruction tuning: Improving cross-task generalization by training on prompt sensitive tasks. *arXiv preprint arXiv:2311.00288*.

Kuroiwa, Tomoyuki, Aida Sarcon, Takuya Ibara, Eriku Yamada, Akiko Yamamoto, Kazuya Tsukamoto, and Koji Fujita. 2023. The potential of chatgpt as a self-diagnostic tool in common orthopedic diseases: Exploratory study. *J Med Internet Res* 25:e47621.

Kveton, Branislav, Csaba Szepesvári, Anup Rao, Zheng Wen, Yasin Abbasi-Yadkori, and S Muthukrishnan. 2017. Stochastic low-rank bandits. *arXiv preprint arXiv:1712.04644*.

Kveton, Branislav, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. 2015. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd international conference on machine learning*.

Kwon, Jeongyeol, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. 2022. Tractable optimality in episodic latent mabs. *Advances in Neural Information Processing Systems* 35:23634–23645.

Lacoste-Julien, Simon, and Martin Jaggi. 2013. An affine invariant linear convergence analysis for frank-wolfe algorithms. *arXiv preprint arXiv:1312.7864*.

Lagree, Paul, Claire Vernade, and Olivier Cappe. 2016. Multiple-play bandits in the position-based model. In *Advances in neural information processing systems 29*, 1597–1605.

Lai, T. L, and Herbert Robbins. 1985. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6(1):4–22.

Lalitha, Anusha Lalitha, Kousha Kalantari, Yifei Ma, Anoop Deoras, and Branislav Kveton. 2023. Fixed-budget best-arm identification with heterogeneous reward variances. In *Uncertainty in artificial intelligence*, 1164–1173. PMLR.

Lam, Shyong, and Jon Herlocker. 2016. MovieLens Dataset. http://grouplens.org/datasets/movielens/.

Landolfi, Nicholas C, Garrett Thomas, and Tengyu Ma. 2019. A model-based approach for sample-efficient multi-task reinforcement learning. *arXiv preprint arXiv:1907.04964*.

Laskin, Michael, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Hansen, Angelos Filos, Ethan Brooks, et al. 2022. In-context reinforcement learning with algorithm distillation. *arXiv preprint arXiv:2210.14215*.

Lattimore, Tor, Branislav Kveton, Shuai Li, and Csaba Szepesvari. 2018. TopRank: A practical algorithm for online stochastic ranking. In *Advances in neural information processing systems 31*, 3949–3958.

Lattimore, Tor, and Csaba Szepesvári. 2019. An information-theoretic approach to minimax regret in partial monitoring. In *Conference on learning theory*, 2111–2139. PMLR.

———. 2020a. *Bandit algorithms*. Cambridge University Press.

———. 2020b. *Bandit algorithms*. Cambridge University Press.

Laurent, Beatrice, and Pascal Massart. 2000. Adaptive estimation of a quadratic functional by model selection. *Annals of statistics* 1302–1338.

Lee, Jonathan N, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. 2023. Supervised pretraining can learn in-context reinforcement learning. *arXiv preprint arXiv:2306.14892*.

Lee, Kimin, Laura Smith, Anca Dragan, and Pieter Abbeel. 2021. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*.

Lee, Kuang-Huei, Ofir Nachum, Mengjiao Sherry Yang, Lisa Lee, Daniel Freeman, Sergio Guadarrama, Ian Fischer, Winnie Xu, Eric Jang, Henryk Michalewski, et al. 2022. Multi-game decision transformers. *Advances in Neural Information Processing Systems* 35:27921–27936.

Lekang, Tyler, and Andrew Lamperski. 2019. Simple algorithms for dueling bandits. *arXiv preprint arXiv:1906.07611*.

Lepird, John R, Michael P Owen, and Mykel J Kochenderfer. 2015. Bayesian preference elicitation for multiobjective engineering design optimization. *Journal of Aerospace Information Systems* 12(10):634–645.

Lester, Brian, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Lewis, David D. 1995. A sequential algorithm for training text classifiers: Corrigendum and additional data. In *Acm sigir forum*, vol. 29, 13–19. ACM New York, NY, USA.

Li, Anqi, Dipendra Misra, Andrey Kolobov, and Ching-An Cheng. 2024a. Survival instinct in offline reinforcement learning. *Advances in neural information processing systems* 36.

Li, Jiayi, Hongyan Zhang, Liangpei Zhang, Xin Huang, and Lefei Zhang. 2014. Joint collaborative representation with multitask learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* 52(9):5923–5936.

Li, Lanqing, Rui Yang, and Dijun Luo. 2020. Focal: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. *arXiv preprint arXiv:2010.01112*.

Li, Lihong, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on world wide web*, 661–670.

Li, Lihong, Wei Chu, John Langford, and Xuanhui Wang. 2011. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth acm international conference on web search and data mining*, 297–306.

Li, Lihong, Yu Lu, and Dengyong Zhou. 2017a. Provably optimal algorithms for generalized linear contextual bandits. In *Proceedings of the 34th international conference on machine learning*, 2071–2080.

———. 2017b. Provably optimal algorithms for generalized linear contextual bandits. In *International conference on machine learning*, 2071–2080. PMLR.

Li, Lihong, Rémi Munos, and Csaba Szepesvári. 2015. Toward minimax off-policy value estimation. In *Artificial intelligence and statistics*, 608–616. PMLR.

Li, Shuai, Baoxiang Wang, Shengyu Zhang, and Wei Chen. 2016. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd international conference on machine learning*, 1245–1253.

Li, Ting, Chengchun Shi, Jianing Wang, Fan Zhou, et al. 2024b. Optimal treatment allocation for efficient policy evaluation in sequential decision making. *Advances in Neural Information Processing Systems* 36.

Li, Yingcong, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and stability in in-context learning. In *International conference on machine learning*, 19565–19594. PMLR.

Liang, Qingkai, Fanyu Que, and Eytan Modiano. 2018. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1802.06480*.

Lin, Licong, Yu Bai, and Song Mei. 2023. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *arXiv preprint arXiv:2310.08566*.

Liu, Aixin, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, Evan Z, Aditi Raghunathan, Percy Liang, and Chelsea Finn. 2021. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In *International conference on machine learning*, 6925–6935. PMLR.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys* 55(9):1–35.

Liu, Xiaoqian, Jianbin Jiao, and Junge Zhang. 2023b. Self-supervised pretraining for decision foundation model: Formulation, pipeline and challenges. *arXiv preprint arXiv:2401.00031*.

Liu, Xin, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023c. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*.

Liu, Yajing, Yuning Lu, Hao Liu, Yaozu An, Zhuoran Xu, Zhuokun Yao, Baofeng Zhang, Zhiwei Xiong, and Chenguang Gui. 2023d. Hierarchical prompt learning for multi-task learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 10888–10898.

Liu, Yao, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. 2019. Off-policy policy gradient with state distribution correction. *arXiv preprint arXiv:1904.08473*.

———. 2020. Provably good batch off-policy reinforcement learning without great exploration. *Advances in neural information processing systems* 33:1264–1274.

Liu, Zhihan, Hao Hu, Shenao Zhang, Hongyi Guo, Shuqi Ke, Boyi Liu, and Zhaoran Wang. 2023e. Reason for future, act for now: A principled framework for autonomous llm agents with provable sample efficiency. *arXiv preprint arXiv:2309.17382*.

livehealthily. livehealthily. https://www.livehealthily.com/.

Lu, Chris, Yannick Schroecker, Albert Gu, Emilio Parisotto, Jakob Foerster, Satinder Singh, and Feryal Behbahani. 2023. Structured state space models for in-context reinforcement learning. *arXiv preprint arXiv:2303.03982*.

Lu, Yangyi, Amirhossein Meisami, and Ambuj Tewari. 2021. Low-rank generalized linear bandit problems. In *International conference on artificial intelligence and statistics*, 460–468. PMLR.

Luce, Robert Duncan. 2005. *Individual choice behavior: A theoretical analysis*. Dover Publications.

Luo, Yunan, Xinbin Zhao, Jingtian Zhou, Jinglin Yang, Yanqing Zhang, Wenhua Kuang, Jian Peng, Ligong Chen, and Jianyang Zeng. 2017. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nature communications* 8(1):573.

Ma, Yi, Chenjun Xiao, Hebin Liang, and Jianye Hao. 2023. Rethinking decision transformer via hierarchical reinforcement learning. *arXiv preprint arXiv:2311.00267*.

Madotto, Andrea, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Magureanu, Stefan, Richard Combes, and Alexandre Proutiere. 2014. Lipschitz bandits: Regret lower bound and optimal algorithms. In *Conference on learning theory*, 975–999. PMLR.

Maillard, Odalric-Ambrym, and Shie Mannor. 2014. Latent bandits. In *International conference on machine learning*, 136–144. PMLR.

Margatina, Katerina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. 2023. Active learning principles for in-context learning with large language models. *arXiv preprint arXiv:2305.14264*.

Markelle Kelly, Kolby Nottingham, Rachel Longjohn. 1988. The uci machine learning repository.

Mason, Blake, Romain Camilleri, Subhojyoti Mukherjee, Kevin Jamieson, Robert Nowak, and Lalit Jain. 2021. Nearly optimal algorithms for level set estimation. *arXiv preprint arXiv:2111.01768*.

Mason, Blake, Kwang-Sung Jun, and Lalit Jain. 2022. An experimental design approach for regret minimization in logistic bandits. In *Proceedings of the aaai conference on artificial intelligence*.

Massart, Pascal. 2007. *Concentration inequalities and model selection: Ecole d'eté de probabilités de saint-flour xxxiii-2003*. Springer.

Maurer, Andreas, and Massimiliano Pontil. 2009. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*.

Maurer, Andreas, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The benefit of multitask representation learning. *Journal of Machine Learning Research* 17(81):1–32.

Mazumdar, Abhijit, Rafal Wisniewski, and Manuela L Bujorianu. 2024. Safe reinforcement learning for constrained markov decision processes with stochastic stopping time. *arXiv preprint arXiv:2403.15928*.

Mehta, Viraj, Vikramjeet Das, Ojash Neopane, Yijia Dai, Ilija Bogunovic, Jeff Schneider, and Willie Neiswanger. 2023. Sample efficient reinforcement learning from human feedback via active exploration. *CoRR* abs/2312.00267.

Ménard, Pierre, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. 2020. Fast active learning for pure exploration in reinforcement learning. *arXiv preprint arXiv:2007.13442*.

Min, Sewon, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Minsker, Stanislav. 2018. Sub-gaussian estimators of the mean of a random matrix with heavy-tailed entries. *The Annals of Statistics* 46(6A):2871–2903.

Mirchandani, Suvir, Fei Xia, Pete Florence, Brian Ichter, Danny Driess, Montserrat Gonzalez Arenas, Kanishka Rao, Dorsa Sadigh, and Andy

Zeng. 2023. Large language models as general pattern machines. *arXiv preprint arXiv:2307.04721*.

Mishra, Nikhil, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Mitchell, Eric, Rafael Rafailov, Xue Bin Peng, Sergey Levine, and Chelsea Finn. 2021. Offline meta-reinforcement learning with advantage weighting. In *International conference on machine learning*, 7780–7791. PMLR.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Moradipari, Ahmadreza, Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. 2021. Safe linear thompson sampling with side information. *IEEE Transactions on Signal Processing* 69:3755–3767.

MS MARCO. 2016. MS MARCO Dataset. https://microsoft.github.io/msmarco/.

Mukherjee, Subhojyoti, Josiah P Hanna, and Robert Nowak. 2024a. Saver: Optimal data collection strategy for safe policy evaluation in tabular mdp. *arXiv preprint arXiv:2406.02165*.

Mukherjee, Subhojyoti, Josiah P Hanna, and Robert D Nowak. 2022a. Revar: Strengthening policy evaluation via reduced variance sampling. In *Uncertainty in artificial intelligence*, 1413–1422. PMLR.

Mukherjee, Subhojyoti, Josiah P Hanna, Qiaomin Xie, and Robert Nowak. 2024b. Pretraining decision transformers with reward prediction for in-context multi-task structured bandit learning. *arXiv preprint arXiv:2406.05064*.

Mukherjee, Subhojyoti, Anusha Lalitha, Kousha Kalantari, Aniket Deshmukh, Ge Liu, Yifei Ma, and Branislav Kveton. 2024c. Optimal design for human preference elicitation.

Mukherjee, Subhojyoti, Anusha Lalitha, Sailik Sengupta, Aniket Deshmukh, and Branislav Kveton. 2024d. Multi-objective alignment of large language models through hypervolume maximization. *arXiv preprint arXiv:2412.05469*.

Mukherjee, Subhojyoti, Ge Liu, Aniket Deshmukh, Anusha Lalitha, Yifei Ma, and Branislav Kveton. 2024e. Experimental design for active transductive inference in large language models. *arXiv preprint arXiv:2404.08846*.

Mukherjee, Subhojyoti, KP Naveen, Nandan Sudarsanam, and Balaraman Ravindran. 2018. Efficient-ucbv: An almost optimal algorithm using variance estimates. In *Proceedings of the aaai conference on artificial intelligence*, vol. 32.

Mukherjee, Subhojyoti, Ardhendu S Tripathy, and Robert Nowak. 2022b. Chernoff sampling for active testing and extension to active regression. In *International conference on artificial intelligence and statistics*, 7384–7432. PMLR.

Mukherjee, Subhojyoti, Qiaomin Xie, Josiah Hanna, and Robert Nowak. 2023a. Speed: Experimental design for policy evaluation in linear heteroscedastic bandits. *arXiv preprint arXiv:2301.12357*.

———. 2024f. Multi-task representation learning for pure exploration in bilinear bandits. *Advances in Neural Information Processing Systems* 36.

Mukherjee, Subhojyoti, Qiaomin Xie, Josiah P Hanna, and Robert Nowak. 2023b. Multi-task representation learning for pure exploration in bilinear bandits. *arXiv preprint arXiv:2311.00327*.

———. 2024g. Speed: Experimental design for policy evaluation in linear heteroscedastic bandits. In *International conference on artificial intelligence and statistics*, 2962–2970. PMLR.

Mukherjee, Subhojyoti, Ruihao Zhu, and Branislav Kveton. 2023c. Efficient and interpretable bandit algorithms. *arXiv preprint arXiv:2310.14751*.

Müller, Samuel, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. 2021. Transformers can do bayesian inference. *arXiv preprint arXiv:2112.10510*.

Munos, Rémi. 2005. Error bounds for approximate value iteration. In *Proceedings of the national conference on artificial intelligence*, vol. 20, 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Nagabandi, Anusha, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. 2018. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*.

Nemhauser, G. L., L. A. Wolsey, and M. L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming* 14(1):265–294.

Neufeld, James, Andras Gyorgy, Csaba Szepesvári, and Dale Schuurmans. 2014. Adaptive monte carlo via bandit allocation. In *International conference on machine learning*, 1944–1952. PMLR.

Neyshabur, Behnam, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. 2017. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*.

Nocedal, Jorge, and Stephen J Wright. 1999. *Numerical optimization*. Springer.

Novoseller, Ellen, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. 2020. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on uncertainty in artificial intelligence*, 1029–1038. PMLR.

Oosterhuis, Harrie, and Maarten de Rijke. 2020. Taking the Counterfactual Online: Efficient and Unbiased Online Evaluation for Ranking. *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval* 137–144. ArXiv: 2007.12719.

Opoku-Agyemang, Kweku A. 2023. Randomized controlled trials via reinforcement learning from human feedback.

Osband, Ian, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. 2016. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems* 29.

Ouhamma, Reda, Debabrota Basu, and Odalric Maillard. 2023. Bilinear exponential family of mdps: frequentist regret bound with tractable exploration & planning. In *Proceedings of the aaai conference on artificial intelligence*, vol. 37, 9336–9344.

Ouyang, Long, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35: 27730–27744.

O'Donoghue, Brendan, Ian Osband, Remi Munos, and Volodymyr Mnih. 2018. The uncertainty bellman equation and exploration. In *International conference on machine learning*, 3836–3845.

Pacchiano, Aldo, Mohammad Ghavamzadeh, Peter Bartlett, and Heinrich Jiang. 2021. Stochastic bandits with linear constraints. In *International conference on artificial intelligence and statistics*, 2827–2835. PMLR.

Pal, Soumyabrata, Arun Sai Suggala, Karthikeyan Shanmugam, and Prateek Jain. 2023. Optimal algorithms for latent bandits with cluster structure. In *International conference on artificial intelligence and statistics*, 7540–7577. PMLR.

Pasztor, Barna, Parnian Kassraie, and Andreas Krause. 2024. Bandits with preference feedback: A stackelberg game perspective. *CoRR* abs/2406.16745.

Pathak, Deepak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, vol. 2017.

Pavse, Brahma, Ishan Durugkar, Josiah Hanna, and Peter Stone. 2020. Reducing sampling error in batch temporal difference learning. In *International conference on machine learning*, 7543–7552. PMLR.

Penedo, Guilherme, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*. 2306.01116.

Perera, Pramuditha, Matthew Trager, Luca Zancato, Alessandro Achille, and Stefano Soatto. 2023. Prompt algebra for task composition. *arXiv preprint arXiv:2306.00310*.

Perkins, Theodore J, and Doina Precup. 1999. Using options for knowledge transfer in reinforcement learning title2.

Perlitz, Yotam, Ariel Gera, Michal Shmueli-Scheuer, Dafna Sheinwald, Noam Slonim, and Liat Ein-Dor. 2023. Active learning for natural language generation. *arXiv preprint arXiv:2305.15040*.

Petersen, Kaare, and Michael Pedersen. 2012. The matrix cookbook. http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html.

Plackett, Robin L. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics* 24(2):193–202.

Pong, Vitchyr H, Ashvin V Nair, Laura M Smith, Catherine Huang, and Sergey Levine. 2022. Offline meta-reinforcement learning with online self-supervision. In *International conference on machine learning*, 17811–17829. PMLR.

Pukelsheim, Friedrich. 2006. *Optimal design of experiments*. SIAM.

Puterman, Martin L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qiu, Shuang, Xiaohan Wei, Zhuoran Yang, Jieping Ye, and Zhaoran Wang. 2020. Upper confidence primal-dual reinforcement learning for cmdp with adversarial loss. *Advances in Neural Information Processing Systems* 33:15277–15287.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1(8):9.

Radlinski, Filip, Robert Kleinberg, and Thorsten Joachims. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th international conference on machine learning*, 784–791.

Rafailov, Rafael, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimiza-

tion: Your language model is secretly a reward model. In *Thirty-seventh conference on neural information processing systems*.

———. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36.

Rakelly, Kate, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. 2019. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, 5331–5340. PMLR.

Rashidinejad, Paria, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. 2021. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems* 34:11702–11716.

Reed, Scott, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. 2022. A generalist agent. *arXiv preprint arXiv:2205.06175*.

Ren, Pengzhen, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM Computing Surveys (CSUR)* 54(9):1–40.

Resnick, Sidney. 2019. *A probability path*. Springer.

Reyes, Laura J Padilla, Natalia Bonifaz Oviedo, Edgar C Camacho, and Juan M Calderon. 2021. Adaptable recommendation system for outfit selection with deep learning approach. *IFAC-PapersOnLine* 54(13):605–610.

Rigollet, Phillippe, and Jan-Christian Hütter. 2015. High dimensional statistics. *Lecture notes for course 18S997* 813(814):46.

Riquelme, Carlos, Mohammad Ghavamzadeh, and Alessandro Lazaric. 2017. Active learning for accurate estimation of linear models. In *International conference on machine learning*, 2931–2939. PMLR.

Riquelme, Carlos, George Tucker, and Jasper Snoek. 2018. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*.

Roberts, Adam, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J Liu, Sharan Narang, Wei Li, and Yanqi Zhou. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer.

Rockafellar, R. 2015. Convex analysis. princeton landmarks in mathematics and physics.

Rothfuss, Jonas, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. 2018. Promp: Proximal meta-policy search. *arXiv preprint arXiv:1810.06784*.

Rubinstein, Reuven Y., and Dirk P. Kroese. 2013. *The cross-entropy method: a unified approach to combinatorial optimization, Monte Carlo simulation and machine learning*. Springer Science & Business Media.

Rusmevichientong, Paat, and John N Tsitsiklis. 2010. Linearly parameterized bandits. *Mathematics of Operations Research* 35(2):395–411.

Russo, Daniel J, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. 2018. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning* 11(1):1–96.

Sadigh, Dorsa, Anca Dragan, Shankar Sastry, and Sanjit Seshia. 2017. *Active preference-based learning of reward functions*.

Saha, Aadirupa. 2021. Optimal algorithms for stochastic contextual preference bandits. In *Advances in neural information processing systems 34*.

Saha, Aadirupa, and Pierre Gaillard. 2022. Versatile dueling bandits: Best-of-both-world analyses for online learning from preferences. *arXiv preprint arXiv:2202.06694*.

Saha, Aadirupa, and Akshay Krishnamurthy. 2022. Efficient and optimal algorithms for contextual dueling bandits under realizability. In *Proceedings of the 33rd international conference on algorithmic learning theory*.

Saha, Aadirupa, Aldo Pacchiano, and Jonathan Lee. 2023. Dueling rl: Reinforcement learning with trajectory preferences. In *International conference on artificial intelligence and statistics*, 6263–6289. PMLR.

Schaul, Tom, and Jürgen Schmidhuber. 2010. Metalearning. *Scholarpedia* 5(6):4650.

Sekhari, Ayush, Karthik Sridharan, Wen Sun, and Runzhe Wu. 2024. Contextual bandits and imitation learning with preference-based active queries. *Advances in Neural Information Processing Systems* 36.

Semnani, Sina, Violet Yao, Heidi Zhang, and Monica Lam. 2023. Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia. In *Findings of the association for computational linguistics: Emnlp 2023*, 2387–2413.

Sener, Ozan, and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Settles, Burr. 2009. Active learning literature survey.

———. 2011. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with aistats 2010*, 1–18. JMLR Workshop and Conference Proceedings.

Shafiullah, Nur Muhammad, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. 2022. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems* 35:22955–22968.

Shamir, Ohad. 2011. A variant of azuma's inequality for martingales with subgaussian tails. *arXiv preprint arXiv:1110.2392*.

Shen, Qian, Siteng Han, Yu Han, and Xi Chen. 2023a. User review analysis of dating apps based on text mining. *Plos one* 18(4):e0283896.

Shen, Tianhao, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. 2023b. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*.

Siegel, Noah Y, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. 2020. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*.

Sinii, Viacheslav, Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, and Sergey Kolesnikov. 2023. In-context reinforcement learning for variable action spaces. *arXiv preprint arXiv:2312.13327*.

Soare, Marta, Alessandro Lazaric, and Rémi Munos. 2014. Best-arm identification in linear bandits. *Advances in Neural Information Processing Systems* 27.

Soudry, Daniel, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. 2018. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research* 19(70):1–57.

Srivastava, Aarohi, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya

Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Stein, Charles, Persi Diaconis, Susan Holmes, and Gesine Reinert. 2004. Use of exchangeable pairs in the analysis of simulations. *Lecture Notes-Monograph Series* 1–26.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2020. Energy and policy considerations for modern deep learning research. In *Proceedings of the aaai conference on artificial intelligence*, vol. 34, 13693–13696.

Su, Hongjin, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2022. One embedder, any task: Instruction-finetuned text embeddings.

Su, Yi, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. 2020. Doubly robust off-policy evaluation with shrinkage. In *International conference on machine learning*, 9167–9176. PMLR.

Sui, Yanan, Masrour Zoghi, Katja Hofmann, and Yisong Yue. 2018. Advancements in dueling bandits. In *Ijcai*, 5502–5510.

Sutton, Richard S. 1988. Learning to predict by the methods of temporal differences. *Machine learning* 3(1):9–44.

Sutton, Richard S, and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Suzgun, Mirac, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Swaminathan, Adith, and Thorsten Joachims. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd international conference on machine learning*, 814–823.

Swaminathan, Adith, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. 2017. Off-policy evaluation for slate recommendation. *Advances in Neural Information Processing Systems* 30.

Szita, István. 2012. Reinforcement learning in games. *Reinforcement Learning: State-of-the-art* 539–577.

Szörényi, Balázs, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. 2015. Online rank elicitation for plackett-luce: A dueling bandits approach. *Advances in neural information processing systems* 28.

Takeno, Shion, Masahiro Nomura, and Masayuki Karasuyama. 2023. Towards practical preferential Bayesian optimization with skew Gaussian processes. In *Proceedings of the 40th international conference on machine learning*.

Talebi, Mohammad Sadegh, and Odalric-Ambrym Maillard. 2019. Learning multiple markov chains via adaptive allocation. *arXiv preprint arXiv:1905.11128*.

Thompson, William R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika* 25(3-4):285–294.

Tomkins, Sabina, Peng Liao, Predrag Klasnja, Serena Yeung, and Susan Murphy. 2020. Rapidly personalizing mobile health treatment policies with limited data. *arXiv preprint arXiv:2002.09971*.

Tong, Simon, and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2(Nov):45–66.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tripuraneni, Nilesh, Chi Jin, and Michael Jordan. 2021. Provable meta-learning of linear representations. In *International conference on machine learning*, 10434–10443. PMLR.

Tripuraneni, Nilesh, Michael Jordan, and Chi Jin. 2020. On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems* 33:7852–7862.

Tsybakov. 2009. Introduction to nonparametric estimation.

Tsybakov, Alexandre B. 2008. *Introduction to nonparametric estimation*. Springer Science & Business Media.

Tucker, Aaron David, and Thorsten Joachims. 2022a. Variance-Optimal Augmentation Logging for Counterfactual Evaluation in Contextual Bandits. *arXiv:2202.01721 [cs]*. ArXiv: 2202.01721.

———. 2022b. Variance-optimal augmentation logging for counterfactual evaluation in contextual bandits. *arXiv preprint arXiv:2202.01721*.

Turchetta, Matteo, Felix Berkenkamp, and Andreas Krause. 2019. Safe exploration for interactive machine learning. In *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurips 2019, december 8-14, 2019, vancouver, bc, canada*, ed. Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, 2887–2897.

Uehara, Masatoshi, Xuezhou Zhang, and Wen Sun. 2021. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*.

Valko, Michal, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. 2013. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*.

Valko, Michal, Rémi Munos, Branislav Kveton, and Tomáš Kocák. 2014. Spectral bandits for smooth graph functions. In *International conference on machine learning*, 46–54. PMLR.

Vanschoren, Joaquin, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2013. Openml: Networked science in machine learning. *SIGKDD Explorations* 15(2):49–60.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30.

Vaswani, Sharan, Lin F Yang, and Csaba Szepesvári. 2022. Near-optimal sample complexity bounds for constrained mdps. *arXiv preprint arXiv:2206.06270*.

Vershynin, Roman. 2020. High-dimensional probability. *University of California, Irvine*.

Voloshin, Cameron, Hoang M Le, Nan Jiang, and Yisong Yue. 2019. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*.

Voorhees, EM. 1999. Proceedings of the 8th text retrieval conference. *TREC-8 Question Answering Track Report* 77–82.

Wachi, Akifumi, Wataru Hashimoto, Xun Shen, and Kazumune Hashimoto. 2024. Safe exploration in reinforcement learning: A generalized formulation and algorithms. *Advances in Neural Information Processing Systems* 36.

Wachi, Akifumi, and Yanan Sui. 2020. Safe reinforcement learning in constrained markov decision processes. In *International conference on machine learning*, 9797–9806. PMLR.

Wagenmaker, Andrew, and Kevin G Jamieson. 2022. Instance-dependent near-optimal policy identification in linear mdps via online experiment design. *Advances in Neural Information Processing Systems* 35:5968–5981.

Wagenmaker, Andrew, Max Simchowitz, and Kevin Jamieson. 2021. Beyond no regret: Instance-dependent pac reinforcement learning. *arXiv preprint arXiv:2108.02717*.

Wagenmaker, Andrew J, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. 2022a. Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International conference on machine learning*, 22430–22456. PMLR.

Wagenmaker, Andrew J, Max Simchowitz, and Kevin Jamieson. 2022b. Beyond no regret: Instance-dependent pac reinforcement learning. In *Conference on learning theory*, 358–418. PMLR.

Wainwright, Martin J. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*, vol. 48. Cambridge university press.

Wan, Runzhe, Branislav Kveton, and Rui Song. 2022. Safe exploration for efficient policy evaluation and comparison. *arXiv preprint arXiv:2202.13234*.

Wang, Dan, and Yi Shang. 2014. A new active labeling method for deep learning. In *2014 international joint conference on neural networks (ijcnn)*, 112–119. IEEE.

Wang, Jane X, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. 2016. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, Tao, Wenbo Du, Chunxiao Jiang, Yumeng Li, and Haijun Zhang. 2024. Safety constrained trajectory optimization for completion time minimization for uav communications. *IEEE Internet of Things Journal*.

Wang, Xinyi, Wanrong Zhu, and William Yang Wang. 2023a. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*.

Wang, Xu, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. 2022. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems* 35(4): 5064–5078.

Wang, Yihan, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. 2023b. Universality and limitations of prompt tuning. *arXiv preprint arXiv:2305.18787*.

Wang, Yining, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, 25–54. PMLR.

Wang, Yu-Xiang, Alekh Agarwal, and Miroslav Dudık. 2017. Optimal and adaptive off-policy evaluation in contextual bandits. In *International conference on machine learning*, 3589–3597. PMLR.

Wang, Yuanhao, Qinghua Liu, and Chi Jin. 2023c. Is rlhf more difficult than standard rl? a theoretical perspective. In *Thirty-seventh conference on neural information processing systems*.

Wang, Zhendong, Yifan Jiang, Yadong Lu, Yelong Shen, Pengcheng He, Weizhu Chen, Zhangyang Wang, and Mingyuan Zhou. 2023d. In-context learning unlocked for diffusion models. *arXiv preprint arXiv:2305.01115*.

Wei, Jason, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35:24824–24837.

Weisz, Gellért, Philip Amortila, and Csaba Szepesvári. 2021. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic learning theory*, 1237–1264. PMLR.

Wen, Yuxin, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *arXiv preprint arXiv:2302.03668*.

Whittle, P. 1958. A multivariate generalization of tchebichev's inequality. *The Quarterly Journal of Mathematics* 9(1):232–240.

Wirth, Christian, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. 2017. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research* 18(136):1–46.

Wolke, R., and H. Schwetlick. 1988. Iteratively reweighted least squares: Algorithms, convergence analysis, and numerical comparisons. *SIAM Journal on Scientific and Statistical Computing* 9(5):907–921.

Wu, Runzhe, and Wen Sun. 2023. Making rl with preference-based feedback efficient via randomization. *arXiv preprint arXiv:2310.14554*.

Wu, Yifan, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. 2016. Conservative bandits. In *International conference on machine learning*, 1254–1262. PMLR.

Wu, Yifan, George Tucker, and Ofir Nachum. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.

Wu, Zhiyong, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning. *arXiv preprint arXiv:2212.10375*.

Xi, Zhiheng, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Xie, Sang Michael, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Xiong, Nuoya, Yihan Du, and Longbo Huang. 2024. Provably safe reinforcement learning with step-wise violation constraints. *Advances in Neural Information Processing Systems* 36.

Xiong, Wei, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. 2023. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*.

Xu, Frank F, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. 2024a. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*.

Xu, Wenjie, Wenbin Wang, Yuning Jiang, Bratislav Svetozarevic, and Colin Jones. 2024b. Principled preferential Bayesian optimization. In *Proceedings of the 41th international conference on machine learning*.

Xu, Yichong, Aparna Joshi, Aarti Singh, and Artur Dubrawski. 2020a. Zeroth order non-convex optimization with dueling-choice bandits. In *Proceedings of the 36th conference on uncertainty in artificial intelligence*.

Xu, Yichong, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. 2020b. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems* 33:18784–18794.

Yang, Adam X, Maxime Robeyns, Xi Wang, and Laurence Aitchison. 2023. Bayesian low-rank adaptation for large language models. *arXiv preprint arXiv:2308.13111*.

Yang, Jiaqi, Wei Hu, Jason D Lee, and Simon S Du. 2020. Impact of representation learning in linear bandits. *arXiv preprint arXiv:2010.06531*.

Yang, Jiaqi, Wei Hu, Jason D Lee, and Simon Shaolei Du. 2021a. Impact of representation learning in linear bandits. In *International conference on learning representations*.

Yang, Jiaqi, Qi Lei, Jason D Lee, and Simon S Du. 2022a. Nearly minimax algorithms for linear bandits with shared representation. *arXiv preprint arXiv:2203.15664*.

Yang, Junwen, and Vincent Tan. 2022. Minimax optimal fixed-budget best arm identification in linear bandits. In *Advances in neural information processing systems 35*.

Yang, Junwen, and Vincent YF Tan. 2021. Minimax optimal fixed-budget best arm identification in linear bandits. *arXiv preprint arXiv:2105.13017*.

Yang, Lin, and Mengdi Wang. 2020. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International conference on machine learning*, 10746–10756. PMLR.

Yang, Mengjiao, Dale Schuurmans, Pieter Abbeel, and Ofir Nachum. 2022b. Dichotomy of control: Separating what you can control from what you cannot. *arXiv preprint arXiv:2210.13435*.

Yang, Yunchang, Tianhao Wu, Han Zhong, Evrard Garcelon, Matteo Pirotta, Alessandro Lazaric, Liwei Wang, and Simon Shaolei Du. 2021b. A reduction-based framework for conservative bandits and reinforcement learning. In *International conference on learning representations*.

Yang, Zhaoxing, Haiming Jin, Yao Tang, and Guiyun Fan. 2024. Risk-aware constrained reinforcement learning with non-stationary policies. In *Proceedings of the 23rd international conference on autonomous agents and multiagent systems*, 2029–2037.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32.

Yin, Shukang, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*.

Ying, Donghao, Yunkai Zhang, Yuhao Ding, Alec Koppel, and Javad Lavaei. 2024. Scalable primal-dual actor-critic method for safe multi-agent rl with general utilities. *Advances in Neural Information Processing Systems* 36.

Yu, Chao, Jiming Liu, and Shamim Nemati. 2019. Reinforcement learning in healthcare: a survey. arxiv. *arXiv preprint arXiv:1908.08796*.

Yu, Tianhe, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. 2021. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems* 34:28954–28967.

Yu, Youngjae, Jiwan Chung, Heeseung Yun, Jack Hessel, Jae Sung Park, Ximing Lu, Rowan Zellers, Prithviraj Ammanabrolu, Ronan Le Bras, Gunhee Kim, et al. 2023. Fusing pre-trained language models with multimodal prompts through reinforcement learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 10845–10856.

Yu, Youngjae, Jiwan Chung, Heeseung Yun, Jack Hessel, JaeSung Park, Ximing Lu, Prithviraj Ammanabrolu, Rowan Zellers, Ronan Le Bras, Gunhee Kim, et al. 2022. Multimodal knowledge alignment with reinforcement learning. *arXiv preprint arXiv:2205.12630*.

Yuan, Michelle, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. *arXiv preprint arXiv:2010.09535*.

Yue, Yisong, Josef Broder, Robert Kleinberg, and Thorsten Joachims. 2012. The k-armed dueling bandits problem. *Journal of Computer and System Sciences* 78(5):1538–1556.

Zambaldi, Vinicius, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. 2018. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*.

Zanette, Andrea. 2021. *Reinforcement learning: When can we do sample efficient exploration?* Stanford University.

Zanette, Andrea, and Emma Brunskill. 2019. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International conference on machine learning*, 7304–7312. PMLR.

Zanette, Andrea, Mykel J Kochenderfer, and Emma Brunskill. 2019. Almost horizon-free structure-aware best policy identification with a generative model. *Advances in Neural Information Processing Systems* 32.

Zanette, Andrea, Martin J Wainwright, and Emma Brunskill. 2021. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems* 34:13626–13640.

Zhan, Wenhao, Masatoshi Uehara, Wen Sun, and Jason Lee. 2024. Provable reward-agnostic preference-based reinforcement learning. In *Proceedings of the 12th international conference on learning representations*.

Zhan, Xueying, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B Chan. 2022. A comparative survey of deep active learning. *arXiv preprint arXiv:2203.13450*.

Zhang, Daoqiang, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. 2012. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *NeuroImage* 59(2):895–907.

Zhang, Yiming, Shi Feng, and Chenhao Tan. 2022a. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*.

Zhang, Yuanhan, Kaiyang Zhou, and Ziwei Liu. 2023. What makes good examples for visual in-context learning? *arXiv preprint arXiv:2301.13670*.

Zhang, Zhuosheng, Aston Zhang, Mu Li, and Alex Smola. 2022b. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Zhang, Zihan, Jiaqi Yang, Xiangyang Ji, and Simon S Du. 2021. Improved variance-aware confidence sets for linear bandits and linear mixture mdp. *Advances in Neural Information Processing Systems* 34:4342–4355.

Zhao, Andrew, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.

Zhao, Heyang, Dongruo Zhou, Jiafan He, and Quanquan Gu. 2022. Bandit learning with general function classes: Heteroscedastic noise and variance-dependent regret bounds. *arXiv preprint arXiv:2202.13603*.

Zhao, Yufan, Michael R Kosorok, and Donglin Zeng. 2009. Reinforcement learning design for cancer clinical trials. *Statistics in medicine* 28(26):3294–3315.

Zheng, Yinan, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. 2024. Safe offline reinforcement learning with feasibility-guided diffusion model. *arXiv preprint arXiv:2401.10700*.

Zhong, Rujie, Duohan Zhang, Lukas Schäfer, Stefano V. Albrecht, and Josiah P. Hanna. 2022a. Robust on-policy sampling for data-efficient policy evaluation. In *Proceedings of advances in neural information processing systems (neurips)*.

Zhong, Rujie, Duohan Zhang, Lukas Schäfer, Stefano V. Albrecht, and Josiah P. Hanna. 2022b. Robust On-Policy Sampling for Data-Efficient Policy Evaluation in Reinforcement Learning. In *Proceedings of Neural and Information Processing Systems (NeurIPS)*.

Zhou, Dongruo, and Quanquan Gu. 2022. Computationally efficient horizon-free reinforcement learning for linear mixture mdps. *arXiv preprint arXiv:2205.11507*.

Zhou, Dongruo, Quanquan Gu, and Csaba Szepesvari. 2021. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on learning theory*, 4532–4576. PMLR.

Zhou, Dongruo, Lihong Li, and Quanquan Gu. 2020. Neural contextual bandits with ucb-based exploration. In *International conference on machine learning*, 11492–11502. PMLR.

Zhou, Tianchen, Jia Liu, Yang Jiao, Chaosheng Dong, Yetian Chen, Yan Gao, and Yi Sun. 2023. Bandit learning to rank with position-based click models: Personalized and equal treatments. *arXiv preprint arXiv:2311.04528*.

Zhou, Xin, Nicole Mayer-Hamblett, Umer Khan, and Michael R Kosorok. 2017. Residual weighted learning for estimating individualized treatment rules. *Journal of the American Statistical Association* 112(517):169–187.

Zhu, Banghua, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023a. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

Zhu, Banghua, Jiantao Jiao, and Michael I Jordan. 2023b. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*.

Zhu, Qiuyu, and Vincent Tan. 2020. Thompson sampling algorithms for mean-variance bandits. In *International conference on machine learning*, 11599–11608. PMLR.

Zhu, Ruihao, and Branislav Kveton. 2021. Safe data collection for offline and online policy learning. *arXiv preprint arXiv:2111.04835*.

———. 2022a. Safe optimal design with applications in off-policy learning. In *Proceedings of the 25th international conference on artificial intelligence and statistics*, ed. Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, vol. 151 of *Proceedings of Machine Learning Research*, 2436–2447. PMLR.

———. 2022b. Safe optimal design with applications in off-policy learning. In *International conference on artificial intelligence and statistics*, 2436–2447. PMLR.

Zhu, Yinglun, Dongruo Zhou, Ruoxi Jiang, Quanquan Gu, Rebecca Willett, and Robert Nowak. 2021. Pure exploration in kernel and neural bandits. *Advances in neural information processing systems* 34:11618–11630.

Zintgraf, Luisa, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. 2019. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*.

Zoghi, Masrour, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. 2017. Online learning to rank in stochastic click models. In *Proceedings of the 34th international conference on machine learning*.

Zong, Shi, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. 2016. Cascading bandits for large-scale recommendation problems. In *Proceedings of the 32nd conference on uncertainty in artificial intelligence*.

A APPENDIX: REVAR: STRENGTHENING POLICY EVALUATION VIA REDUCED VARIANCE SAMPLING

A.1 Optimal Sampling in Bandit Setting

Proposition 1. (*Restatement*) *In an A-action bandit setting, the estimated return of π after n action-reward samples is denoted by Y_n as defined in (2.1). Note that the expectation of Y_n after each action has been sampled once is given by $\nu(\pi)$. Minimal MSE, $\mathbb{E}_{\mathcal{D}}\left[(Y_n - \nu(\pi))^2\right]$, is obtained by taking actions in the proportion:*

$$b^*(a) = \frac{\pi(a)\sigma(a)}{\sum_{a'=1}^{A} \pi(a')\sigma(a')}. \tag{A.1}$$

where $b^*(a)$ denotes the optimal sampling proportion.

Proof. Recall that we have a budget of n samples and we are allowed to draw samples from their respective distributions. Suppose we have $T_n(1), T_n(2), \ldots, T_n(A)$ samples from actions $1, 2, \ldots, A$. Then we can calculate the estimator

$$Y_n = \frac{1}{n}\sum_{t=1}^{n} Y_t = \sum_{a=1}^{A} \frac{\pi(a)}{T_n(a)} \sum_{i=1}^{T_n(a)} R_i(a)$$

where, $n = \sum_{a=1}^{A} T_n(a)$ samples and $R_i(a)$ is the i^{th} reward received after taking action a. We collect a dataset $\mathcal{D}$ of n action-reward samples. Now we use the MSE to estimate how close is Y_n to $\nu(\pi)$ as follows:

$$\mathbb{E}_{\mathcal{D}}\left[(Y_n - \nu(\pi))^2\right] = \text{Var}(Y_n) + \text{bias}^2(Y_n).$$

Note that once we have sampled each action once, since $\mathbb{E}_{\mathcal{D}}[Y_n] = \nu(\pi)$ so $\text{bias}(Y_n) = 0$. So we need to focus only on variance. We can decompose

the variance as follows:

$$\begin{aligned}\mathrm{Var}(Y_n) &\overset{(a)}{=} \sum_{a=1}^{A} \mathrm{Var}\left(\frac{\pi(a)}{T_n(a)} \sum_{i=1}^{T_n(a)} R_i(a)\right) \\ &= \sum_{a=1}^{A} \frac{\pi^2(a)}{T_n^2(a)} \sum_{i=1}^{T_n(a)} \mathrm{Var}\left(R_i(a)\right) = \sum_{a=1}^{A} \frac{\pi^2(a)\sigma^2(a)}{T_n(a)}\end{aligned}$$

where, (a) follows as $R_i(a)$ and $R_{i'}(a')$ are independent for every (i, i') and (a, a') pairs. Now we want to optimize $T_n(1), T_n(2), \ldots, T_n(A)$ so that the variance $\mathrm{Var}(Y_n)$ is minimized. We can do this as follows: Let's first write the variance in terms of the proportion $\mathbf{b} \coloneqq \{b(1), b(2), \ldots, b(A)\}$ such that

$$b(a) = \frac{T_n(a)}{\sum_{a'=1}^{A} T_n(a')}.$$

We can then rewrite the optimization problem as follows:

$$\begin{aligned}\min_{\mathbf{b}} \sum_{a=1}^{A} \frac{\pi^2(a)\sigma^2(a)}{b(a)}, \quad \textbf{s.t.} \sum_{a} b(a) = 1 \\ \forall a, b(a) > 0. \end{aligned} \tag{A.2}$$

Note that we use $b(a)$ to denote the optimization variable and $b^*(a)$ to denote the optimal sampling proportion. Given this optimization in (A.2) we can get a closed form solution by introducing the Lagrange multiplier as follows:

$$L(\mathbf{b}, \lambda) = \sum_{a=1}^{A} \frac{\pi^2(a)\sigma^2(a)}{b(a)} + \lambda\left(\sum_{a=1}^{A} b(a) - 1\right). \tag{A.3}$$

Now to get the Karush-Kuhn-Tucker (KKT) condition we differentiate

(A.3) with respect to $b(a)$ and λ as follows:

$$\nabla_{b(a)} L(\mathbf{b}, \lambda) = -\frac{\pi^2(a)\sigma^2(a)}{b^2(a)} + \lambda \tag{A.4}$$

$$\nabla_{\lambda} L(\mathbf{b}, \lambda) = \sum_{a} b(a) - 1. \tag{A.5}$$

Now equating (A.4) and (A.5) to zero and solving for the solution we obtain:

$$\lambda = \frac{\pi^2(a)\sigma^2(a)}{b^2(a)} \implies b(a) = \sqrt{\frac{\pi^2(a)\sigma^2(a)}{\lambda}}$$

$$\sum_{a} b(a) = 1 \implies \sum_{a=1}^{A} \sqrt{\frac{\pi^2(a)\sigma^2(a)}{\lambda}} = 1 \implies \sqrt{\lambda} = \sum_{a=1}^{A} \sqrt{\pi^2(a)\sigma^2(a)}.$$

This gives us the optimal sampling proportion

$$b^*(a) = \frac{\pi(a)\sigma(a)}{\sum_{a'=1}^{A} \sqrt{\pi(a')^2\sigma^2(a')}} \implies b^*(a) = \frac{\pi(a)\sigma(a)}{\sum_{a'=1}^{A} \pi(a')\sigma(a')}.$$

Finally, observe that the above optimal sampling for the bandit setting for an action a only depends on the standard deviation $\sigma(a)$ of the action. □

A.2 Optimal Sampling in Three State Stochastic Tree MDP

Lemma 1. (***Restatement***) *Let* $\mathbf{T}$ *be a* 2*-depth stochastic tree MDP as defined in Theorem 2.1 (see Figure A.1 in Section A.2). Let* $Y_n(s_1^1)$ *be the estimated return of the starting state* s_1^1 *after observing* n *state-action-reward samples. Note that* $v^{\pi}(s_1^1)$ *is the expectation of* $Y_n(s_1^1)$ *under Assumption 1. Let* $\mathcal{D}$ *be the observed data over* n *state-action-reward samples. To minimise MSE,* $\mathbb{E}_{\mathcal{D}}[(Y_n(s_1^1) - v^{\pi}(s_1^1))^2]$,

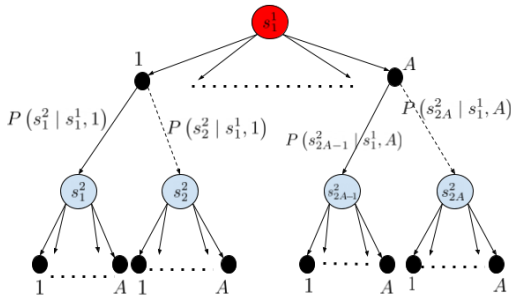


Figure A.1: 2-Depth, A-action Tree MDP

is obtained by taking actions in each state in the following proportions:

$$b^*(a|s_j^2) \propto \pi(a|s_j^2)\sigma(s_j^2, a)$$

$$b^*(a|s_1^1) \propto \sqrt{\pi^2(a|s_1^1)\left[\sigma^2(s_1^1, a) + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1, a)B^2(s_j^2)\right]},$$

where, $B(s_j^2) = \sum_a \pi(a|s_j^2)\sigma(s_j^2, a)$.

Proof. We define an estimator $Y_n(s)$ that visits each state-action pair

$$\sum_{s\in\mathcal{S}} \sum_{a} T_n(s, a) = n$$

times and then plug-ins the estimated sample mean. For the i^{th} state in level ℓ of the tree, this estimator is given as:

$$Y_n(s_i^\ell) = \sum_{a=1}^{A} \left(\underbrace{\frac{\pi(a|s_i^\ell)}{T_n(s_i^\ell, a)} \sum_{h=1}^{T_n(s_i^\ell, a)} R_h(s_i^\ell, a)}_{\text{mean reward weighted by } \pi(a|s_i^\ell)} + \gamma\pi(a|s_i^\ell) \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) \underbrace{Y_n(s_j^{\ell+1})}_{\text{Next state estimate}} \right), \text{if } \ell \neq L$$

where we take $Y_n(s_j^{l+1}) = 0$ if s_i^l is a leaf state (i.e., $l = L$).

Step 1 ($Y_n(s_1^1)$ is an unbiased estimator of $v(\pi)$): We first show that $Y_n(s_1^1)$ is an unbiased estimator of $v(\pi)$. We use this fact to show that minimizing variance is equivalent to minimizing MSE. The expectation of $Y_n(s_1^1)$ is given as:

$$
\begin{aligned}
&\mathbb{E}[Y_n(s_1^1)] \\
&= \mathbb{E}\left[\sum_{a=1}^{A}\left(\sum_{s_j^2}\frac{\pi(a|s_1^1)}{T_n(s_1^1,a)}\sum_{h=1}^{T_n(s_1^1,a)} R_h(s_1^1,a)+\gamma\pi(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)Y_n(s_j^2)\right)\right] \\
&\overset{(a)}{=} \sum_{a=1}^{A}\left(\sum_{s_j^2}\frac{\pi(a|s_1^1)}{T_n(s_1^1,a)}\sum_{h=1}^{T_n(s_1^1,a)}\mathbb{E}\left[R_h(s_1^1,a)\right]+\gamma\pi(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)\mathbb{E}\left[Y_n(s_j^2)\right]\right) \\
&= \sum_{a=1}^{A}\left(\sum_{s_j^2}\frac{\pi(a|s_1^1)}{T_n(s_1^1,a)}T_n(s_1^1,a)\mathbb{E}\left[R_h(s_1^1,a)\right]+\gamma\pi(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)\mathbb{E}\left[Y_n(s_j^2)\right]\right) \\
&= v^{\pi}(s_1^1)
\end{aligned}
$$

where, (a) follow from the linearity of expectation. Thus, $Y_n(s_1^1)$ is a unbiased estimator of $v(\pi)$.

Step 2 (Variance of $Y_n(s_1^1)$): Next we look into the variance of $\mathrm{Var}(Y_n(s_1^1))$.

$$
\begin{aligned}
\mathrm{Var}(Y_n(s_1^1)) &= \mathrm{Var}\left[\sum_{a=1}^{A}\left(\frac{\pi(a|s_1^1)}{T_n(s_1^1,a)}\sum_{h=1}^{T_n(s_1^1,a)} R_h(s_1^1,a)+\gamma\pi(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)Y_n(s_j^2)\right)\right] \\
&\overset{(a)}{=} \sum_{a=1}^{A}\left(\frac{\pi^2(a|s_1^1)}{T_n^2(s_1^1,a)}\sum_{h=1}^{T_n(s_1^1,a)}\mathrm{Var}[R_h(s_1^1,a)]+\gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}P^2(s_j^2|s_1^1,a)\mathrm{Var}[Y_n(s_j^2)]\right) \\
&\overset{(b)}{=} \sum_{a=1}^{A}\left(\frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{T_n(s_1^1,a)}+\gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}P^2(s_j^2|s_1^1,a)\mathrm{Var}[(Y_n(s_j^2)]\right),
\end{aligned}
\tag{A.6}
$$

where (a) follows because the reward in next state is conditionally inde-

pendent given the current state and action and (b) follows from $\sigma(s,a) = \mathrm{Var}[R(s,a)]$.

The goal is to reduce the variance $\mathrm{Var}(Y_n(s_1^1))$ in (A.6). We first unroll the (A.6) to take into account the conditional behavior probability of each of the path from s_1^1 to s_j^2 for $j \in \{1,2,3,4\}$. This is shown as follows:

$$\begin{aligned}
\mathrm{Var}(Y_n(s_1^1))) &= \sum_a \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{T_n(s_1^1,a)} \\
&\quad + \sum_a \sum_{s_j^2} \sum_{a'} \frac{\gamma^2\pi^2(a|s_1^1)P^2(s_j^2|s_1^1,a)\pi^2(a|s_j^2)\sigma^2(s_j^2,a)}{T_n(s_j^2,a)} \\
\implies n\mathrm{Var}(Y_n(s_1^1))) &= \sum_a \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{T_n(s_1^1,a)/n} \\
&\quad + \sum_a \sum_{s_j^2} \sum_{a'} \frac{\gamma^2\pi^2(a|s_1^1)P^2(s_j^2|s_1^1,a)\pi^2(a|s_j^2)\sigma^2(s_j^2,a')}{T_n(s_j^2,a')/n} \\
&\overset{(a)}{\implies} \sum_a \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{b(a|s_1^1)} \\
&\quad + \sum_a \sum_{s_j^2} \sum_{a'} \frac{\gamma^2\pi^2(a|s_1^1)P^2(s_j^2|s_1^1,a)\pi^2(a|s_j^2)\sigma^2(s_j^2,a')}{P(s_j^2|s_1^1,a)b(a|s_1^1)b(a'|s_j^2)}
\end{aligned}$$

where, (a) follows as

$$\begin{aligned}
b(a'|s_i^2) = \frac{T_n(s_i^2,a')}{\sum_a T_n(s_i^2,a)} &\overset{(a)}{=} \frac{T_n(s_i^2,a')}{P(s_i^2|s_1^1,a)T_n(s_1^1,a)} = \frac{T_n(s_i^2,a')/n}{P(s_i^2|s_1^1,a)T_n(s_1^1,a)/n} \\
&\implies T_n(s_i^2,a')/n = P(s_i^2|s_1^1,a)b(a|s_1^1)b(a'|s_i^2)
\end{aligned}$$

where, in (a) the action a is used from state s_1^1 to transition to state s_i^2. Similarly in (a) we can substitute $T_n(s_j^2,a)/n$ for all $s \in \mathcal{S}$. Note that this follows because of the tree MDP structure as path to state $s_j^{\ell+1}$ depends on it immediate parent state s_i^ℓ (see **(3)** in Theorem 2.1). Recall that

$P(s'|s,a)T_n(s'|s,a)$ is the expected times we end up in next state, not the actual number of times. We use P in this formulation instead of $\widehat{P}$ as this is the oracle setting which has access to the transition model and our goal is to minimize the number of samples n.

Step 3 (Minimal Variance Objective function): Note that we use $b(a|s)$ to denote the optimization variable and $b^*(a|s)$ to denote the optimal sampling proportion. Now, we determine the b values that give minimal variance by minimizing the following objective:

$$\begin{aligned}
&\min_{b} \sum_{a} \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{b(a|s_1^1)} + \sum_{a}\sum_{s_j^2}\sum_{a'} \frac{\gamma^2\pi^2(a|s_1^1)P(s_j^2|s_1^1,a)\pi^2(a|s_j^2)\sigma^2(s_j^2,a')}{b(a|s_1^1)b(a'|s_j^2)} \\
&\textbf{s.t.} \quad \forall s, \quad \sum_{a} b(a|s) = 1 \\
&\qquad \forall s,a \quad b(a|s) > 0. \qquad \text{(A.7)}
\end{aligned}$$

We can get a closed form solution by introducing the Lagrange multiplier as follows:

$$\begin{aligned}
L(\lambda,\mathbf{b}) = &\sum_{a} \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{b(a|s_1^1)} + \sum_{a}\sum_{s_j^2}\sum_{a'} \frac{\gamma^2\pi^2(a|s_1^1)P(s_j^2|s_1^1,a)\pi^2(a|s_j^2)\sigma^2(s_j^2,a')}{b(a|s_1^1)b(a'|s_j^2)} \\
&+ \sum_{s} \lambda_s \left(\sum_{a} b(a|s) - 1 \right)
\end{aligned}$$

Step 5 (Solving for KKT condition): Now we want to get the KKT condi-

tion for the Lagrangian function $L(\lambda, \mathbf{b})$ as follows:

$$\nabla_{\lambda_s} L(\lambda, \mathbf{b}) = \sum_a b(a|s) - 1 \tag{A.8}$$

$$\begin{aligned}&\nabla_{b(a|s_1^1)} L(\lambda, \mathbf{b}) \\ &= -\frac{\pi^2(a|s_1^1)\sigma^2(s_1^1, a)}{b(a|s_1^1)^2} - \gamma^2\pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{P(s_j^2|s_1^1, a)\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b^2(a|s_1^1)b(a'|s_j^2)} + \lambda_{s_1^1}\end{aligned} \tag{A.9}$$

$$\nabla_{b(a'|s_j^2)} L(\lambda, \mathbf{b}) = -\sum_{a'} \gamma^2\pi^2(a|s_1^1) \frac{P(s_j^2|s_1^1, a)\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b(a|s_1^1)b^2(a'|s_j^2)} + \lambda_{s_j^2} \tag{A.10}$$

Setting (A.10) equal to 0, we obtain:

$$\lambda_{s_j^2} = \sum_a \gamma^2\pi^2(a|s_1^1) \frac{P(s_j^2|s_1^1, a)\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b(a|s_1^1)b^2(a'|s_j^2)} \tag{A.11}$$

$$\implies b(a|s_j^2) = \sqrt{\sum_a \gamma^2\pi^2(a|s_1^1) \frac{P(s_j^2|s_1^1, a)\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b(a|s_1^1)\lambda_{s_j^2}}} \tag{A.12}$$

Finally, we eliminate $\lambda_{s_j^2}$ by setting (A.8) to 0 and using the fact that $\sum_a b(a|s_j^2) = 1$:

$$b^*(a|s_j^2) = \frac{\pi(a|s_j^2)\sigma(s_j^2, a)}{\sum_{a'} \pi(a'|s_j^2)\sigma(s_j^2, a')} \tag{A.13}$$

which gives us the optimal proportion in level 2. Similarly, setting (A.9)

equal to 0, we obtain:

$$\lambda_{s_1^1} = \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{b^2(a|s_1^1)} + \gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}\sum_{a'}\frac{P(s_j^2|s_1^1,a)\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{b^2(a|s_1^1)b(a'|s_j^2)}$$

$$\implies b(a|s_1^1) = \sqrt{\frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{\lambda_{s_1^1}} + \gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}\sum_{a'}\frac{P(s_j^2|s_1^1,a)\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{\lambda_{s_1^1}b(a'|s_j^2)}}$$

$$b(a|s_1^1) = \frac{1}{\sqrt{\lambda_{s_1^1}}}\sqrt{\pi^2(a|s_1^1)\sigma^2(s_1^1,a) + \gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}\sum_{a'}\frac{P(s_j^2|s_1^1,a)\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{b(a'|s_j^2)}}$$

$$\implies b(a|s_1^1) \overset{(a)}{=} \sqrt{\frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{\lambda_{s_1^1}} + \gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}\sum_{a'}\frac{P(s_j^2|s_1^1,a)\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{\lambda_{s_1^1}b(a'|s_j^2)}}$$

$$b^*(a|s_1^1) = \frac{1}{\sqrt{\lambda_{s_1^1}}}\sqrt{\pi^2(a|s_1^1)\sigma^2(s_1^1,a) + \gamma^2\pi^2(a|s_1^1)\sum_{s_j^2}\sum_{a'}P(s_j^2|s_1^1,a)B^2(s_j^2)}$$

where, (a) follows by plugging in the definition of $b(a'|s_j^2)$ and substituting $B(s_j^2) = \sum_a \pi(a|s_j^2)\sigma(s_j^2,a)$. This concludes the proof for the optimal sampling in the 2-depth stochastic tree MDP **T**. □

A.3 Three State Deterministic Tree Sampling

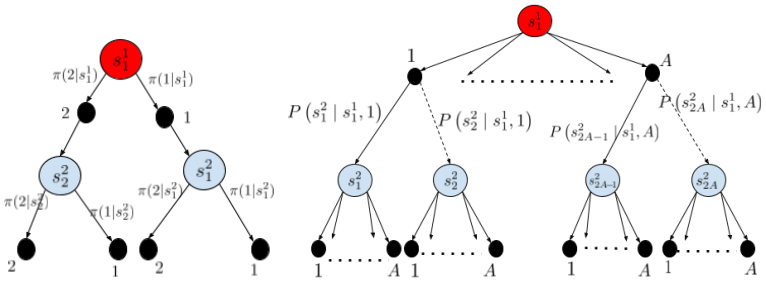


Figure A.2: (Left) Deterministic 2-depth Tree. (Right) Stochastic 2-Depth Tree with varying model.

Consider the 2-depth, 2-action deterministic tree MDP $\mathbf{T}$ in Figure A.2 (left) where we have equal target probabilities $\pi(1|s_1^1) = \pi(2|s_1^1) = \pi(1|s_1^2) = \pi(2|s_1^2) = \pi(1|s_1^2) = \pi(2|s_1^2) = \frac{1}{2}$. The variance is given by $\sigma^2(s_1^1, 1) = 400$, $\sigma^2(s_1^1, 2) = 600$, $\sigma^2(s_1^2, 1) = 400$, $\sigma^2(s_1^2, 2) = 400$, $\sigma^2(s_2^2, 1) = 4$, $\sigma^2(s_2^2, 2) = 4$. So the left sub-tree has lesser variance than right sub-tree. Let discount factor $\gamma = 1$. Then we get the optimal sampling behavior policy as follows:

$$b^*(1|s_1^2) \propto \pi(1|s_1^2)\sigma(1|s_1^2) = \frac{1}{2} \cdot 20 = 10, b^*(2|s_1^2) \propto \pi(2|s_1^2)\sigma(2|s_1^2) = \frac{1}{2} \cdot 20 = 10$$
$$b^*(1|s_2^2) \propto \pi(1|s_2^2)\sigma(1|s_2^2) = \frac{1}{2} \cdot 2 = 1, b^*(2|s_2^2) \propto \pi(2|s_2^2)\sigma(2|s_2^2) = \frac{1}{2} \cdot 2 = 1,$$
$$B(s_1^2) = \pi(1|s_1^2)\sigma(1|s_1^2) + \pi(2|s_1^2)\sigma(2|s_1^2) = 20,$$
$$B(s_2^2) = \pi(1|s_2^2)\sigma(1|s_2^2) + \pi(2|s_2^2)\sigma(2|s_2^2) = 2$$

$$b^*(1|s_1^1) \propto \sqrt{\pi^2(1|s_1^1)\left[\sigma^2(s_1^1, 1) + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1, 1)B^2(s_j^2)\right]}$$
$$= \sqrt{\pi^2(1|s_1^1)\sigma^2(s_1^1, 1) + \gamma^2\pi^2(1|s_1^1)P(s_1^2|s_1^1, 1)B^2(s_1^2) + \gamma^2\pi^2(2|s_1^1)P(s_2^2|s_1^1, 1)B^2(s_2^2)}$$
$$\overset{(a)}{=} \sqrt{400 \cdot \frac{1}{4} + \frac{1}{4} \cdot 1 \cdot 400 + \frac{1}{4} \cdot 0 \cdot 4} \approx 14$$

$$b^*(2|s_1^1) \propto \sqrt{\pi^2(2|s_1^1)\left[\sigma^2(s_1^1, 2) + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1, 2)B^2(s_j^2)\right]}$$
$$= \sqrt{\pi^2(2|s_1^1)\sigma^2(s_1^1, 2) + \gamma^2\pi^2(2|s_1^1)P(s_1^2|s_1^1, 2)B^2(s_1^2) + \gamma^2\pi^2(2|s_1^1)P(s_2^2|s_1^1, 2)B^2(s_2^2)}$$
$$\overset{(b)}{=} \sqrt{600 \cdot \frac{1}{4} + \frac{1}{4} \cdot 0 \cdot 400 + \frac{1}{4} \cdot 1 \cdot 4} \approx 12$$

where, (a) follows because $P(s_2^2|s_1^1, 1) = 0$ and (b) follows $P(s_1^2|s_1^1, 2) = 0$.

Note that $b(1|s_1^1)$ and $b(2|s_1^1)$ are un-normalized values. After normalization we can show that $b(1|s_1^1) > b(2|s_1^1)$. Hence the right sub-tree with higher variance will have higher proportion of pulls.

A.4 Three State Stochastic Tree Sampling with Varying Model

In this tree MDP **T** in Figure A.2 (right) we have $P(s_1^2|s_1^1, 1) = p$, $P(s_1^2|s_1^1, 2) = 1 - p$ and $P(s_2^2|s_1^1, 1) = p$, $P(s_2^2|s_1^1, 2) = 1 - p$. Plugging this transition probabilities from the result of Theorem 2.2 we get

$$
\begin{aligned}
b^*(a|s_j^2) &\propto \pi(a|s_j^2)\sigma(s_j^2, a), \quad \text{for } j \in \{1, 2, 3, 4\} \\
b^*(1|s_1^1) &\propto \sqrt{\pi^2(1|s_1^1)\left[\sigma^2(s_1^1, 1) + \gamma^2 p B^2(s_1^2) + \gamma^2(1-p)B^2(s_2^2)\right]}, \\
b^*(2|s_1^1) &\propto \sqrt{\pi^2(2|s_1^1)\left[\sigma^2(s_1^1, 2) + \gamma^2 p B^2(s_3^2) + \gamma^2(1-p)B^2(s_4^2)\right]}
\end{aligned}
$$

where, $B(s_j^2) = \sum_a \pi(a|s_j^2)\sigma(s_j^2, a)$. Now if $p \gg 1-p$, then we only need to consider the variance of state s_1^2 when estimating the sampling proportion for states s_1^2 and s_3^2 as

$$
\begin{aligned}
b^*(1|s_1^1) &\propto \sqrt{\pi^2(1|s_1^1)\left[\sigma^2(s_1^1, 1) + \gamma^2 p B^2(s_1^2)\right]}, \\
b^*(2|s_1^1) &\propto \sqrt{\pi^2(2|s_1^1)\left[\sigma^2(s_1^1, 2) + \gamma^2 p B^2(s_3^2)\right]}.
\end{aligned}
$$

Remark A.1. (***Transition Model Matters***) *Observe that the main goal of the optimal sampling proportion in Theorem 2.2 is to reduce the variance of the estimate of the return. However, the sampling proportion is not geared to estimate the model $\widehat{P}$ well. An interesting extension to combine the optimization problem*

in Theorem 2.2 with some model estimation procedure as in Zanette et al. (2019); Agarwal et al. (2019); Wagenmaker et al. (2021) to derive the optimal sampling proportion.

A.5 Multi-level Stochastic Tree MDP Formulation

Theorem 1. **(*Restatement*)** *Assume the underlying MDP is an* L*-depth tree MDP as defined in Theorem 2.1. Let the estimated return of the starting state* s_1^1 *after* n *state-action-reward samples be defined as* $Y_n(s_1^1)$*. Note that the* $v^\pi(s_1^1)$ *is the expectation of* $Y_n(s_1^1)$ *under Assumption 1. Let* $\mathcal{D}$ *be the observed data over* n *state-action-reward samples. To minimize the MSE,* $\mathbb{E}_{\mathcal{D}}[(Y_n(s_1^1)) - \mu(Y_n(s_1^1)))^2]$*, the optimal sampling proportions for any arbitrary state is given by:*

$$b^*(a|s_i^\ell) \propto \sqrt{\pi^2(a|s_i^\ell)\left[\sigma^2(s_i^\ell, a) + \gamma^2 \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) B^2(s_j^{\ell+1})\right]},$$

where, $B(s_j^2)$ *is the normalization factor defined as follows:*

$$B(s_i^\ell) = \sum_a \sqrt{\pi^2(a|s_i^\ell)\left(\sigma^2(s_i^\ell, a) + \gamma^2 \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) B^2(s_j^{\ell+1})\right)}$$

Proof. **Step 1 (Base case for Level** L **and** $L-1$**):** The proof of this theorem follows from induction. First consider the last level L containing the leaf states. An arbitrary state in the last level is denoted by s_i^L. Then we have

the estimate of the expected return from the state s_i^L as

$$Y_n(s_1^1) = \sum_{a=1}^{A} \pi(a|s_1^1)\left(\frac{1}{T_n(s_1^1,a)}\sum_{h=1}^{T_n(s_1^1,a)} R_h(s_1^1,a) + \gamma \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_1^1,a)Y_n(s_j^2)\right)$$
$$= \sum_{a=1}^{A} \pi(a|s_1^1)\left(\widehat{\mu}(s_1^1,a) + \gamma \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_1^1,a)Y_n(s_j^2)\right)$$

Observe that for the leaf-state the $Y_n(s_i^L)$ the transition probability to next states $P(s_j^{L+1}|s_i^L,a) = 0$ for any action a. So $Y_n(s_i^L) = \sum_{a=1}^{A}\left(\pi(a|s_i^L)\widehat{\mu}(s_i^L,a)\right)$ which matches the bandit setting. We define an estimator $Y_n(s_i^\ell)$ as defined in (2.2). Following the previous derivation in Theorem 2.2 we can show its expectation is given as:

$$\mathbb{E}[Y_n(s_i^L)] = \sum_a \frac{\pi(a|s_i^L)}{T_n(s_i^L,a)} \sum_{h=1}^{T_n(s_i^L,a)} \mathbb{E}[R_h(s_i^L,a)] = \sum_a \pi(a|s_i^L)\mu(s_i^L,a) = v^\pi(s_i^L).$$

$$\mathrm{Var}[Y_n(s_i^L)] = \sum_a \frac{\pi^2(a|s_i^L)}{T_n^2(s_i^L,a)} \sum_{h=1}^{T_n(s_i^L,a)} \mathrm{Var}[R_h(s_i^L,a)] = \sum_a \frac{\pi^2(a|s_i^L)\sigma^2(s_i^L,a)}{T_n(s_i^L,a)}$$

Now consider the second last level $L-1$ containing the leaves. An arbitrary state in the last level is denoted by s_i^{L-1}. Then we have the expected return from the state s_i^{L-1} as follows:

$$Y_n(s_i^{L-1})$$
$$= \sum_a \pi(a|s_i^{L-1})\left(\frac{1}{T_n(s_i^{L-1},a)}\sum_{h=1}^{T_n(s_i^{L-1},a)} R_h(s_i^{L-1},a) + \gamma \sum_{s_j^L} P(s_j^L|s_i^{L-1},a)Y_n(s_j^L)\right)$$
$$= \sum_a \pi(a|s_i^{L-1})\left(\widehat{\mu}(s_i^{L-1},a) + \gamma \sum_{s_j^L} P(s_j^L|s_i^{L-1},a)Y_n(s_j^L)\right).$$

Then for the estimator $Y_n(s_i^{L-1})$ we can show that its expectation is given as follows:

$$\mathbb{E}[Y_n(s_i^{L-1})] = \sum_a \pi(a|s_i^{L-1}) \Bigg[\frac{1}{T_n(s_i^{L-1}, a)} \sum_{h=1}^{T_n(s_i^{L-1},a)} \mathbb{E}[R_h(s_i^{L-1}, a)] + \gamma \sum_{s_j^L} P(s_j^L|s_i^{L-1}, a)\mathbb{E}[Y_n(s_j^L)] \Bigg]$$

$$= \sum_a \pi(a|s_i^{L-1}) \Bigg[\mu(s_i^{L-1}, a) + \gamma \sum_{s_j^L} P(s_j^L|s_i^{L-1}, a) v_n^\pi(Y(s_j^L)) \Bigg] = v_n^\pi(s_i^{L-1}).$$

$$\mathrm{Var}[Y_n(s_i^{L-1})] = \sum_a \pi^2(a|s_i^{L-1}) \Bigg[\frac{1}{T_n^2(s_i^{L-1}, a)} \sum_{h=1}^{T_n(s_i^{L-1},a)} \mathrm{Var}[R_h(s_i^{L-1}, a)] + \gamma^2 \sum_{s_j^L} P^2(s_j^L|s_i^{L-1}, a)\mathrm{Var}[Y_n(s_j^L)] \Bigg]$$

$$\overset{(a)}{=} \sum_a \pi^2(a|s_i^{L-1}) \Bigg[\frac{\sigma^2(s_i^{L-1}, a)}{T_n(s_i^{L-1}, a)} + \gamma^2 \sum_{s_j^L} P^2(s_j^L|s_i^{L-1}, a)\mathrm{Var}[Y_n(s_j^L)] \Bigg]$$

where, (a) follows as $\sum_{h=1}^{T_n(s_i^{L-1},a)} \mathrm{Var}[R_h(s_i^{L-1}, a)] = T_n(s_i^{L-1}, a)\sigma^2(s_i^{L-1}, a)$. Observe that in state s_i^{L-1} we want to reduce the variance $\mathrm{Var}[Y_n(s_i^{L-1})]$. Also the optimal proportion $b^*(a|s_i^{L-1})$ to reduce variance at state s_i^{L-1} cannot differ from the optimal $b^*(a|s_i^L)$ of level L which reduces the variance of $b^*(a|s_i^L)$. Hence, we can follow the same optimization as done in Theorem 2.2 and show that the optimal sampling proportion in state s_i^{L-1} is given by

$$b^*(a|s_j^L) \propto \pi(a|s_j^L)\sigma(s_j^L, a)$$

$$b^*(a|s_i^{L-1}) \overset{(a)}{\propto} \sqrt{\pi^2(a|s_i^{L-1}) \Bigg[\sigma^2(s_i^{L-1}, a) + \gamma^2 \sum_{s_j^L} P(s_j^L|s_i^{L-1}, a) B^2_{s_j^L} \Bigg]}$$

where, in (a) the s_j^L is the state that follows after taking action a at state s_i^{L-1} and $B_{s_j^L}$ is defined in (2.4). This concludes the base case of the induction proof. Now we will go to the induction step.

Step 2 (Induction step for Arbitrary Level ℓ): We will assume that all the sampling proportion till level $\ell+1$ from L which is

$$b^*(a|s_i^{\ell+1}) \propto \sqrt{\pi^2(a|s_i^\ell)\left[\sigma^2(s_i^\ell,a)+\gamma^2\sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell,a)B^2_{s_j^{\ell+1}}\right]}$$

is true. For the arbitrary level $\ell+1$ we will use dynamic programming. We build up from the leaves (states s_i^L) up to estimate $b^*(a|s_i^{\ell+1})$. Then we need to show that at the previous level ℓ we get a similar recursive sampling proportion. We first define the estimate of the return from an arbitrary state s_i^ℓ in level ℓ after n timesteps as follows:

$$Y_n(s_i^\ell) = \sum_a \pi(a|s_i^\ell)\left(\frac{1}{T_n(s_i^\ell,a)}\sum_{h=1}^{T_n(s_i^\ell,a)} R_h(s_i^\ell,a)+\gamma\sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell,a)Y_n(s_j^{\ell+1})\right)$$

Then we have the expectation of $Y_n(s_i^\ell)$ as follows:

$$\mathbb{E}[Y_n(s_i^\ell)] \overset{(a)}{=} \sum_a \pi(a|s_i^\ell)\left(\mu(s_i^\ell,a)+\gamma\sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell,a)v_n^\pi(Y_n(s_j^{\ell+1}))\right)$$

where, in (a) the $v_n^\pi(Y(s_j^{\ell+1})) = \mathbb{E}[Y_n(s_j^{\ell+1})]$. Then we can also calculate the variance of $Y_n(s_i^\ell)$ as follows:

$$\mathrm{Var}[Y_n(s_i^\ell)] = \sum_a \pi^2(a|s_i^\ell)\left[\frac{\sigma^2(s_i^\ell,a)}{T_n(s_i^\ell,a)}+\gamma^2\sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell,a)\mathrm{Var}(Y_n(s_j^{\ell+1}))\right].$$

Again observe that the goal is to minimize the variance $\mathrm{Var}[Y_n(s_i^\ell)]$. Then

following the same steps in Theorem 2.2 we can have the optimization problem to reduce the variance which results in the following optimal sampling proportion:

$$b^*(a|s_i^\ell) \propto \sqrt{\pi^2(a|s_i^\ell)\left[\sigma^2(s_i^\ell, a) + \gamma^2 \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) B^2(s_j^{\ell+1})\right]}$$

where in the last equation we use $B_{s_j^{\ell+1}}$ which is defined in (2.4). Again we can apply Theorem 2.2 because the optimal proportion $b^*(a|s_i^\ell)$ to reduce variance at state s_i^ℓ cannot differ from the optimal $b^*(a|s_i^{\ell+1})$ of level $\ell+1$ to L which reduces the variance of $b^*(a|s_j^{\ell+1})$ to $b^*(a|s_m^L)$.

Step 3 (Starting state s_1^1:) Finally we conclude by stating that the starting state s_1^1 we have the estimate of the return as follows:

$$Y_n(s_1^1) = \sum_a \pi(a|s_1^1)\left(\frac{1}{T_L^K(s_1^1, a)} \sum_{h=1}^{T_n(s_1^1, a)} R_h(s_1^1, a) + \gamma \sum_{s_j^2} P(s_j^2|s_1^1, a) Y_n(s_j^2)\right).$$

Then we have the expectation of $Y_n(s_1^1)$ as follows:

$$\mathbb{E}[Y_n(s_1^1)] \overset{(a)}{=} \sum_a \pi(a|s_1^1)\left(\mu(s_1^1, a) + \gamma \sum_{s_j^2} P(s_j^2|s_1^1, a) v_n^\pi(Y_n(s_j^2))\right)$$

where, in (a) the $v_n^\pi(s_j^1) = \mathbb{E}[Y_n(s_1^1)]$. Then we can also calculate the variance of $Y_n(s_1^1)$ as follows:

$$\mathrm{Var}[Y_n(s_1^1)] = \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1, a)}{T_n(s_1^1, a)} + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1, a)\mathrm{Var}[Y_n(s_j^2)]\right]$$

Then from the previous step 2 we can show that to reduce the variance

$\mathrm{Var}[Y_n(s_1^1)]$ we should have the sampling proportion at s_1^1 as follows:

$$b^*(a|s_1^1) \propto \sqrt{\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a) + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1,a)B^2(s_j^2)\right]}$$

where, in (a) the s_j^2 is the state that follows after taking action a at state s_1^1, and $B_{s_j^1}$ is defined in (2.4). □

A.6 MSE of the Oracle in Tree MDP

Proposition 2. **(*Restatement*)** *Let there be an oracle which knows the state-action variances and transition probabilities of the* L*-depth tree MDP* $\mathbf{T}$*. Let the oracle take actions in the proportions given by Theorem 1. Let* $\mathcal{D}$ *be the observed data over* n *state-action-reward samples such that* $n = KL$*. Then the oracle suffers a MSE of*

$$\mathcal{L}_n^*(b) = \sum_{\ell=1}^{L}\left[\frac{B^2(s_i^\ell)}{T_L^{*,K}(s_i^\ell)} + \gamma^2 \sum_a \pi^2(a|s_i^\ell) \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell,a)\frac{B^2(s_j^{\ell+1})}{T_L^{*,K}(s_j^{\ell+1})}\right].$$

where, $T_L^{*,K}(s_i^\ell)$ *denotes the optimal state samples of the oracle at the end of episode* K*.*

Proof. **Step 1 (Arbitrary episode k):** First we start at an arbitrary episode k. For brevity we drop the index k in our notation in this step. Let n' be the total number of samples collected up to the k-th episode. We define the estimate of the return from starting state after total of n' samples as

$$Y_{n'}(s_1^1) = \sum_a \pi(a|s_1^1)\left(\frac{1}{T_{n'}(s_1^1,a)}\sum_{h=1}^{T_{n'}(s_1^1,a)} R_h(s_1^1,a) + \gamma \sum_{s_j^2} P(s_j^2|s_1^1,a)Y_{n'}(s_j^2)\right).$$

Then we define the MSE as

$$\mathbb{E}_{\mathcal{D}}\left[\left(Y_{n'}(s_1^1)-\mu(Y_{n'}(s_1^1))\right)^2\right]=\mathrm{Var}(Y_{n'}(s_1^1))+\mathrm{bias}^2(Y_{n'}(s_1^1)).$$

Again it can be shown using Theorem 1 that once all the state-action pairs are visited once we have the bias to be zero. So we want to reduce the variance $\mathrm{Var}(Y_{n'}(s_1^1))$. Note that the variance is given by

$$\mathrm{Var}[Y_{n'}(s_1^1)]=\sum_a \pi^2(a|s_1^1)\Bigg[\underbrace{\frac{\sigma^2(s_1^1,a)}{T_{n'}(s_1^1,a)}}_{\textbf{Variance of } s_1^1}+\gamma^2\sum_{s_j^2}P(s_j^2|s_1^1,a)\underbrace{\mathrm{Var}[Y_{n'}(s_j^2)]}_{\textbf{Variance of } s_j^2 \textbf{ in level 2}}\Bigg]. \tag{A.14}$$

Then we can show from the result of Theorem 1 that to minimize the $\mathrm{Var}[Y_{n'}(s_1^1)]$ the optimal sampling proportion for the level 0 is given by:

$$b^*(a|s_1^1)=\frac{\sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a)+\gamma^2P(s_j^2|s_1^1,a)B^2_{s_j^2}\right]}}{B(s_1^1)}$$

where, s_j^2 are the next states of the state s_1^1, and $B_{s_1^1}$ as defined in (2.4). Let the optimal number of samples of the state-action pair (s_i^ℓ,a) that an oracle can take in the k-th episode be denoted by $T_L^{*,K}(s_i^\ell,a)$. Also let the total number of samples taken in state s_1^1 be $T_L^{*,K}(s_1^1)$. It follows then $n'=\sum_{s_j^\ell\in\mathcal{S}}T_{n'}^{*,k}(s_j^\ell)$. Then we have

$$T_{n'}^{*,k}(s_1^1,a)=\frac{\sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a)+\gamma^2P(s_j^2|s_1^1,a)B^2_{s_j^2}\right]}}{B_{s_1^1}}T_{n'}^k(s_1^1).$$

where we define the normalization factor $B_{s_j^\ell}$ as in (2.4) and $T_{n'}^k(s_1^1)$ is the actual total number of times the state s_1^1 is visited. Plugging this back in

(A.14) we get that

$$
\begin{aligned}
\mathrm{Var}[Y_{n'}(s_1^1)] &= \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{T_{n'}^{*,k}(s_1^1,a)} + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_{n'}(s_j^2)]\right] \\
&= \frac{B(s_1^1)}{T_{n'}^{*,k}(s_1^1)} \sum_a \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{\sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a)+\gamma^2 P^2(s_j^2|s_1^1,a)B^2_{s_j^2}\right]}} + \gamma^2 \sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_{n'}(s_j^2)] \\
&\overset{(a)}{\leqslant} \frac{B(s_1^1)}{T_{n'}^{*,k}(s_1^1)} \sum_a \frac{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a)+\gamma^2 P^2(s_j^2|s_1^1,a)B^2_{s_j^2}\right]}{\sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a)+\gamma^2 P(s_j^2|s_1^1,a)B^2_{s_j^2}\right]}} + \gamma^2 \sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_{n'}(s_j^2)] \\
&= \frac{B_{s_1^1}}{T_{n'}^{*,k}(s_1^1)} \sum_a \sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\sigma^2(s_1^1,a)+\gamma^2 P(s_j^2|s_1^1,a)B^2_{s_j^2}\right]} + \gamma^2 \sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_{n'}(s_j^2)] \\
&\overset{(b)}{=} \frac{B^2_{s_1^1}}{T_{n'}^{*,k}(s_1^1)} + \gamma^2 \sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\underbrace{\sum_{a'}\pi^2(a'|s_j^2)\left[\frac{\sigma^2(s_j^2,a')}{T_{n'}^{k}(s_j^2,a')} + \gamma^2\sum_{s_m^3} P(s_m^3|s_j^2,a')\mathrm{Var}[Y_{n'}(s_m^3)]\right]}_{\mathrm{Var}[Y_{n'}(s_j^2)]} \\
&\overset{(c)}{\leqslant} \frac{B^2_{s_1^1}}{T_{n'}^{*,k}(s_1^1)} + \gamma^2 \sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\frac{B^2_{s_j^2}}{T_{n'}^{*,k}(s_j^2)} \\
&\quad + \gamma^4 \sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\sum_{a'}\pi^2(a'|s_j^2)\sum_{s_m^3} P(s_m^3|s_j^2,a')\mathrm{Var}[Y_{n'}(s_m^3)] \\
&\overset{(d)}{\leqslant} \sum_{\ell=1}^{L}\left[\frac{B^2(s_i^\ell)}{T_{n'}^{*,k}(s_i^\ell)} + \gamma^{2\ell}\sum_a \pi^2(a|s_i^\ell)\sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell,a)\frac{B^2(s_j^{\ell+1})}{T_{n'}^{*,k}(s_j^{\ell+1})}\right]
\end{aligned}
$$

where, (a) follows as $\gamma^2 B^2_{s_j^1} \geqslant 0$, (b) follows by the definition of $\mathrm{Var}[Y_{s_j^2}]$ and the definition of $B(s_1^1)$ and $T_{n'}^k(s_j^2)$ is the actual number of samples observed for s_j^2, (c) follows by substituting the value of $T_{n'}^{*,k}(s_j^2,a') = b^*(a'|s_j^2)/B(s_j^2)$, and (d) follows when unrolling the equation for L times.

Step 2 (End of K episodes): Note that the above derivation holds for an arbitrary episode k which consist of L step horizon from root to leaf. Hence the MSE of the oracle after K episodes when running behavior

policy b is given as

$$\mathcal{L}_n^*(b) = \sum_{\ell=1}^{L} \left[\frac{B^2(s_i^\ell)}{T_n^{*,K}(s_i^\ell)} + \gamma^{2\ell} \sum_a \pi^2(a|s_i^\ell) \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) \frac{B^2(s_j^{\ell+1})}{T_n^{*,K}(s_j^{\ell+1})} \right]$$

Note that $n = \sum_a \sum_{s_i^\ell \in \mathcal{S}} T_n^{*,K}(s_i^\ell, a)$ is the total samples collected after K episodes of L trajectories. This gives the MSE following optimal proportion in Theorem 1.

□

A.7 Support Lemmas

Lemma A.2. (***Wald's lemma for variance***) (*Resnick, 2019*) *Let $\{\mathcal{F}_t\}$ be a filtration and R_t be a $\mathcal{F}_t$-adapted sequence of i.i.d. random variables with variance σ^2. Assume that $\mathcal{F}_t$ and the σ-algebra generated by $\{R_{t'} : t' \geqslant t+1\}$ are independent and T is a stopping time w.r.t. $\mathcal{F}_t$ with a finite expected value. If $\mathbb{E}\left[R_1^2\right] < \infty$ then*

$$\mathbb{E}\left[\left(\sum_{t'=1}^{T} R_{t'} - T\mu\right)^2\right] = \mathbb{E}[T]\sigma^2$$

Lemma A.3. (***Hoeffding's Lemma***)(*Massart, 2007*) *Let Y be a real-valued random variable with expected value $\mathbb{E}[Y] = \mu$, such that $a \leqslant Y \leqslant b$ with probability one. Then, for all $\lambda \in \mathbb{R}$*

$$\mathbb{E}\left[e^{\lambda Y}\right] \leqslant \exp\left(\lambda\mu + \frac{\lambda^2(b-a)^2}{8}\right)$$

Lemma A.4. (***Concentration lemma 1***) *Let $V_t = R_t(s,a) - \mathbb{E}[R_t(s,a)]$ and be bounded such that $V_t \in [-\eta, \eta]$. Let the total number of times the state-action*

(s, a) *is sampled be* T. *Then we can show that for an* $\epsilon > 0$

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T} R_t(s,a) - \mathbb{E}[R_t(s,a)]\right| \geqslant \epsilon\right) \leqslant 2\exp\left(-\frac{2\epsilon^2 T}{\eta^2}\right).$$

Proof. Let $V_t = R_t(s,a) - \mathbb{E}[R_t(s,a)]$. Note that $\mathbb{E}[V_t] = 0$. Hence, for the bounded random variable $V_t \in [-\eta, \eta]$ (by Assumption 2) we can show from Hoeffding's lemma in Theorem A.3 that

$$\mathbb{E}[\exp(\lambda V_t)] \leqslant \exp\left(\frac{\lambda^2}{8}(\eta - (-\eta))^2\right) \leqslant \exp\left(2\lambda^4\eta^2\right)$$

Let s_{t-1} denote the last time the state s is visited and action a is sampled. Observe that the reward $R_t(s,a)$ is conditionally independent. For this proof we will only use the boundedness property of $R_t(s,a)$ guaranteed by

Assumption 2. Next we can bound the probability of deviation as follows:

$$
\begin{aligned}
&\mathbb{P}\left(\sum_{t=1}^{T}\left(R_t(s,a)-\mathbb{E}[R_t(s,a)]\right)\geqslant\epsilon\right)\\
&=\mathbb{P}\left(\sum_{t=1}^{T}V_t\geqslant\epsilon\right)\\
&\overset{(a)}{=}\mathbb{P}\left(e^{\lambda\sum_{t=1}^{T}V_t}\geqslant e^{\lambda\epsilon}\right)\\
&\overset{(b)}{\leqslant}e^{-\lambda\epsilon}\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T}V_t}\right]\\
&=e^{-\lambda\epsilon}\mathbb{E}\left[\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T}V_t}\middle|s_{T-1}\right]\right]\\
&\overset{(c)}{=}e^{-\lambda\epsilon}\mathbb{E}\left[\mathbb{E}\left[e^{-\lambda V_T}|S_{T-1}\right]\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T-1}V_t}\middle|s_{T-1}\right]\right]\\
&\leqslant e^{-\lambda\epsilon}\mathbb{E}\left[\exp\left(2\lambda^4\eta^2\right)\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T-1}V_t}\middle|s_{T-1}\right]\right]\\
&=e^{-\lambda\epsilon}e^{2\lambda^2\eta^2}\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T-1}V_t}\right]\\
&\vdots\\
&\overset{(d)}{\leqslant}e^{-\lambda\epsilon}e^{2\lambda^2T\eta^2}\\
&\overset{(e)}{\leqslant}\exp\left(-\frac{2\epsilon^2}{T\eta^2}\right)
\end{aligned}
\tag{A.15}
$$

where (a) follows by introducing $\lambda\in\mathbb{R}$ and exponentiating both sides, (b) follows by Markov's inequality, (c) follows as V_t is conditionally independent given s_{T-1}, (d) follows by unpacking the term for T times and (e) follows by taking $\lambda=\epsilon/4T\eta^2$. Hence, it follows that

$$
\begin{aligned}
\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T}R_t(s,a)-\mathbb{E}[R_t(s,a)]\right|\geqslant\epsilon\right)&=\mathbb{P}\left(\sum_{t=1}^{T}\left(R_t(s,a)-\mathbb{E}[R_t(s,a)]\right)\geqslant T\epsilon\right)\\
&\overset{(a)}{\leqslant}2\exp\left(-\frac{2\epsilon^2T}{\eta^2}\right).
\end{aligned}
$$

where, (a) follows by (A.15) by replacing ϵ with ϵT, and accounting for deviations in either direction. □

Lemma A.5. (***Concentration lemma 2***) *Let* $\mu^2(s,a) = \mathbb{E}\left[R_t^2(s,a)\right]$. *Let* $R_t(s,a) \leqslant 2\eta$ *and* $R_t^2(s,a) \leqslant 4\eta^2$ *for any time* t *and following Assumption 2. Let* $n = KL$ *be the total budget of state-action samples. Let*

$$C_n(\eta,\delta) = (2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_n(s,a)}}.$$

Define the event

$$\xi_\delta = \left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t^2(s,a) - \mu^2(s,a)\right| \leqslant C_n(\eta,\delta)\right\}\right)\bigcap \left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t(s,a) - \mu(s,a)\right| \leqslant C_n(\eta,\delta)\right\}\right) \tag{A.16}$$

Then we can show that $\mathbb{P}(\xi_\delta) \geqslant 1 - 2\delta$.

Proof. First note that the total budget $n = KL$. Observe that the random variable $R_t^k(s,a)$ and $R_t^{(2),k}(s,a)$ are conditionally independent given the previous state S_{t-1}^k. Also observe that for any $\eta > 0$ we have that $R_t^k(s,a), R_t^{(2),k}(s,a) \leqslant 2\eta + 4\eta^2$, where $R_t^{(2),k}(s,a) = (R_t^k(s,a))^2$. Hence we

can show that

$$\mathbb{P}\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t^2(s,a)-\mu^2(s,a)\right|\geqslant C_n(\eta,\delta)\right\}\right)$$
$$\leqslant \mathbb{P}\left(\bigcup_{s\in\mathcal{S}}\bigcup_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t^2(s,a)-\mu^2(s,a)\right|\geqslant C_n(\eta,\delta)\right\}\right)$$
$$\overset{(a)}{\leqslant}\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{t=1}^{n}\sum_{T_n(s,a)=1}^{t} 2\exp\left(-\frac{2T_n}{4(\eta^2+\eta)^2}\cdot\frac{4(\eta^2+\eta)^2\log(SAn(n+1)/\delta)}{2T_n(s,a)}\right)=\delta.$$

where, (a) follows from Theorem A.4. Note that in (a) we have to take a double union bound summing up over all possible pulls T_n from 1 to n as T_n is a random variable. Similarly we can show that

$$\mathbb{P}\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t(s,a)-\mu(s,a)\right|\geqslant C_n(\eta,\delta)\right\}\right)$$
$$\overset{(a)}{\leqslant}\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{t=1}^{n}\sum_{T_n(s,a)=1}^{t} 2\exp\left(-\frac{2T_n}{4(\eta^2+\eta)^2}\cdot\frac{4(\eta^2+\eta)^2\log(SAn(n+1)/\delta)}{2T_n(s,a)}\right)=\delta.$$

where, (a) follows from Theorem A.4. Hence, combining the two events above we have the following bound

$$\mathbb{P}(\xi_\delta)\geqslant 1-2\delta.$$

□

Corollary A.6. *Under the event ξ_δ in (A.16) we have for any state-action pair in an episode k the following relation with probability greater than $1-\delta$*

$$|\widehat{\sigma}_t^k(s,a)-\sigma(s,a)|\leqslant(2\eta+4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_L^K(s,a)}}.$$

where, $T_L^K(s,a)$ *is the total number of samples of the state-action pair* (s,a) *till episode* k*.*

Proof. Observe that the event ξ_δ bounds the sum of rewards $R_t^k(s,a)$ and squared rewards $R_t^{k,(2)}(s,a)$ for any $T_L^K(s,a) \geqslant 1$. Hence we can directly apply the Theorem A.5 to get the bound. □

Lemma A.7. **(*Bound samples in level* 2)** *Suppose that, at an episode* k*, the action* p *in state* s_i^2 *in a* 2*-depth* **T** *is under-pulled relative to its optimal proportion. Then we can lower bound the actual samples* $T_L^K(s_i^2,p)$ *with respect to the optimal samples* $T_L^{*,K}(s_i^2,p)$ *with probability* $1-\delta$ *as follows*

$$T_L^K(s_i^2,p) \geqslant T_L^{*,K}(s_i^2,p) - 4cb^*(p|s_i^2)\frac{\sqrt{\log(H/\delta)}}{B(s_i^2)b_{min}^{*,3/2}(s_i^2)}\sqrt{T_L^K(s_i^2)} - 4Ab^*(p|s_i^2),$$

where $B(s_i^2)$ *is defined in* (2.4)*,* $c = (\eta+\eta^2)/\sqrt{2}$*, and* $H = SAn(n+1)$*.*

Proof. **Step 1 (Properties of the algorithm):** Let us first define the confidence interval term for (s,a) at time t as

$$U_t^k(s,a) = 2c\sqrt{\frac{\log(H/\delta)}{T_t^k(s_i^2,a)}} \tag{A.17}$$

where, $c = (\eta+\eta^2)/\sqrt{2}$, and $H = SAn(n+1)$. Also note that on ξ_δ using Theorem A.6 we have

$$\begin{aligned}
\widehat{\sigma}_t^k(s_i^2,a) &\overset{(a)}{\leqslant} \sigma(s_i^2,a) + U_t^k(s,a) \\
&\implies \widehat{\sigma}_t^{(2),k}(s_i^2,a) \leqslant \sigma^2(s_i^2,a) + 2\sigma(s_i^2,a)U_t^k(s,a) + U_t^{(2),k}(s,a) \\
&= \sigma^2(s_i^2,a) + 4\sigma c\sqrt{\frac{\log(H/\delta)}{T_t^k(s_i^2,a)}} + 4c^2\frac{\log(H/\delta)}{T_t^k(s_i^2,a)} \\
&\overset{(b)}{\leqslant} \sigma^2(s_i^2,a) + 4dc^2\sqrt{\frac{\log(H/\delta)}{T_t^k(s_i^2,a)}}
\end{aligned} \tag{A.18}$$

where, (a) follows from Theorem A.6, and (b) follows for some constant $d > 0$ and noting that $\sqrt{\frac{\log(H/\delta)}{T_t^k(s_i^2,a)}} > \frac{\log(H/\delta)}{T_t^k(s_i^2,a)}$ and $c^2 > c$. Let a be an arbitrary action in state s_i^2. Recall the definition of the upper bound used in ReVar when $t > 2SA$:

$$\begin{aligned}\overline{U}_{t+1}^k(a|s_i^2) &= \frac{\widehat{b}_t^k(a|s_i^2)}{T_t^k(s_i^2,a)} = \frac{\sqrt{\pi^2(a|s_i^2)\widehat{\sigma u}_t^{(2),k}(s_i^2,a)}}{T_t^k(s_i^2,a)} \\ &= \frac{\sqrt{\pi^2(a|s_i^2)\left(\widehat{\sigma}_t^{(2),k}(s_i^2,a) + 4dc^2\sqrt{\frac{\log(H/\delta)}{T_t^k(s_i^2,a)}}\right)}}{T_t^k(s_i^2,a)}\end{aligned}$$

Under the good event ξ_δ using Theorem A.6, we obtain the following upper and lower bounds for $\overline{U}_{t+1}^k(a|s_i^2)$:

$$\frac{\sqrt{\pi^2(a|s_i^2)\sigma^2(s_i^2,a)}}{T_t^k(s_i^2,a)} \overset{(a)}{\leqslant} \overline{U}_{t+1}^k(a|s_i^2) \overset{(b)}{\leqslant} \frac{\sqrt{\pi^2(a|s_i^2)\left(\sigma^2(s_i^2,a) + 8dc^2\sqrt{\frac{\log(H/\delta)}{T_t^k(s_i^2,a)}}\right)}}{T_t^k(s_i^2,a)} \tag{A.19}$$

where, (a) follows as $\sigma^2(s_i^2,a) \leqslant \widehat{\sigma}_t^{(2),k}(s_i^2,a) + 4dc^2\sqrt{\log(H/\delta)/T_k^t(s_i^2,a)}$ and (b) follows as $\widehat{\sigma}_t^{(2),k}(s_i^2,a) + 4dc^2\sqrt{\log(H/\delta)/T_k^t(s_i^2,a)} \leqslant \widehat{\sigma}_t^{(2),k}(s_i^2,a) + 8dc^2\sqrt{\log(H/\delta)/T_k^t(s_i^2,a)}$. Let ReVar chooses to pull action m at $t+1 > 2SA$ in s_i^L for the last time. Then we have that for any action $p \neq m$ the following:

$$\overline{U}_{t+1}^k(p|s_i^2) \leqslant \overline{U}_{t+1}^k(m|s_i^2).$$

Recall that $T_t^k(s_i^2,m)$ is the last time the action m is sampled. Hence, $T_t^k(s_i^2,m) = T_L^K(s_i^2,m) - 1$ because we are sampling action m again in time $t+1$. Note that $T_L^K(s_i^2,m)$ is the total pulls of action m at the end of time

n. It follows from (A.19) then

$$\overline{U}^k_{t+1}(m|s^2_i) \leqslant \frac{\sqrt{\pi^2(m|s^2_i)\left(\sigma^2(s^2_i,m)+8dc^2\sqrt{\frac{\log(H/\delta)}{T^k_t(s^2_i,m)}}\right)}}{T^k_t(s^2_i,m)} = \frac{\sqrt{\pi^2(m|s^2_i)\left(\sigma^2(s^2_i,m)+8dc^2\sqrt{\frac{\log(H/\delta)}{T^k_t(s^2_i,m)-1}}\right)}}{T^k_t(s^2_i,m)-1}.$$

Let p be the arm in state s^2_i that is under-pulled. Recall that $T^K_L(s^2_i) = \sum_a T^K_L(s^2_i,a)$. Using the lower bound in (A.19) and the fact that $T^k_t(s^2_i,p) \leqslant T^K_L(s^2_i,p)$, we may lower bound $I^k_{t+1}(p|s^2_i)$ as

$$\overline{U}^k_{t+1}(p|s^2_i) \geqslant \frac{\sqrt{\pi^2(p|s^2_i)\sigma^2(s^2_i,p)}}{T^k_t(s^2_i,p)} \geqslant \frac{\sqrt{\pi^2(p|s^2_i)\sigma^2(s^2_i,p)}}{T^K_L(s^2_i,p)}.$$

Combining all of the above we can show

$$\frac{\sqrt{\pi^2(p|s^2_i)\sigma^2(s^2_i,p)}}{T^K_L(s^2_i,p)} \leqslant \frac{\sqrt{\pi^2(m|s^2_i)\left(\sigma^2(s^2_i,m)+8dc^2\sqrt{\frac{\log(H/\delta)}{T^K_L(s^2_i,m)-1}}\right)}}{T^K_L(s^2_i,m)-1}. \tag{A.20}$$

Observe that there is no dependency on t, and thus, the probability that (A.20) holds for any p and for any m is at least $1-\delta$ (probability of event ξ_δ).

Step 2 (Lower bound on $T^K_L(s^2_i,p)$): If an action p is under-pulled compared to its optimal allocation without taking into account the initialization phase,i.e., $T^K_L(s^2_i,p)-2 < b(p|s^2_i)(T_n(s^2_i)-2A)$, then from the constraint $\sum_a \left(T^K_L(s^2_i,a)-2\right) = T^K_L(s^2_i)-2A$ and the definition of the optimal allocation, we deduce that there exists at least another action m that is over-pulled compared to its optimal allocation without taking into account

the initialization phase, i.e., $T_n^k(s_i^2, m) - 2 > b(m|s_i^2)(T_L^K(s_i^2) - 2SA)$.

$$\begin{aligned}
\frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{T_L^K(s_i^2,p)} &\leqslant \frac{\sqrt{\pi^2(m|s_i^2)\left(\sigma^2(s_i^2,m) + 8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_i^2,m)-1}}\right)}}{T_L^K(s_i^2,m)-1} \\
&\overset{(a)}{\leqslant} \frac{\sqrt{\pi^2(m|s_i^2)\left(\sigma^2(s_i^2,m) + 8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_i^2,m)-2}}\right)}}{T_L^K(s_i^2,m)-1} \\
&\overset{(b)}{\leqslant} \frac{\sqrt{\pi^2(m|s_i^2)\sigma^2(s_i^2,m)} + 4d\pi(m|s_i^2)c\sqrt{\frac{\log(H/\delta)}{T_L^K(s_i^2,m)-2}}}{T_L^{*,K}(s_i^2,m)} \\
&\overset{(c)}{\leqslant} \frac{\sqrt{\pi^2(m|s_i^2)\sigma^2(s_i^2,m)} + \left(4dc\sqrt{\frac{\log(H/\delta)}{b^*(m|s_i^2)(T_L^K(s_i^2)-2SA)+1}}\right)}{T_L^{*,K}(s_i^2,m)} \\
&\overset{(d)}{\leqslant} \frac{B(s_i^2)}{T_L^K(s_i^2)} + 4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_i^2)b^*(m|s_i^2)^{3/2}} + \frac{4AB(s_i^2)}{T_L^{(2),K}(s_i^2)} \\
&\overset{(e)}{\leqslant} \frac{B(s_i^2)}{T_L^K(s_i^2)} + 4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_i^2)b_{min}^{*,3/2}(s_i^2)} + \frac{4AB(s_i^2)}{T_L^{(2),K}(s_i^2)}.
\end{aligned} \tag{A.21}$$

where, (a) follows as $T_L^K(s_i^2,m)-2 \leqslant T_L^K(s_i^2,m)-1$, (b) follows as $T_n^{*,(k)}(s_i^2,m) \geqslant T_L^K(s_i^2,m)-1$ as action m is over-pulled and $\sqrt{a+b} \leqslant \sqrt{a}+\sqrt{b}$ for $a, b > 0$, (c) follows as $T_L^K(s_i^2) = \sum_a T_L^K(s_i^2,a)$ and $T_n^k(s_i^2,m)-2 > b^*(m|s_i^2)(T_L^K(s_i^2)-2SA)$, (d) follows by setting the optimal samples $T_L^{*,K}(s_i^2,m) = \frac{\sqrt{\pi^2(m|s_i^2)\sigma^2(s_i^2,m)}}{B(s_i^2)}T_L^K(s_i^2)$, and (e) follows as $b^*(m|s_i^2) \geqslant b_{min}(s_i^2)$. By rearranging (A.21) , we obtain

the lower bound on $T_L^K(s_i^2, p)$:

$$
\begin{aligned}
T_L^K(s_i^2,p) &\geqslant \frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{\frac{B(s_i^2)}{T_L^K(s_i^2)} + 4dcx\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_i^2)b_{min}^{*,3/2}(s_i^2)} + \frac{4AB(s_i^2)}{T_L^{(2),K}(s_i^2)}} \\
&= \frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{\frac{B(s_i^2)}{T_L^K(s_i^2)}}\left[\frac{1}{1+4dc\frac{\sqrt{\log(H/\delta)}}{B(s_i^2)T_L^{(1/2),K}(s_i^2)b_{min}^{*,3/2}(s_i^2)} + \frac{4A}{T_n^k(s_i^2)}}\right] \\
&\overset{(a)}{\geqslant} \frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{\frac{B(s_i^2)}{T_L^K(s_i^2)}}\left[1-4dc\frac{\sqrt{\log(H/\delta)}}{B(s_i^2)T_L^{(1/2),K}(s_i^2)b_{min}^{*,3/2}(s_i^2)} - \frac{4A}{T_n^k(s_i^2)}\right] \\
&\geqslant T_L^{*,K}(s_i^2,p) - 4dcb^*(p|s_i^2)\frac{\sqrt{\log(H/\delta)}}{B(s_i^2)b_{min}^{*,3/2}(s_i^2)}\sqrt{T_L^K(s_i^2)} - 4Ab^*(p|s_i^2),
\end{aligned}
$$

where in (a) we use $1/(1+x) \geqslant 1-x$ (for $x > -1$). □

Lemma A.8. (***Bound samples in level 1***) *Suppose that, at an episode* k*, the action* p *in state* s_1^1 *in a* 2*-depth* **T** *is under-pulled relative to its optimal proportion. Then we can lower bound the actual samples* $T_L^K(s_1^1, p)$ *with respect to the optimal samples* $T_L^{*,K}(s_1^1, p)$ *with probability* $1-\delta$ *as follows*

$$
\begin{aligned}
T_L^K(s_1^1,p) \geqslant\ & T_L^{*,K}(s_1^1,p) - 4cb^*(p|s_1^1)\frac{\sqrt{\log(H/\delta)}}{B(s_1^1)b_{min}^{*,3/2}(s_1^1)}\sqrt{T_L^K(s_1^1)} - 4Ab^*(p|s_1^1) \\
& - \gamma\pi(m|s_1^1)\frac{T_L^K(s_1^1)}{B^2(s_1^1)}\sum_{s_j^2} P(s_j^2|s_1^1,m)\frac{B(s_j^2)}{b^*(m|s_j^2)}\cdot \\
& \sum_{a'}\left[T_L^{*,K}(s_j^2,a') + 4cb^*(a'|s_j^2)\frac{\sqrt{\log(H/\delta)}}{b_{min}^{*,3/2}(s_j^2)}\sqrt{T_L^K(s_1^1)} + 4Ab(a'|s_j^2)\right]
\end{aligned}
$$

where $B(s_i^2)$ *is defined in* (2.4)*,* $c = (\eta+\eta^2)/\sqrt{2}$*, and* $H = SAn(n+1)$*.*

Proof. **Step 1 (Properties of the algorithm):** Again note that on ξ_δ using

Theorem A.6 we have

$$\widehat{\sigma}_t^k(s_1^1,a) \leqslant \sigma(s_1^1,a) + U_t^k(s,a) \implies \widehat{\sigma}_t^{(2),k}(s_1^1,a) \leqslant \sigma^2(s_1^1,a) + U_t^{(2),k}(s,a)$$
$$\overset{(a)}{=} \sigma^2(s_1^1,a) + 4dc^2\sqrt{\frac{\log(H/\delta)}{T_t^k(s_1^1,a)}}$$

for any action a in s_1^1, where (a) follows by the definition of $U_t^{(2),k}$(A.17), some constant $d > 0$ and the same derivation as in (A.18). Let a be an arbitrary action in state s_1^1. Recall the definition of the upper bound used in ReVar when $t > 2SA$ and define $W(k,t,s,a) = \widehat{\sigma}_t^{(2),k}(s,a) + 4dc^2\sqrt{\frac{\log(H/\delta)}{T_t^k(s,a)}}$:

$$\overline{U}_{t+1}^k(a|s_1^1) = \frac{\widehat{b}_t^k(a|s_1^1)}{T_t^k(s_1^1,a)} = \frac{\sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[\widehat{\sigma u}_t^{(2),k}(s_1^1,a) + \gamma^2 P(s_j^2|s_1^1,a)\widehat{B}_t^{(2),k}(s_j^2)\right]}}{T_t^k(s_1^1,a)}$$
$$= \frac{\sqrt{\sum_{s_j^2}\pi^2(a|s_1^1)\left[W(k,t,s_1^1,a) + \gamma^2 P(s_j^2|s_1^1,a)\sum_{a'}\sqrt{\pi^2(a'|s_j^2)W(k,t,s_j^2,a')}\right]}}{T_t^k(s_1^1,a)}$$

Under the good event ξ_δ using the Theorem A.6, we obtain the following upper and lower bounds for $\overline{U}_{t+1}^k(a|s_1^1)$:

$$\overline{U}_{t+1}^k(a|s_1^1) \leqslant \frac{\sqrt{\sum\limits_{s_j^2}\pi^2(a|s_1^1)\left[2W(k,t,s_1^1,a) + \gamma^2 P(s_j^2|s_1^1,a)\sum\limits_{a'}\sqrt{\pi^2(a'|s_j^2)2W(k,t,s_j^2,a')}\right]}}{T_t^k(s_1^1,a)}$$
$$\overline{U}_{t+1}^k(a|s_1^1) \geqslant \frac{\sqrt{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}}{T_t^k(s_1^1,a)} \tag{A.22}$$

where, (a) follows as $\sigma^2(s_1^1,a) \leqslant \widehat{\sigma}_t^{(2),k}(s_1^1,a) + 4dc^2\sqrt{\log(H/\delta)/T_k^t(s_1^1,a)}$ and (b) follows as $\widehat{\sigma}_t^{(2),k}(s_1^1,a) + 4dc^2\sqrt{\log(H/\delta)/T_k^t(s_1^1,a)} \leqslant \widehat{\sigma}_t^{(2),k}(s_1^1,a) + 8dc^2\sqrt{\log(H/\delta)/T_k^t(s_1^1,a)}$. Let ReVar chooses to take action m at $t+1$ in

s_1^1 for the last time. Then we have that for any action $p \neq m$ the following:

$$\overline{U}_{t+1}^k(p|s_1^1) \leqslant \overline{U}_{t+1}^k(m|s_1^1).$$

Recall that $T_t^k(s_1^1, m)$ is the last time the action m is sampled. Hence, $T_t^k(s_1^1, m) = T_L^K(s_1^1, m) - 1$ because we are sampling action m again in time $t+1$. Note that $T_L^K(s_1^1, m)$ is the total pulls of action m at the end of time n. It follows from (A.22)

$$\begin{aligned}
\overline{U}_{t+1}^k(m|s_1^1) &\leqslant \frac{\sqrt{\sum_{s_j^2} \pi^2(a|s_1^1)\left[2W(k,t,s_1^1,a) + \gamma^2 P(s_j^2|s_1^1,a) \sum_{a'} \sqrt{\pi^2(a'|s_j^2) 2W(k,t,s_j^2,a')}\right]}}{T_t^k(s_1^1,a)} \\
&\overset{(a)}{\leqslant} \frac{\sqrt{\sum_{s_j^2}\left(\pi^2(a|s_1^1)2W(k,t,s_1^1,a)\right)} + \gamma\pi(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\left[\sum_{a'}\sqrt{\pi^2(a'|s_j^2)2W(k,t,s_j^2,a')}\right]}{T_t^k(s_1^1,a)} \\
&\overset{(b)}{\leqslant} \frac{\sqrt{\sum_{s_j^2}\pi^2(m|s_1^1)\left(\sigma^2(s_1^1,m) + 8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_1^1,m)-1}}\right)}}{T_L^K(s_1^1,m)-1} \\
&\quad + \gamma\pi(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\sum_{a'}\left[\frac{\sqrt{\pi^2(a'|s_j^2)\left(\sigma^2(s_j^2,a') + 8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_j^2,a')}}\right)}}{T_L^K(s_j^2,a')-1}\right].
\end{aligned}$$

where, (a) follows as $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$ for $a, b > 0$ and (b) follows as $T_t^k(s_1^1, a) \geqslant T_t^k(s_j^2, a')$ where s_j^2 is the next state of s_1^1 following action a.

Let p be the arm in state s_1^1 that is under-pulled. Recall that $T_L^K(s_1^1) = \sum_a T_L^K(s_1^1, a)$. Using the lower bound in (A.22) and the fact that $T_t^k(s_1^1, p) \leqslant T_L^K(s_1^1, p)$, we may lower bound $\overline{U}_{t+1}^k(p|s_1^1)$ as

$$\overline{U}_{t+1}^k(p|s_1^1) \geqslant \frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{T_t(s_1^1,p)} \geqslant \frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{T_L^K(s_1^1,p)}.$$

Combining all of the above we can show

$$\frac{\sqrt{\pi^2(p|s_i^2)\sigma^2(s_i^2,p)}}{T_L^K(s_1^1,p)} \leqslant \frac{\sqrt{\sum_{s_j^2}\pi^2(m|s_1^1)\left(\sigma^2(s_1^1,m)+8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_1^1,m)-1}}\right)}}{T_L^K(s_1^1,m)-1}$$
$$+\gamma\pi(m|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,m)\sum_{a'}\left[\frac{\sqrt{\pi^2(a'|s_j^2)\left(\sigma^2(s_j^2,a')+8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_j^2,a')-1}}\right)}}{T_L^K(s_j^2,a')-1}\right]. \tag{A.23}$$

Observe that there is no dependency on t, and thus, the probability that (A.23) holds for any p and for any m is at least $1-\delta$ (probability of event ξ_δ).

Step 2 (Lower bound on $T_L^K(s_1^1,p)$): If an action p is under-pulled compared to its optimal allocation without taking into account the initialization phase,i.e., $T_L^K(s_1^1,p)-2 < b^*(p|s_1^1)(T_L^K(s_1^1)-2A)$, then from the constraint $\sum_a\left(T_L^K(s_1^1,a)-2\right)=T_L^K(s_1^1)-2A$ and the definition of the optimal allocation, we deduce that there exists at least another action m that is over-pulled compared to its optimal allocation without taking into account

the initialization phase, i.e., $T_n^k(s_1^1, m) - 2 > b^*(m|s_1^1)(T_L^K(s_1^1) - 2SA)$.

$$
\begin{aligned}
\frac{\pi(p|s_1^1)\sigma(s_1^1,p)}{T_L^K(s_1^1,p)} &\leqslant \frac{\sqrt{\sum_{s_j^2}\pi^2(m|s_1^1)\left(\sigma^2(s_1^1,m)+8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_1^1,m)-2}}\right)}}{T_L^K(s_1^1,m)-1} \\
&\quad + \gamma\pi(m|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,m)\sum_{a'}\left[\frac{\sqrt{\pi^2(a'|s_j^2)\left(\sigma^2(s_j^2,a')+8dc^2\sqrt{\frac{\log(H/\delta)}{T_L^K(s_j^2,a')-2}}\right)}}{T_L^K(s_j^2,a')-1}\right] \\
&\overset{(a)}{\leqslant} \sum_{s_j^2}\frac{\sqrt{\pi^2(m|s_1^1)\sigma^2(s_1^1,m)}+4dc\sqrt{\frac{\log(H/\delta)}{T_L^K(s_1^1,m)-2}}}{T_L^{*,K}(s_1^1,m)} \\
&\quad + \gamma\pi(m|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,m)\sum_{a'}\left[\frac{\sqrt{\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}+4dc\sqrt{\frac{\log(H/\delta)}{T_L^K(s_j^2,a')-2}}}{T_L^{*,K}(s_j^2,a')}\right] \\
&\overset{(b)}{\leqslant} \sum_{s_j^2}\frac{\sqrt{\pi^2(m|s_1^1)\sigma^2(s_1^1,m)}+\left(4dc\sqrt{\frac{\log(H/\delta)}{b^*(m|s_1^1)(T_L^K(s_1^1)-2SA)+1}}\right)}{T_L^{*,K}(s_1^1,m)} \\
&\quad + \gamma\pi(m|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,m)\sum_{a'}\left[\frac{\sqrt{\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}+4dc\sqrt{\frac{\log(H/\delta)}{b^*(a'|s_j^2)(T_L^K(s_j^2)-2SA)+1}}}{T_L^{*,K}(s_j^2,a')}\right] \\
&\overset{(c)}{\leqslant} \sum_{s_j^2}\left[\frac{B(s_1^1)}{T_L^K(s_1^1)}+4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_1^1)b^*(m|s_1^1)^{3/2}}+\frac{4AB(s_1^1)}{T_L^{(2),K}(s_1^1)}\right] \\
&\quad + \gamma\pi(m|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,m)\underbrace{\sum_{a'}\left[\frac{B(s_j^2)}{T_L^K(s_j^2)}+4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2)b_{min}^{*,3/2}(s_j^2)}+\frac{4AB(s_j^2)}{T_L^{(2),K}(s_j^2)-1}\right]}_{\mathbb{V}(s_j^2)} \\
&\overset{(d)}{\leqslant} \sum_{s_j^2}\left[\frac{B(s_1^1)}{T_L^K(s_1^1)}+4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_1^1)b_{min}^{*,3/2}(s_1^1)}+\frac{4AB(s_1^1)}{T_L^{(2),K}(s_1^1)-1}\right]+\gamma\pi(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)\mathbb{V}(s_j^2)
\end{aligned}
\tag{A.24}
$$

where, (a) follows as $T_n^{*,(k)}(s_1^1, m) \geqslant T_L^K(s_1^1, m) - 1$ as action m is over-pulled, (b) follows as $T_L^K(s_1^1) = \sum_a T_L^K(s_1^1, a)$ and $T_n^k(s_1^1, m) - 2 > b^*(m|s_1^1)(T_L^K(s_1^1) - 2SA)$ and a similar argument follows in state s_j^2, (c) follows $T_L^{*,K}(s_1^1, m) = \frac{\sqrt{\pi^2(m|s_1^1)\sigma^2(s_1^1,m)}}{B(s_1^1)}T_L^K(s_1^1)$, and using the result of theorem A.7. Finally, (d) follows as $b^*(m|s_1^1) \geqslant b_{min}(s_1^1)$. In (d) we also define the total over samples

in state s_j^2 as $\mathbb{V}(s_j^2)$ such that

$$\mathbb{V}(s_j^2) \coloneqq \sum_{a'} \left[\frac{B(s_j^2)}{T_L^K(s_j^2)} + 4dc \frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2) b_{min}^{*,3/2}(s_j^2)} + \frac{4AB(s_j^2)}{T_L^{(2),K}(s_j^2) - 1} \right]$$

By rearranging (A.24) , we obtain the lower bound on $T_L^K(s_1^1, p)$:

$$T_L^K(s_1^1,p) \geqslant \frac{\sqrt{\pi^2(p|s_1^1)\sigma^2(s_1^1,p)}}{\frac{B(s_1^1)}{T_L^K(s_1^1)} + 4dc \frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_1^1) b_{min}^{*,3/2}(s_1^1)} + \frac{4AB(s_1^1)}{T_L^{(2),K}(s_1^1)} + \gamma\pi(m|s_1^1) \sum_{s_j^2} P(s_j^2|s_1^1,m)\mathbb{V}(s_j^2)}$$

$$= \frac{\sqrt{\pi^2(p|s_1^1)\sigma^2(s_1^1,p)}}{\frac{B(s_1^1)}{T_L^K(s_1^1)}} \left[\frac{1}{1 + 4dc \frac{\sqrt{\log(H/\delta)}}{B(s_1^1) T_L^{(1/2),K}(s_1^1) b_{min}^{*,3/2}(s_1^1)} + \frac{4A}{T_n^k(s_1^1)} + \gamma\pi(m|s_1^1) \frac{T_L^K(s_1^1)}{B(s_1^1)} \sum_{s_j^2} P(s_j^2|s_1^1,m)\mathbb{V}(s_j^2)} \right]$$

$$\geqslant \frac{\sqrt{\pi^2(p|s_1^1)\sigma^2(s_1^1,p)}}{\frac{B(s_1^1)}{T_L^K(s_1^1)}} \left[\frac{1}{1 + 4dc \frac{\sqrt{\log(H/\delta)}}{B(s_1^1) T_L^{(1/2),K}(s_1^1) b_{min}^{*,3/2}(s_1^1)} + \frac{4A}{T_n^k(s_1^1)} + \gamma\pi(m|s_1^1) \sum_{s_j^2} P(s_j^2|s_1^1,m)\mathbb{V}(s_j^2)} \right]$$

$$\overset{(a)}{\geqslant} \frac{\sqrt{\pi^2(p|s_1^1)\sigma^2(s_1^1,p)}}{\frac{B(s_1^1)}{T_L^K(s_1^1)}} \left[1 - 4dc \frac{\sqrt{\log(H/\delta)}}{B(s_1^1) T_L^{(1/2),K}(s_1^1) b_{min}^{*,3/2}(s_1^1)} - \frac{4A}{T_n^k(s_1^1)} - \gamma\pi(m|s_1^1) \frac{T_L^K(s_1^1)}{B(s_1^1)} \sum_{s_j^2} P(s_j^2|s_1^1,m)\mathbb{V}(s_j^2) \right]$$

$$\overset{(b)}{=} \frac{\sqrt{\pi^2(p|s_1^1)\sigma^2(s_1^1,p)}}{\frac{B(s_1^1)}{T_L^K(s_1^1)}} \Bigg[1 - 4dc \frac{\sqrt{\log(H/\delta)}}{B(s_1^1) T_L^{(1/2),K}(s_1^1) b_{min}^{*,3/2}(s_1^1)} - \frac{4A}{T_n^k(s_1^1)}$$
$$- \gamma\pi(m|s_1^1) \frac{T_L^K(s_1^1)}{B(s_1^1)} \sum_{s_j^2} P(s_j^2|s_1^1,m) \left(\frac{B(s_j^2)}{T_L^K(s_j^2)} + 4dc \frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2) b_{min}^{*,3/2}(s_j^2)} + \frac{4AB(s_j^2)}{T_L^{(2),K}(s_j^2) - 1} \right) \Bigg]$$

$$\overset{(c)}{\geqslant} T_L^{*,K}(s_1^1,p) - 4dc\, b^*(p|s_1^1) \frac{\sqrt{\log(H/\delta)}}{B(s_1^1) b_{min}^{*,3/2}(s_1^1)} \sqrt{T_L^K(s_1^1)} - 4A\, b^*(p|s_1^1)$$
$$- \gamma\pi(m|s_1^1) \sum_{s_j^2} \frac{B(s_j^2) T_L^K(s_j^2)}{b^*(m|s_j^2) B(s_1^1)} P(s_j^2|s_1^1,m) \left(\frac{B(s_j^2)}{T_L^K(s_j^2)} + 4dc \frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2) b_{min}^{*,3/2}(s_j^2)} + \frac{4AB(s_j^2)}{T_L^{(2),K}(s_j^2) - 1} \right) \Bigg]$$

$$
\begin{aligned}
&\geqslant T_L^{*,K}(s_1^1,p) - 4dcb^*(p|s_1^1)\frac{\sqrt{\log(H/\delta)}}{B(s_1^1)b_{min}^{*,3/2}(s_1^1)}\sqrt{T_L^K(s_1^1)} - 4Ab^*(p|s_1^1) \\
&\quad - \gamma\pi(m|s_1^1)\frac{T_L^K(s_1^1)}{B^2(s_1^1)}\sum_{s_j^2} P(s_j^2|s_1^1,m)\left(\frac{B(s_j^2)}{b^*(m|s_j^2)}\sum_{a'}\left[T_L^{*,K}(s_j^2,a')\right.\right. \\
&\quad \left.\left. +4dcb^*(a'|s_j^2)\frac{\sqrt{\log(H/\delta)}}{b_{min}^{*,3/2}(s_j^2)}\sqrt{T_L^K(s_1^1)} + 4Ab^*(a'|s_j^2)\right)\right] \\
&\geqslant T_L^{*,K}(s_1^1,p) - 4dcb^*(p|s_1^1)\frac{\sqrt{\log(H/\delta)}}{B(s_1^1)b_{min}^{*,3/2}(s_1^1)}\sqrt{T_L^K(s_1^1)} - 4Ab^*(p|s_1^1) \\
&\quad - \gamma\pi(m|s_1^1)\frac{T_L^K(s_1^1)}{B^2(s_1^1)}\sum_{s_j^2} P(s_j^2|s_1^1,m)\frac{B(s_j^2)}{b^*(m|s_j^2)}\sum_{a'}\left[T_L^{*,K}(s_j^2,a')\right. \\
&\quad \left. +4dcb^*(a'|s_j^2)\frac{\sqrt{\log(H/\delta)}}{b_{min}^{*,3/2}(s_j^2)}\sqrt{T_L^K(s_1^1)} + 4Ab^*(a'|s_j^2)\right]
\end{aligned}
$$

where in (a) we use $1/(1+x) \geqslant 1-x$ (for $x > -1$), in (b) we substitute the value $\mathbb{V}(s_j^2)$, and (c) follows as $T_L^K(s_j^2) = \left(b(m|s_j^2)/B(s_j^2)\right) T_L^K(s_1^1)$. □

Lemma A.9. *Let the total budget be* $n = KL$ *and* $n \geqslant 4SA$. *Then the total regret in a deterministic* 2*-depth* **T** *at the end of* K*-th episode when sampling according to the* (2.8) *is given by*

$$
\mathcal{R}_n \leqslant \widetilde{O}\left(\frac{B^2(s_1^1)\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}^{*,3/2}(s_1^1)} + \gamma \max_{s_j^2,a}\pi(a|s_1^1)P(s_j^2|s_1^1,a)\frac{B^2(s_j^2)\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}^{*,3/2}(s_j^2)}\right)
$$

where, the $\widetilde{O}$ *hides other lower order terms resulting out of the expansion of the squared terms and* $B(s_i^\ell)$ *is defined in* (2.4).

Proof. **Step 1 (**$T_t^k(s_i^\ell, a)$ **is a stopping time):** Let τ be a random variable, which is defined on the filtered probability space. Then τ is called a stopping time (with respect to the filtration $\left((\mathcal{F}_n)_{n\in\mathbb{N}}\right)$, if the following condition holds: $\{\tau = n\} \in \mathcal{F}_n$ for all n Intuitively, this condition means that the "decision" of whether to stop at time n must be based only on the information present at time n, not on any future information. Now consider the state s_i^ℓ and an action a. At each time step $t+1$, the ReVar algorithm decides which action to pull according to the current values

of the upper-bounds $\left\{\widehat{\sigma^u}^k_{t+1}(s^\ell_i, a)\right\}_a$ in state s^ℓ_i. Thus for any action a, $T^k_{t+1}(s^\ell_i, a)$ depends only on the values $\left\{T^k_{t+1}(s^\ell_i, a)\right\}_a$ and $\left\{\widehat{\sigma}^k_t(s^\ell_i, a)\right\}_k$ in state s^ℓ_i. So by induction, $T^k_t(s^\ell_i, a)$ depends on the sequence of rewards $\left\{R^k_1(s^\ell_i, a), \ldots, R^k_{T^k_t(s^\ell_i, a)}(s^\ell_i, a)\right\}$, and on the samples of the other arms (which are independent of the samples of arm k). So we deduce that $T^K_L(s^\ell_i, a)$ is a stopping time adapted to the process $\left(R^k_t(s^\ell_i, a)\right)_{t \leqslant n}$.

Step 2 (Regret bound): By definition, given the dataset $\mathcal{D}$ after K episodes each of trajectory length L, we have n state-action samples. Then the loss of the algorithm is

$$\begin{aligned}\mathcal{L}_n &= \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s^1_1) - v^\pi(s^1_1)\right)^2\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s^1_1) - v^\pi(s^1_1)\right)^2 \mathbb{I}\{\xi_\delta\}\right] + \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s^1_1) - v^\pi(s^1_1)\right)^2 \mathbb{I}\left\{\xi^C_\delta\right\}\right]\end{aligned}$$

where, $n = KL$ is the total budget. To handle the second term, we recall that ξ^C_δ holds with probability 2δ. Further due to the bounded reward assumption we have

$$\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s^1_1) - v^\pi(s^1_1)\right)^2\right] \leqslant 2n^2K\delta(4\eta^2+2\eta) \leqslant 2(4\eta^2+2\eta)n^2A\delta\left(1 + \log\left(c_2/2nA\delta\right)\right)$$

where $c_2 > 0$ is a constant. Following Lemma 2 of (Carpentier and Munos, 2011) and setting $\delta = n^{-7/2}$ gives us an upper bounds of the quantity

$$\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s^1_1) - v^\pi(s^1_1)\right)^2 \mathbb{I}\left\{\xi^C_\delta\right\}\right] \leqslant O\left(\frac{\log n}{n^{3/2}}\right).$$

Note that Carpentier and Munos (2011) uses a similar $\delta = n^{-7/2}$ due to the sub-Gaussian assumption on their reward distribution. Also observe that under the Assumption 2 we also have a sub-Gaussian assumption. Hence we can use Lemma 2 of Carpentier and Munos (2011). Now, using

the definition of $Y_n(s_1^1)$ and Theorem A.2 we bound the first term as

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-v^{\pi}(s_1^1)\right)^2\mathbb{I}\{\xi_\delta\}\right] \overset{(a)}{=} \mathrm{Var}[Y_n(s_1^1)]\mathbb{E}[T_L^K(s_1^1)] \\
&\leqslant \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)] \\
&+\gamma^2\sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_n(s_j^2)]\mathbb{E}[T_L^K(s_j^2,a)] \\
&\leqslant \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)] \\
&+\gamma^2\sum_a \pi^2(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\sum_{a'}\pi^2(a'|s_j^2)\left[\frac{\sigma^2(s_j^2,a')}{\underline{T}_L^{(2),K}(s_j^2,a')}\right]\mathbb{E}[T_L^K(s_j^2,a')]
\end{aligned} \tag{A.25}$$

where, (a) follows from Theorem A.2, and $\underline{T}_n(s_i^\ell,a)$ is the lower bound on $T_L^K(s_i^\ell,a)$ on the event ξ_δ. Note that as $\sum_a T_L^K(s_1^1,a)=n$, we also have $\sum_a \mathbb{E}\left[T_L^K(s_1^1,a)\right]=n$. Using eq. (A.25) and eq. (A.24) for

$$\pi^2(a|s_1^1)\sigma^2(s_1^1,a)/\underline{T}_n^k(s_1^1,a)$$

(which is equivalent to using a lower bound on $T_L^K(s_1^1,a)$ on the event ξ_δ), we obtain

$$\begin{aligned}
&\sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_n(s_1^1)] \\
&\leqslant \sum_a\Bigg(\left[\frac{B(s_1^1)}{T_L^K(s_1^1)}+4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_1^1)b_{\min}^{*,3/2}(s_1^1)}+\frac{4AB(s_1^1)}{T_L^{(2),K}(s_1^1)-1}\right] \\
&+\gamma\pi(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)\sum_{a'}\left[\frac{B(s_j^2)}{T_L^K(s_j^2)}+4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2)b_{\min}^{*,3/2}(s_j^2)}+\frac{4AB(s_j^2)}{T_L^{(2),K}(s_j^2)-1}\right]\Bigg)^2\cdot \\
&\mathbb{E}[T_L^K(s_1^1,a)].
\end{aligned} \tag{A.26}$$

Finally the R.H.S. of eq. (A.26) may be bounded using the fact that $\sum_a \mathbb{E}\left[T_L^K(s_1^1, a)\right] = n$ as

$$\begin{aligned}
&\sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{T_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1)] \leqslant \sum_a \Bigg(\left[\frac{B(s_1^1)}{T_L^K(s_1^1)} + 4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_1^1)b_{min}^{*,3/2}(s_1^1)} + \frac{4AB(s_1^1)}{T_L^{(2),K}(s_1^1)-1}\right] \\
&\quad + \gamma\pi(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\sum_{a'}\left[\frac{B(s_j^2)}{T_L^K(s_j^2)} + 4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2)b_{min}^{*,3/2}(s_j^2)} + \frac{4AB(s_j^2)}{T_K^{(2),L}(s_j^2)-1}\right]\Bigg)^2 \mathbb{E}[T_L^K(s_1^1,a)] \\
&\overset{(a)}{\leqslant} 2\Bigg(\left[\frac{B(s_1^1)}{T_L^K(s_1^1)} + 4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_1^1)b_{min}^{*,3/2}(s_1^1)} + \frac{4AB(s_1^1)}{T_L^{(2),K}(s_1^1)-1}\right]\Bigg)^2 \sum_a \mathbb{E}[T_L^K(s_1^1,a)] \\
&\quad + 2\Bigg(\gamma\pi(a|s_1^1)\sum_{s_j^2} P(s_j^2|s_1^1,a)\sum_{a'}\left[\frac{B(s_j^2)}{T_L^K(s_j^2)} + 4dc\frac{\sqrt{\log(H/\delta)}}{T_L^{(3/2),K}(s_j^2)b_{min}^{*,3/2}(s_j^2)} + \frac{4AB(s_j^2)}{T_L^{(2),K}(s_j^2)-1}\right]\Bigg)^2 \sum_a \mathbb{E}[T_L^K(s_1^1,a)] \\
&\overset{(b)}{\leqslant} \widetilde{O}\left(\frac{B^2(s_1^1)\sqrt{\log(H/\delta)}}{n^{3/2}b_{min}^{*,3/2}(s_1^1)} + \gamma\max_{s_j^2,a}\pi(a|s_1^1)P(s_j^2|s_1^1,a)\frac{B^2(s_j^2)\sqrt{\log(H/\delta)}}{n^{3/2}b_{min}^{*,3/2}(s_j^2)}\right) \\
&\overset{(c)}{=} \widetilde{O}\left(\frac{B^2(s_1^1)\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}^{*,3/2}(s_1^1)} + \gamma\max_{s_j^2,a}\pi(a|s_1^1)P(s_j^2|s_1^1,a)\frac{B^2(s_j^2)\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}^{*,3/2}(s_j^2)}\right)
\end{aligned}$$

where, (a) follows as $(a+b)^2 \leqslant 2(a^2+b^2)$ for any $a, b > 0$, in (b) we have $T_L^K(s_1^1) = n$, and the $\widetilde{O}$ hides other lower order terms resulting out of the expansion of the squared terms, and (c) follows by setting $\delta = n^{-7/2}$ and using $H = SAn(n+1)$. □

A.8 Regret for a Deterministic L-Depth Tree

Theorem 2. *Let the total budget be* $n = KL$ *and* $n \geqslant 4SA$. *Then the total regret in a deterministic* L*-depth* **T** *at the end of* K*-th episode when taking actions according to* (2.8) *is given by*

$$\mathcal{R}_n \leqslant \widetilde{O}\left(\frac{B_{s_1^1}^2\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}^{*,3/2}(s_1^1)} + \gamma\sum_{\ell=2}^{L}\max_{s_j^\ell,a}\pi(a|s_1^1)P(s_j^\ell|s_1^1,a)\frac{B_{s_j^\ell}^2\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}^{*,3/2}(s_j^\ell)}\right)$$

where, the $\widetilde{O}$ *hides other lower order terms and* $B_{s_i^\ell}$ *is defined in* (2.4) *and* $b_{min}^*(s) = \min_a b^*(a|s)$.

Proof. The proof follows directly by using Theorem A.7, Theorem A.8, and Theorem A.9.

Step 1 ($T_t^k(s_i^\ell, a)$ is a stopping time): This step is same as Theorem A.9 as all the arguments hold true even for the L depth deterministic tree.

Step 2 (MSE decomposition): Given the dataset $\mathcal{D}$ of K episodes each of trajectory length L, the MSE of the algorithm is

$$\begin{aligned}\mathcal{L}_n &= \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - v^\pi(s_1^1)\right)^2\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - v^\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_\delta\}\right] + \mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - v^\pi(s_1^1)\right)^2 \mathbb{I}\left\{\xi_\delta^C\right\}\right]\end{aligned}$$

where, $n = KL$ is the total budget. Using Theorem A.9 we can upper bound the second term as $O\left(n^{-3/2}\log(n)\right)$. Using the definition of $Y_n(s_1^1)$ and Theorem A.2 we bound the first term as

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - v^\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_\delta\}\right] \overset{(a)}{=} \mathrm{Var}[Y_n(s_1^1)]\mathbb{E}[T_L^K(s_1^1)] = \\ &\overset{b}{\leqslant} \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)] \\ &+ \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_n(s_j^2)]\mathbb{E}[T_L^K(s_j^2)] \\ &\overset{(c)}{\leqslant} \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)] \\ &\quad + \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{\ell=2}^{L} \sum_{s_j^\ell} P(s_j^\ell|s_1^1,a) \sum_{a'} \pi^2(a'|s_j^\ell)\left[\frac{\sigma^2(s_j^\ell,a')}{\underline{T}_L^{(2),K}(s_j^\ell,a')}\right]\mathbb{E}[T_L^K(s_j^\ell,a')]\end{aligned} \tag{A.27}$$

where, (a) follows from Theorem A.2, (b) follows from by unrolling the variance for $Y_n(s_1^1)$, and where $\underline{T}_n(s_i^\ell, a)$ is the lower bound on $T_L^K(s_i^\ell, a)$ on the event ξ_δ. Finally, (c) follows by unrolling the variance for all the states till level L and taking the lower bound of $\underline{T}_n(s_i^\ell, a)$ for each state-action pair.

Step 2 (MSE at level L): Now we want to upper bound the total MSE in (A.27). Using eq. (A.21) in Theorem A.7 we can directly get the MSE upper bound for a state s_i^L as

$$\sum_{a'} \pi^2(a'|s_i^L)\left[\frac{\sigma^2(s_i^L,a')}{\underline{T}_L^{(2),K}(s_i^L,a')}\right]\mathbb{E}[T_L^K(s_i^L,a')] \leqslant \widetilde{O}\left(\frac{B^2_{s_i^L}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}(s_i^L)}\right).$$

Step 3 (MSE at level L − 1): This step follows directly from eq. (A.24) in Theorem A.8. We can get the loss upper bound for a state s_i^{L-1} (which takes into account the loss at level L as well) as follows:

$$\begin{aligned}
&\sum_{a'}\left(\frac{b^*(a'|s_i^{L-1})}{\underline{T}_L^{(2),K}(s_i^{L-1},a')}\right)\mathbb{E}[T_L^K(s_i^{L-1},a')] \\
&\leqslant \widetilde{O}\left(\frac{B^2_{s_i^{L-1}}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}(s_i^{L-1})} + \gamma\max_{s_j^L,a}\pi(a|s_i^{L-1})P(s_j^L|s_i^{L-1},a)\frac{B^2_{s_j^L}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}(s_j^L)}\right).
\end{aligned}$$

Step 4 (MSE at arbitrary level ℓ): This step follows by combining the results of step 2 and 3 iteratively from states in level ℓ to L under the good event ξ_δ. We can get the regret upper bound for a state s_i^ℓ as

$$\begin{aligned}
&\sum_{a'}\left(\frac{b^*(a'|s_i^\ell)}{\underline{T}_L^{(2),K}(s_i^\ell,a')}\right)\mathbb{E}[T_L^K(s_i^\ell,a')] \\
&\leqslant \widetilde{O}\left(\frac{B^2_{s_i^\ell}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}(s_i^\ell)} + \gamma\sum_{\ell'=\ell+1}^{L}\max_{s_j^{\ell'},a}\pi(a|s_i^{\ell'-1})P(s_j^{\ell'}|s_i^{\ell'-1},a)\frac{B^2_{s_j^{\ell'}}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b_{min}(s_j^{\ell'})}\right).
\end{aligned}$$

Step 4 (Regret at level 1): Finally, combining all the steps above we get the regret upper bound for the state s_1^1 as follows

$$\begin{aligned}
&\mathcal{R}_n = \mathcal{L}_n - \mathcal{L}_n^* \\
&= \widetilde{O}\left(\frac{B^2(s_1^1)\sqrt{\log(SAn^{11/2})}}{n^{3/2}b^{*,3/2}_{min}(s_1^1)} + \gamma\sum_{\ell=2}^{L}\max_{s_j^\ell,a}\pi(a|s_1^1)P(s_j^\ell|s_1^1,a)\frac{B^2(s_j^\ell)\sqrt{\log(SAn^{11/2})}}{n^{3/2}b^{*3/2}_{min}(s_j^\ell)}\right).
\end{aligned}$$

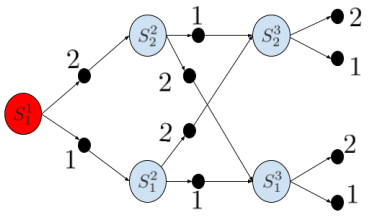


Figure A.3: A 3-depth 2-Action DAG

□

Remark A.10. (***Stochastic MDP extension***)**:** *Observe that the Theorem 2 is quite general as the regret*

$$\mathcal{R}_n \leqslant \widetilde{O}\left(\frac{B^2_{s_1^1}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b^{*,3/2}_{\min}(s_1^1)} + \gamma\sum_{\ell=2}^{L}\max_{s_j^\ell,a}\pi(a|s_1^1)P(s_j^\ell|s_1^1,a)\frac{B^2_{s_j^\ell}\sqrt{\log(SAn^{11/2})}}{n^{3/2}b^{*,3/2}_{\min}(s_j^\ell)}\right)$$

incorporates the transition probability $P(s'|s,a)$*. Hence, the result of Theorem 2 holds not only for the deterministic case but also for the stochastic setting, when the algorithm is provided with the knowledge of* $P(s'|s,a)$ *upto some constant scaling. Note that ReVar does not perform any exploration to estimate the transition probabilities, and it is not clear how to extend the current UCB based approach that minimizes MSE to also estimate the* P*. We leave this direction for future works.*

A.9 DAG Optimal Sampling

Proposition 3. (***Restatement***) *Let* $\mathcal{G}$ *be a* 3*-depth,* A*-action DAG defined in Theorem 2.5. The minimal-MSE sampling proportions* $b^*(a|s_1^1), b^*(a|s_j^2)$ *depend on themselves such that* $b(a|s_1^1) \propto f(1/b(a|s_1^1))$ *and* $b(a|s_j^2) \propto f(1/b(a|s_j^2))$ *where* $f(\cdot)$ *is a function that hides other dependencies on variances of* s *and its children.*

Proof. **Step 1 (Level 3):** For an arbitrary state s_i^3 we can calculate the expectation and variance of $Y_n(s_i^3)$ as follows:

$$\mathbb{E}[Y_n(s_i^3)] = \sum_a \frac{\pi(a|s_i^3)}{T_n(s_i^3, a)} \sum_{h=1}^{T_n(s_i^3,a)} \mathbb{E}[R_h(s_i^3, a)] = \sum_a \pi(a|s_i^3)\mu(s_i^3, a)$$

$$\mathrm{Var}[Y_n(s_i^3)] = \sum_a \frac{\pi^2(a|s_i^3)}{T_n^2(s_i^3, a)} \sum_{h=1}^{T_n(s_i^3,a)} \mathrm{Var}[R_h(s_i^3, a)] = \sum_a \frac{\pi^2(a|s_i^3)}{T_n(s_i^3, a)}\sigma^2(s_i^3, a).$$

Step 2 (Level 2): For the arbitrary state s_i^2 we can calculate the expectation of $Y_n(s_1^2)$ as follows:

$$\begin{aligned}
\mathbb{E}[Y_n(s_i^2)] &= \sum_a \frac{\pi(a|s_i^2)}{T_n(s_i^2, a)} \sum_{h=1}^{T_n(s_i^2,a)} \mathbb{E}[R_h(s_i^2, a)] \\
&\quad + \gamma \sum_a \pi(a|s_i^2) \sum_{s_j^3} P(s_j^3|s_i^2, a) \sum_{a'} \frac{\pi(a'|s_j^3)}{T_n(s_j^3, a')} \sum_{h=1}^{T_n(s_j^3,a')} \mathbb{E}[R_h(s_j^3, a')] \\
&= \sum_a \pi(a|s_i^2) \left(\mu(s_i^2, a) + \gamma \sum_{s_j^3} P(s_j^3|s_i^2, a)\mathbb{E}[Y_n(s_j^3)] \right) \\
\mathrm{Var}[Y_n(s_i^2)] &= \sum_a \frac{\pi^2(a|s_1^2)}{T_n^2(s_1^2, a)} \sum_{h=1}^{T_n(s_1^2,a)} \mathrm{Var}[R_h(s_1^2, a)] \\
&\quad + \gamma^2 \sum_a \pi^2(a|s_1^2) \sum_{s_j^3} P(s_j^3|s_1^2, a) \sum_{a'} \frac{\pi^2(a'|s_j^3)}{T_n^2(s_j^3, a')} \sum_{h=1}^{T_n(s_j^3,a')} \mathrm{Var}[R_h(s_j^3, a')] \\
&= \sum_a \frac{\pi^2(a|s_1^2)}{T_n(s_1^2, a)} \left(\sigma^2(s_1^2, a) + \gamma^2 \sum_{s_j^3} P(s_j^3|s_1^2, a)\mathrm{Var}[Y_n(s_j^3)] \right)
\end{aligned}$$

Step 3 (Level 1): Finally for the state s_1^1 we can calculate the expectation

and variance of $Y_n(s_1^1)$ as follows:

$$\begin{aligned}
\mathbb{E}[Y_n(s_1^1)] &= \sum_a \frac{\pi(a|s_1^1)}{T_n(s_1^1,a)} \sum_{h=1}^{T_n(s_1^1,a)} \mathbb{E}[R_h(s_1^1,a)] \\
&\quad + \gamma\pi(a|s_1^1) \sum_{s_j^2} P(s_j^2|s_1^1,a) \sum_{a'} \frac{\pi(a'|s_j^2)}{T_n(s_j^2,a')} \sum_{h=1}^{T_n(s_j^2,a')} \mathbb{E}[R_h(s_j^2,a')] \\
&= \sum_a \pi(a|s_1^1) \left(\mu(s_1^1,a) + \gamma \sum_{s_j^2} P(s_j^2|s_1^1,a)\mathbb{E}[Y_n(s_j^2)] \right) \\
\mathrm{Var}[Y_n(s_1^1)] &= \sum_a \frac{\pi^2(a|s_1^1)}{T_n^2(s_1^1,a)} \sum_{h=1}^{T_n(s_1^1,a)} \mathrm{Var}[R_h(s_1^1,a)] \\
&\quad + \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} P(s_j^2|s_1^1,a) \sum_{a'} \frac{\pi^2(a'|s_j^2)}{T_n^2(s_j^2,a')} \sum_{h=1}^{T_n(s_j^2,a')} \mathrm{Var}[R_h(s_j^2,a')] \\
&= \sum_a \frac{\pi^2(a|s_1^1)}{T_n(s_1^1,a)} \left(\sigma^2(s_1^1,a) + \gamma^2 \sum_{s_j^2} P(s_j^2|s_1^1,a)\mathrm{Var}[Y_n(s_j^2)] \right)
\end{aligned}$$

Unrolling out the above equation we re-write the equation below:

$$\begin{aligned}
\mathrm{Var}[Y_n(s_1^1)] &= \sum_a \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{T_n(s_1^1,a)} + \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{T_n(s_j^2,a')} \\
&\quad + \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3,a'')}{T_n(s_m^3,a'')}
\end{aligned} \tag{A.28}$$

Since we follow a path $s_1^1 \xrightarrow{a} s_j^2 \xrightarrow{a'} s_m^3 \xrightarrow{a''}$ Terminate for any $a, a', a'' \in \mathcal{A}$

and $j, m \in \{1, 2, \ldots, A\}$. Hence we have the following constraints

$$\sum_a T_n(s_1^1, a) = n \tag{A.29}$$

$$\sum_a T_n(s_i^2, a) \overset{(a)}{=} \sum_a P(s_i^2|s_1^1, a) T_n(s_1^1, a) \tag{A.30}$$

$$\sum_a T_n(s_i^3, a) \overset{(b)}{=} \sum_{s_j^2} \sum_{a'} P(s_i^3|s_j^2, a') T_n(s_j^2, a') \tag{A.31}$$

observe that in (a) in the deterministic case the $\sum_a P(s_i^2|s_1^1, a) T_n(s_1^1, a)$ is all the possible paths from s_1^1 to s_i^2 that were taken for n samples over any action a. Similarly in (b) in the deterministic case the $\sum_{a'} P(s_i^3|s_j^2, a) T_n(s_j^2, a')$ is all the possible paths from s_j^2 to s_i^3 that were taken for n samples over any action a'.

Step 4 (Formulate objective): We want to minimize the variance in (A.28) subject to the above constraints. We can show that

$$T_n(s_1^1, a)/n = b(a|s_1^1). \tag{A.32}$$

$$\begin{aligned} \text{and,} \qquad b(a|s_i^2) &= \frac{T_n(s_i^2, a)}{\sum_{a'} T_n(s_i^2, a')} = \frac{T_n(s_i^2, a)}{\sum_{a'} P(s_i^2|s_1^1, a') T_n(s_1^1, a')} \\ &\overset{(a)}{=} \frac{T_n(s_i^2, a)/n}{\sum_{a'} P(s_i^2|s_1^1, a') T_n(s_1^1, a')/n} \\ \implies T_n(s_i^2, a)/n &\overset{(b)}{=} b(a|s_i^2) \sum_{a'} P(s_i^2|s_1^1, a') b(a'|s_1^1), \end{aligned} \tag{A.33}$$

where, (a) follows from (A.30), and (b) follows from (A.32) and taking into account all the possible paths to reach s_i^2 from s_1^1. For the third level

we can show that the proportion

$$
\begin{aligned}
b(a|s_i^3) = \frac{T_n(s_i^3, a)}{\sum_{a'} T_n(s_i^3, a')} &\overset{(a)}{=} \frac{T_n(s_i^3, a)}{\sum_{s_j^2} \sum_{a'} P(s_i^3|s_j^2, a') T_n(s_j^2, a')} \\
&\overset{(b)}{=} \frac{T_n(s_i^3, a)}{\sum_{s_j^2} \sum_{a'} P(s_j^2|s_1^1, a') b(a'|s_1^1) \sum_{a''} P(s_i^3|s_j^2, a'') b(a''|s_j^2)} \\
&\implies T_n(s_i^3, a)/n = b(a|s_i^3) \sum_{s_j^2} \sum_{a'} P(s_j^2|s_1^1, a') b(a'|s_1^1) \sum_{a''} P(s_i^3|s_j^2, a'') b(a''|s_j^2)
\end{aligned}
$$

where, (a) follows from (A.31), and (b) follows from (A.32) and taking into account all the possible paths to reach s_i^3 from s_1^1. Again note that we use $b(a|s)$ to denote the optimization variable and $b^*(a|s)$ to denote the optimal sampling proportion. Then the optimization problem in (A.28) can be restated as,

$$
\begin{aligned}
\min_{b} \sum_{a} \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1, a)}{b(a|s_1^1)} &+ \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b(a'|s_j^2) \underbrace{\sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1)}_{\text{All possible path to reach } s_j^2 \text{ from } s_1^1}} \\
&+ \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \cdot \\
&\sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{b(a''|s_m^3) \underbrace{\sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2, a_2) b(a_2|s_j^2)}_{\text{All possible path to reach } s_m^3 \text{ from } s_1^1}}
\end{aligned}
$$

$$
\begin{aligned}
\textbf{s.t.} \quad \forall s, \quad & \sum_{a} b(a|s) = 1 \\
\forall s, a \quad & b(a|s) > 0.
\end{aligned}
$$

Now introducing the Lagrange multiplier we get that

$$
\begin{aligned}
L(\mathbf{b},\lambda) = \min_{b} & \sum_{a} \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1,a)}{b(a|s_1^1)} \\
& + \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{b(a'|s_j^2)\sum_{a_1} P(s_j^2|s_1^1,a_1)b(a_1|s_1^1)} \\
& \quad + \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2)\cdot \\
& \quad \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3,a'')}{b(a''|s_m^3)\sum_{s_j^2}\sum_{a_1} P(s_j^2|s_1^1,a_1)b(a_1|s_1^1)\sum_{a_2} P(s_i^3|s_j^2,a_2)b(a_2|s_j^2)} \\
& \quad + \sum_{s} \lambda_s \left(\sum_{a} b(a|s) - 1 \right). \qquad \text{(A.34)}
\end{aligned}
$$

Now we need to solve for the KKT condition. Differentiating (A.34) with respect to $b(a''|s_m^3)$, $b(a'|s_j^2)$, $b(a|s_1^1)$, and λ_s we get

$$\nabla_{b(a''|s_m^3)} L(\mathbf{b}, \check{}) = -\sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{b^2(a''|s_m^3) \sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2, a_2) b(a_2|s_j^2)} + \lambda_{s_m^3} \tag{A.35}$$

$$\nabla_{b(a'|s_j^2)} L(\mathbf{b}, \check{}) = -\sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b^2(a'|s_j^2) \sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1)} \tag{A.36}$$

$$- \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{b(a''|s_m^3) \left(\sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2, a_2) b(a_2|s_j^2) \right)^2} + \lambda_{s_j^2}$$

$$\nabla_{b(a|s_1^1)} L(\mathbf{b}, \check{}) = -\sum_a \frac{\pi^2(a|s_1^1)\sigma^2(s_1^1, a)}{b^2(a|s_1^1)} - \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{b(a'|s_j^2) \left(\sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1) \right)^2} \tag{A.37}$$

$$- \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{b(a''|s_m^3) \left(\sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2, a_2) b(a_2|s_j^2) \right)^2} + \lambda_{s_1^1}$$

$$\nabla_{\lambda_s} L(\mathbf{b}, \check{}) = \sum_a b(a|s) - 1. \tag{A.38}$$

Now to remove $\lambda_{s_m^3}$ from (A.35) we first set (A.35) to 0, define

$$P(s_j 2 \to s_j^3) = \sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1, a_1) b(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2, a_2) b(a_2|s_j^2)$$

and show that

$$\lambda_{s_m^3} = \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{b^2(a''|s_m^3) \left(P(s_j 2 \to s_j^3) \right)^2}$$

$$\implies b(a''|s_m^3) = \sqrt{\frac{1}{\lambda_{s_m^3}} \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{\left(P(s_j 2 \to s_j^3) \right)^2}}. \tag{A.39}$$

Then setting (A.38) to 0 we have

$$\sum_{a''} \sqrt{\frac{1}{\lambda_{s_m^3}} \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{\left(P(s_j 2 \to s_j^3)\right)^2}} = 1$$

$$\implies \lambda_{s_m^3} = \sum_{a''} \sqrt{\sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{\left(P(s_j 2 \to s_j^3)\right)^2}} \tag{A.40}$$

Using (A.39) and (A.40) we can show that the optimal sampling proportion is given by

$$b^*(a''|s_m^3) = \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{\sum_a \pi^2(a|s_m^3)\sigma^2(s_m^3, a)}$$

Similarly we can show that setting (A.36) and (A.38) setting to 0 and removing $\lambda_{s_j^2}$

$$b^{*,(2)}(a'|s_j^2) \propto \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2, a')}{\sum_{a_1} P(s_j^2|s_1^1, a_1) b^*(a_1|s_1^1)}$$

$$+ \sum_{a} \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2)$$

$$\cdot \sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3, a'')}{b^*(a''|s_m^3)} \left(\frac{\sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1, a_1) b^*(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2, a_2) b^*(a_2|s_j^2)}{b^*(a'|s_j^2)} \right)^2$$

Finally, setting (A.37) and (A.38) setting to 0 and removing $\lambda_{s_1^1}$ we have

$$b^{*,(2)}(a|s_1^1) \propto \sum_a \pi^2(a|s_1^1)\sigma^2(s_1^1,a) + \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \frac{\pi^2(a'|s_j^2)\sigma^2(s_j^2,a')}{b^*(a'|s_j^2)}$$

$$+ \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} \sum_{a'} \pi^2(a'|s_j^2) \cdot$$

$$\sum_{s_m^3} \sum_{a''} \frac{\pi^2(a''|s_m^3)\sigma^2(s_m^3,a'')}{b^*(a''|s_m^3)} \left(\frac{\sum_{s_j^2} \sum_{a_1} P(s_j^2|s_1^1,a_1) b^*(a_1|s_1^1) \sum_{a_2} P(s_i^3|s_j^2,a_2) b^*(a_2|s_j^2)}{b^*(a|s_1^1)} \right)^2$$

This shows the cyclical dependency of $b^*(a|s_1^1)$ and $b^*(a|s_j^2)$. □

A.10 Additional Experimental Details

Estimate B in DAG

Recall that in a DAG $\mathcal{G}$ we have a cyclical dependency following Proposition 3. Hence, we do an approximation of the optimal sampling proportion in $\mathcal{G}$ by using the tree formulation from Theorem 1. However, since there are multiple paths to the same state in $\mathcal{G}$ we have to iteratively compute the normalization factor B. To do this we use the following Algorithm 12.

Algorithm 12 Estimate $B_0(s)$ for $\mathcal{G}$

1: Initialize $B_L(s) = 0$ for all $s \in \mathcal{S}$
2: **for** $t' \in L-1,\dots,0$ **do**
3: $$B_{t'}(s) = \sum_a \sqrt{\pi^2(a|s)\left(\sigma^2(s,a) + \gamma^2 \sum_{s'} P(s'|s,a) B^2_{t'+1}(s)\right)}$$
4: **Return** B_0.

Implementation Details

In this section we state additional experimental details. We implement the following competitive baselines:

(1) Onpolicy: The Onpolicy baseline follows the target probability when sampling actions at each state.

(2) CB-Var: This baseline is a bandit policy which samples an action based only on the statistics of the current state. At every time $t+1$ in episode k, CB-Var sample an action

$$I_{t+1}^k = \arg\max_{a \in \mathcal{A}} (2\eta + 4\eta^2) \sqrt{\frac{2\pi(a|s)\widehat{\sigma}_t^{(2),k}(s,a)\log(SAn(n+1))}{T_t^k(s,a)}} + \frac{7\log(SAn(n+1))}{3T_t^k(s,a)}$$

where, n is the total budget. This policy is similar to UCB-variance of Audibert et al. (2009) and uses the empirical Bernstein inequality (Maurer and Pontil, 2009). However we do not use the mean estimate $\widehat{\mu}_t^k(s,a)$ of an action so that CB-Var explores continuously rather than maximizing the rewards. Also note that to have a fair comparison with ReVar we use a large constant $(2\eta + 4\eta^2)$ and log term instead of just 2 and $\log t$.

Ablation study

In this experiment we show an ablation study of different values of the upper confidence bound constant associated with $\widehat{\sigma^u}_t^k(s,a)$. Recall from (2.9) that

$$\widehat{\sigma^u}_t^k(s_i^\ell, a) := \widehat{\sigma}_t^k(s_i^\ell, a) + 2c\sqrt{\frac{\log(SAn(n+1)/\delta)}{T_t^k(s_i^\ell, a)}}$$

where, c is the upper confidence bound constant, and $n = KL$. From Theorem 2 we know that the theoretically correct constant is to use $2\eta + 4\eta^2$.

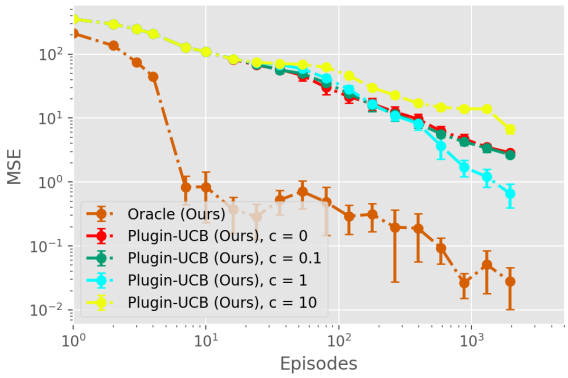


Figure A.4: Ablation study of UCB constant

However, since our upper bound is loose because of union bounds over states, actions, episodes and horizon, we ablate the value of c to see its impact on ReVar . From Figure A.4 we see that too large a value of $c = 10$ and we end up doing too much exploration rather than focusing on the state-action pair that reduces variance. However, even with too small values of $c \in \{0, 0.1\}$ we end up doing less exploration and have very bad plug-in estimates of the variance. Consequently this increases the MSE of ReVar . The value $c = 1$ seems to do relatively well against all the other choices.

A.11 Table of Notations

Notations	Definition
s_i^ℓ	State s in level ℓ indexed by i
$\pi(a\|s_i^\ell)$	Target policy probability for action a in s_i^ℓ
$b(a\|s_i^\ell)$	Behavior policy probability for action a in s_i^ℓ
$\sigma^2(s_i^\ell, a)$	Variance of action a in s_i^ℓ
$\widehat{\sigma}_t^{(2),k}(s_i^\ell, a)$	Empirical variance of action a in s_i^ℓ at time t in episode k
$\widehat{\sigma^u}_t^{(2),k}(s_i^\ell, a)$	UCB on variance of action a in s_i^ℓ at time t in episode k
$\mu(s_i^\ell, a)$	Mean of action a in s_i^ℓ
$\widehat{\mu}_t^k(s_i^\ell, a)$	Empirical mean of action a in s_i^ℓ at time t in episode k
$\mu^2(s_i^\ell, a)$	Square of mean of action a in s_i^ℓ
$\widehat{\mu}_t^{(2),k}(s_i^\ell, a)$	Square of empirical mean of action a in s_i^ℓ at time t in episode k
$T_n(s_i^\ell, a)$	Total Samples of action a in s_i^ℓ after n timesteps
$T_n(s_i^\ell)$	Total samples of actions in s_i^ℓ as $\sum_a T_n(s_i^\ell, a)$ after n timesteps (State count)
$T_t^k(s_i^\ell, a)$	Total samples of action a taken till episode k time t in s_i^ℓ
$T_t^k(s_i^\ell, a, s_j^{\ell+1})$	Total samples of action a taken till episode k time t in s_i^ℓ to transition to $s_j^{\ell+1}$
$P(s_j^{\ell+1}\|s_i^\ell, a)$	Transition probability of taking action a in state s_i^ℓ and transition to state $s_j^{\ell+1}$
$\widehat{P}_t^k(s_j^{\ell+1}\|s_i^\ell, a)$	Empirical transition probability of taking action a in state s_i^ℓ and moving to state $s_j^{\ell+1}$ at time t episode k
$\widehat{P}_t^{(2),k}(s_j^{\ell+1}\|s_i^\ell, a)$	Empirical square of transition probability of taking action a in state s_i^ℓ and moving to state $s_j^{\ell+1}$ at time t episode k

Table A.1: Table of Notations for ReVar

B APPENDIX: SPEED: EXPERIMENTAL DESIGN FOR POLICY EVALUATION IN LINEAR HETEROSCEDASTIC BANDITS

Probability Tools

Lemma B.1. (***Kiefer and Wolfowitz, 1960***) *Assume that $\mathcal{A} \subset \mathbb{R}^d$ is compact and $\operatorname{span}(\mathcal{A}) = \mathbb{R}^d$. Let $\pi : \mathcal{A} \to [0,1]$ be a distribution on $\mathcal{A}$ so that $\sum_{a\in\mathcal{A}} \pi(a) = 1$ and $\mathbf{V}(\pi) \in \mathbb{R}^{d\times d}$ and $g(\pi) \in \mathbb{R}$ be given by*

$$\mathbf{V}(\pi) = \sum_{a\in\mathcal{A}} \pi(a) a a^\top, \quad g(\pi) = \max_{a\in\mathcal{A}} \|a\|^2_{\tilde{\mathbf{X}}(\pi)^{-1}}$$

Then the following are equivalent:

(a) *π^* is a minimizer of g.*

(b) *π^* is a maximizer of $f(\pi) = \log\det \mathbf{V}(\pi)$.*

(c) *$g(\pi^*) = d$.*

Furthermore, there exists a minimizer π^* of g such that $|\operatorname{Supp}(\pi^*)| \leqslant d(d+1)/2$.

Lemma B.2. (***Sub-Exponential Concentration***) *Suppose that X is sub-exponential with parameters (ν, α). Then*

$$\mathbb{P}[X \geqslant \mu + t] \leqslant \begin{cases} e^{-\frac{t^2}{2\nu^2}} & \text{if } 0 \leqslant t \leqslant \frac{\nu^2}{\alpha} \\ e^{-\frac{t}{2\alpha}} & \text{if } t > \frac{\nu^2}{\alpha} \end{cases}$$

which can be equivalently written as follows:

$$\mathbb{P}[X \geqslant \mu + t] \leqslant \exp\left\{-\frac{1}{2}\min\left\{\frac{t}{\alpha}, \frac{t^2}{\nu^2}\right\}\right\}.$$

Lemma B.3. (*Restatement of Theorem 2.2 in Rigollet and Hütter (2015)*) *Assume that the linear model holds where the noise* $\varepsilon \sim \mathrm{subG}_n\left(\sigma^2\right)$. *Then the least squares estimator* $\widehat{\theta}_r$ *satisfies*

$$\mathbb{E}\left[\mathrm{MSE}\left(X\widehat{\theta}_r\right)\right] = \frac{1}{n}\mathbb{E}\left|X\widehat{\theta}_r - X\theta^*\right|_2^2 \lesssim \sigma^2\frac{r}{n}$$

where $r = \mathrm{rank}\left(X^\top X\right)$. *Moreover, for any* $\delta > 0$, *with probability at least* $1-\delta$, *it holds*

$$\mathrm{MSE}\left(X\widehat{\theta}_r\right) \lesssim \sigma^2\frac{r+\log(1/\delta)}{n}$$

Formulation for PE-Optimal Design to Reduce MSE

Proposition 1. *Let* $\widehat{\theta}_n$ *be the Weighted Least Square (WLS) estimate (3.1) of* θ_* *after observing* n *samples and define* $\mathbf{w}(a) = \pi(a)\mathbf{x}(a)$. *Define the design matrix as* $\mathbf{A}_{\mathbf{b},\mathbf{\Sigma}_*}$ *(see (3.2)). Then the loss is given by*

$$\mathbb{E}\left[\left(\sum_{a=1}^{A}\mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\right)^2\right] = \frac{1}{n}\left(\sum_{a,a'}\mathbf{w}(a)^\top\mathbf{A}_{\mathbf{b},\mathbf{\Sigma}_*}^{-1}\mathbf{w}(a')\right).$$

Proof. Let $T_n(a) \geqslant 0$ be the number of samples of $\mathbf{x}(a)$, hence $n = \sum_{a=1}^{A} T_n(a)$. For each $a \in [A]$, the linear model yields:

$$\frac{1}{T_n(a)}\sum_{i=1}^{T_n(a)} R_i(a) = \mathbf{x}(a)^\top\theta_* + \frac{1}{T_n(a)}\sum_{i=1}^{T_n(a)}\eta_i(a).$$

with $R_i(a)$ being the reward observed for action a taken for the i-th time, $\eta_i(a)$ being the corresponding noise, and $T_n(a)$ is the number of samples of action a. We define the following:

$$\widetilde{Y}_n(a) = \sum_{i=1}^{T_n(a)}\frac{R_i(a)}{\sigma(a)\sqrt{T_n(a)}}, \widetilde{x}_n(a) = \frac{\sqrt{T_n(a)}\mathbf{x}(a)}{\sigma(a)}, \widetilde{\eta}_n(a) = \sum_{i=1}^{T_n(a)}\frac{\eta_i(a)}{\sigma(a)\sqrt{T_n(a)}}$$

so that for all $a \in [A]$, $\widetilde{Y}_n(a) = \widetilde{\mathbf{x}}_n(a)^\top \theta_* + \widetilde{\eta}_n(a)$ where we can show the following regarding the expectation of $\widetilde{\eta}_n(a)$ as

$$\mathbb{E}[\widetilde{\eta}_n(a)] = \mathbb{E}\left[\sum_{i=1}^{T_n(a)} \frac{\eta_i(a)}{\sigma(a)\sqrt{T_n(a)}}\right] = \sum_{i=1}^{T_n(a)} \frac{\mathbb{E}\left[\eta_i(a)\right]}{\sigma(a)\sqrt{T_n(a)}} = 0$$

and the variance as

$$\begin{aligned}\mathrm{Var}\left[\widetilde{\eta}_n(a)\right] = \mathrm{Var}\left[\sum_{i=1}^{T_n(a)} \frac{\eta_i(a)}{\sigma(a)\sqrt{T_n(a)}}\right] &\overset{(a)}{=} \sum_{i=1}^{T_n(a)} \mathrm{Var}\left[\frac{\eta_i(a)}{\sigma(a)\sqrt{T_n(a)}}\right] \\ &= \sum_{i=1}^{T_n(a)} \frac{\mathrm{Var}\left[\eta_i(a)\right]}{\sigma^2(a)T_n(a)} = \frac{T_n(a)\sigma^2(a)}{\sigma^2(a)T_n(a)} = 1\end{aligned}$$

where (a) follows as the noises are independent. We denote by $\mathbf{X} = \left(\widetilde{\mathbf{x}}_n(1)^\top, \cdots, \widetilde{\mathbf{x}}_n(A)^\top\right)^\top \in \mathbb{R}^{A \times d}$ the induced design matrix of the policy. Under the assumption that $\mathbf{X}$ has full rank, the above weighted least squares (WLS) problem has an optimal unbiased estimator $\widehat{\theta}_n = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}$, where

$$\mathbf{Y} = [\widetilde{Y}_n(1), \widetilde{Y}_n(2), \ldots, \widetilde{Y}_n(A)]^\top.$$

Let $\mathbf{J} = [\widetilde{\eta}_n(1), \widetilde{\eta}_n(2), \ldots, \widetilde{\eta}_n(A)]^\top$. Let $\mathbf{w}(a) = \pi(a)\mathbf{x}(a)$. Then the objec-

tive is to bound the loss as follows

$$
\begin{aligned}
&\mathbb{E}\left[\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\widehat{\theta}_n-\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\theta_*\right)^2\right]=\mathbb{E}\left[\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}(\widehat{\theta}_n-\theta_*)\right)^2\right]\\
&=\mathbb{E}\left[\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}\mathbf{Y}-\theta_*\right)\right)^2\right]\\
&=\mathbb{E}\left[\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}(\mathbf{X}\theta_*+\mathbf{J})-\theta_*\right)\right)^2\right]\\
&=\mathbb{E}\left[\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}\mathbf{J}\right)^2\right]\\
&\overset{(a)}{=}\mathbb{E}\left[\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}\mathbf{J}\mathbf{J}^{\top}\mathbf{X}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\right]\\
&=\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}\mathbb{E}\left[\mathbf{J}\mathbf{J}^{\top}\right]\mathbf{X}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\\
&\overset{(b)}{=}\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\mathbf{X}^{\top}\mathbf{I}\mathbf{X}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\\
&=\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\mathbf{X}^{\top}\mathbf{X}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\\
&=\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\sum_{a=1}^{A}\widetilde{\mathbf{x}}_n(a)\widetilde{\mathbf{x}}_n(a)^{\top}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\\
&=\frac{1}{n}\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\sum_{a=1}^{A}\frac{\mathbf{b}(a)\mathbf{x}(a)\mathbf{x}(a)^{\top}}{\sigma(a)^2}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\\
&\overset{(c)}{=}\frac{1}{n}\mathbf{Tr}\left(\sum_{a=1}^{A}\mathbf{w}(a)^{\top}\left(\sum_{a=1}^{A}\mathbf{b}(a)\widetilde{\mathbf{x}}(a)\widetilde{\mathbf{x}}(a)^{\top}\right)^{-1}\sum_{a=1}^{A}\mathbf{w}(a)\right)\\
&=\frac{1}{n}\mathbf{Tr}\left(\sum_{a,a'}\mathbf{w}(a)^{\top}\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*}\mathbf{w}(a')\right)
\end{aligned}
$$

where, in (a) we can introduce the trace operator as for any vector $\mathbf{x}$ we have $\mathbf{Tr}(\mathbf{x}^\top \mathbf{x}) = \|\mathbf{x}\|^2$, (b) follows as the matrix $\mathbb{E}[\mathbf{J}\mathbf{J}^\top]$ has all the non-diagonal element as 0 (since noises are independent and $\mathbf{Cov}(\widetilde{\epsilon}_n(a), \widetilde{\epsilon}_n(a')) = 0$) and the diagonal element are the $\mathrm{Var}[\widetilde{\epsilon}_n(a)] = 1$, and (c) follows as we redefine $\widetilde{\mathbf{x}}(a) = \mathbf{x}(a)/\sigma(a)$. □

Loss is convex

Proposition 2. *The loss function*

$$\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*) = \frac{1}{n}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \mathbf{w}(a')\right)$$

for any arbitrary design proportion $\mathbf{b} \in \triangle(\mathcal{A})$ *and co-variance matrix* $\mathbf{\Sigma}_*$ *is strictly convex.*

Proof. Let $\mathbf{b}, \mathbf{b}' \in \triangle(\mathcal{A})$, so that $\mathbf{A}_\mathbf{b}$ and $\mathbf{A}_{\mathbf{b}'}$ are invertible. Recall that we have the loss for a design proportion $\mathbf{b}$ as

$$\begin{aligned}
\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*) &= \frac{1}{n}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \mathbf{w}(a')\right) \overset{(a)}{=} \frac{1}{n}\mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \mathbf{w}(a')\right) \\
&= \frac{1}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top\right) \\
&= \frac{1}{n}\mathbf{Tr}\left(\mathbf{V}\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*}\right)
\end{aligned}$$

where, in (a) we can introduce the trace as the R.H.S. is a scalar quantity, $\mathbf{w}(a) = \pi(a)\mathbf{x}(a)$ and $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$. Similarly for a $\lambda \in [0,1]$ we have

$$\mathcal{L}_n(\pi, \lambda\mathbf{b} + (1-\lambda)\mathbf{b}', \mathbf{\Sigma}_*) = \frac{1}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{b}',\mathbf{\Sigma}_*} \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top\right) = \frac{1}{n}\mathbf{Tr}\left(\mathbf{V}\mathbf{A}^{-1}_{\mathbf{b},\mathbf{b}',\mathbf{\Sigma}_*}\right).$$

Let the matrix $\mathbf{A}_{\mathbf{b},\mathbf{b}',\boldsymbol{\Sigma}_*}$ be defined as

$$\mathbf{A}_{\mathbf{b},\mathbf{b}',\boldsymbol{\Sigma}_*} \coloneqq \lambda \mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} + (1-\lambda)\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*}.$$

Now observe that

$$\mathbf{A}_{\mathbf{b},\mathbf{b}',\boldsymbol{\Sigma}_*} = \lambda \mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} + (1-\lambda)\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*} = \sum_{a=1}^{A} \left(\lambda \mathbf{b}(a) + (1-\lambda)\mathbf{b}'(a)\right) \widetilde{\mathbf{x}}(a)\widetilde{\mathbf{x}}(a)^\top.$$

Also observe that this is a positive semi-definite matrix. Now using Lemma 1 from (Whittle, 1958) we can show that

$$\left(\lambda \mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} + (1-\lambda)\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*}\right)^{-1} \prec \lambda \mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*} + (1-\lambda)\mathbf{A}^{-1}_{\mathbf{b}',\boldsymbol{\Sigma}_*}$$

for any positive semi-definite matrices $\mathbf{A}_{\mathbf{b}}, \mathbf{A}_{\mathbf{b}'}$, and $\lambda \in [0,1]$. Now taking the trace on both sides we get

$$\mathbf{Tr}\left(\lambda \mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} + (1-\lambda)\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*}\right)^{-1} \prec \mathbf{Tr}\lambda \mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*} + \mathbf{Tr}(1-\lambda)\mathbf{A}^{-1}_{\mathbf{b}',\boldsymbol{\Sigma}_*}.$$

Now using Lemma 2 from Whittle (1958) we can show that

$$\mathbf{Tr}\left(\lambda \mathbf{V}\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} + (1-\lambda)\mathbf{V}\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*}\right)^{-1} \prec \mathbf{Tr}\lambda \mathbf{V}\mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*} + \mathbf{Tr}(1-\lambda)\mathbf{V}\mathbf{A}^{-1}_{\mathbf{b}',\boldsymbol{\Sigma}_*}.$$

for any positive semi-definite matrix $\mathbf{V}$. This implies that

$$\mathcal{L}_n(\pi, \lambda\mathbf{b} + (1-\lambda)\mathbf{b}', \boldsymbol{\Sigma}_*) < \lambda \mathcal{L}_n(\pi, \mathbf{b}, \boldsymbol{\Sigma}_*) + (1-\lambda)\mathcal{L}_n(\pi, \mathbf{b}', \boldsymbol{\Sigma}_*).$$

Hence, the loss function is convex. □

Remark B.4. (***Bound on variance***) *We can use singular value decomposition of* $\boldsymbol{\Sigma}_*$ *as* $\boldsymbol{\Sigma}_* = \overline{\mathrm{U}}\mathbf{D}\mathbf{P}^\top$ *with orthogonal matrices* $\overline{\mathrm{U}}, \mathbf{P}^\top$ *and* $\mathbf{D} = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_d)$

where λ_i *denotes a singular value. Then we can bound* $\mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a)$ *as*

$$\left\|\mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a)\right\| = \left\|\mathbf{x}(a)^\top \overline{U} \mathbf{D} \mathbf{P}^\top \mathbf{x}(a)\right\| \overset{(a)}{=} \left\|\mathbf{u}^\top \mathbf{D} \mathbf{p}\right\| \leqslant \left\|\mathbf{u}^\top\right\| \max_i |\lambda_i| \, \|\mathbf{p}\|$$
$$\overset{(b)}{=} \|\mathbf{x}(a)\| \max_i |\lambda_i| \, \|\mathbf{x}(a)\| = \max_i |\lambda_i| \, \|\mathbf{x}(a)\|^2$$

where in (a) *we have* $\mathbf{u} = \overline{U}^\top \mathbf{x}(a)$, $\mathbf{p} = \mathbf{P}^\top \mathbf{x}(a)$ *and* (b) *uses the fact that* $\left\|\overline{U}^\top \mathbf{x}(a)\right\| = \|\mathbf{x}(a)\|$ *for any orthogonal matrix* $\overline{U}^\top$*. Similarly we can show that* $\left\|\mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a)\right\| \geqslant \min_i |\lambda_i| \, \|\mathbf{x}(a)\|^2$*. Let* $H_L^2 \leqslant \|\mathbf{x}(a)\|^2 \leqslant H_U^2$ *for any* $a \in [A]$*. This implies that*

$$\underbrace{\min_i |\lambda_i| H_L^2}_{\sigma^2_{\min}} \leqslant \min_i |\lambda_i| \, \|\mathbf{x}(a)\|^2 \leqslant \underbrace{\mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a)}_{\sigma^2(a)} \leqslant \max_i |\lambda_i| \, \|\mathbf{x}(a)\|^2 \leqslant \underbrace{\max_i |\lambda_i| H_U^2}_{\sigma^2_{\max}}$$

Loss Gradient is Bounded

Proposition 3. *Let* $\mathbf{b}, \mathbf{b}' \in \triangle(\mathcal{A})$*, so that* $\mathbf{A}_{\mathbf{b}, \boldsymbol{\Sigma}_*}$ *and* $\mathbf{A}_{\mathbf{b}', \boldsymbol{\Sigma}_*}$ *are invertible and define* $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a) \mathbf{w}(a')^\top$*. Then the gradient of the loss function is bounded such that*

$$\|\nabla_{\mathbf{b}(a)} \mathcal{L}(\pi, \mathbf{b}, \boldsymbol{\Sigma}_*) - \nabla_{\mathbf{b}(a)} \mathcal{L}(\pi, \mathbf{b}', \boldsymbol{\Sigma}_*)\|_2 \leqslant C_\kappa$$

where, the

$$C_\kappa = \frac{\lambda_d(\mathbf{V}) H_U^2}{\sigma^2(a) \left(\min_{a' \in \mathcal{A}} \frac{\mathbf{b}(a')}{\sigma(a')^2} \lambda_{\min} \left(\sum_{a=1}^{A} \mathbf{w}(a) \mathbf{w}(a)^\top\right)\right)^2}$$
$$+ \frac{\lambda_1(\mathbf{V}) H_U^2}{\sigma^2(a) \left(\min_{a' \in \mathcal{A}} \frac{\mathbf{b}'(a')}{\sigma(a')^2} \lambda_{\min} \left(\sum_{a=1}^{A} \mathbf{w}(a) \mathbf{w}(a)^\top\right)\right)^2}.$$

Proof. Let $\mathbf{b}, \mathbf{b}' \in \triangle(\mathcal{A})$, so that $\mathbf{A}_{\mathbf{b}, \boldsymbol{\Sigma}_*}$ and $\mathbf{A}_{\mathbf{b}', \boldsymbol{\Sigma}_*}$ are invertible. Observe

that the gradient of the loss is given by

$$\begin{aligned}\nabla_{\mathbf{b}(a)}\mathcal{L}(\pi,\mathbf{b},\boldsymbol{\Sigma}_*) &= \nabla_{\mathbf{b}(a)}\mathbf{Tr}\left(\sum_{a,a'}\mathbf{w}(a)^\top\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a')\right)\\ &\overset{(a)}{\leqslant} \lambda_1(\mathbf{V})\nabla_{\mathbf{b}(a)}\mathbf{Tr}(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1})\\ &= -\lambda_1(\mathbf{V})\mathbf{Tr}\left(\left(\frac{\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a)}\right)\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-2}\right)\\ &= -\lambda_1(\mathbf{V})\frac{1}{\sigma^2(a)}\left\|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2\end{aligned}$$

where, in (a) we denote $\mathbf{V} = \sum_{a,a'}\mathbf{w}(a)\mathbf{w}(a')^\top$. Similarly, the gradient of the loss is lower bounded by

$$\nabla_{\mathbf{b}(a)}\mathcal{L}(\pi,\mathbf{b},\boldsymbol{\Sigma}_*) \geqslant -\lambda_d(\mathbf{V})\frac{1}{\sigma^2(a)}\left\|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2$$

which yields a bound on the gradient difference as

$$\begin{aligned}&\|\nabla_{\mathbf{b}(a)}\mathcal{L}(\pi,\mathbf{b},\boldsymbol{\Sigma}_*) - \nabla_{\mathbf{b}'(a)}\mathcal{L}(\pi,\mathbf{b}',\boldsymbol{\Sigma}_*)\|_2\\ &\leqslant \left\|\lambda_d(\mathbf{V})\frac{1}{\sigma^2(a)}\left\|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2 - \lambda_1(\mathbf{V})\frac{1}{\sigma^2(a)}\left\|\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2\right\|_2\\ &\leqslant \left|\lambda_d(\mathbf{V})\frac{1}{\sigma^2(a)}\left\|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2\right| + \left|\lambda_1(\mathbf{V})\frac{1}{\sigma^2(a)}\left\|\mathbf{A}_{\mathbf{b}',\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2\right|.\end{aligned}$$

So now we focus on the quantity

$$\left\|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\mathbf{w}(a)\right\|_2^2 \leqslant \|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\|_2^2\|\mathbf{w}(a)\|_2^2 \leqslant \|\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\|_2^2 H_U^2.$$

Now observe that when $\mathbf{b}(a) \in \triangle(\mathcal{A})$ and initialized uniform randomly, then the optimization in (3.6) results in a non-singular $\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}$ if each action has been sampled at least once which is satisfied by SPEED. So now we need to bound the minimum eigenvalue of $\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}$ denoted as $\lambda_{\min}(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})$.

Using Lemma 7 of Fontaine et al. (2021) we have that for all $\mathbf{b} \in \triangle(\mathcal{A})$,

$$\min_{a \in [A]} \frac{\mathbf{b}(a)}{\sigma(a)^2} \sum_{a=1}^{A} \mathbf{w}(a)\mathbf{w}(a)^\top \preccurlyeq \sum_{a=1}^{A} \frac{\mathbf{b}(a)}{\sigma(a)^2} \mathbf{w}(a)\mathbf{w}(a)^\top.$$

And finally

$$\min_{a \in [A]} \frac{\mathbf{b}(a)}{\sigma(a)^2} \lambda_{\min} \left(\sum_{a=1}^{A} \mathbf{w}(a)\mathbf{w}(a)^\top \right) \leqslant \lambda_{\min}(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})$$

This implies that

$$\lambda_{\min}(\mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*}) \leqslant \frac{1}{\min_{a \in [A]} \frac{\mathbf{b}(a)}{\sigma(a)^2} \lambda_{\min} \left(\sum_{a=1}^{A} \mathbf{w}(a)\mathbf{w}(a)^\top \right)}$$

Plugging everything back we get that

$$\begin{aligned}
&\|\nabla_{\mathbf{b}(a)} \mathcal{L}(\pi, \mathbf{b}, \boldsymbol{\Sigma}_*) - \nabla_{\mathbf{b}'(a)} \mathcal{L}(\pi, \mathbf{b}', \boldsymbol{\Sigma}_*)\|_2 \\
&\leqslant \frac{\lambda_d(\mathbf{V}) H_U^2}{\sigma^2(a) \left(\min_{a' \in \mathcal{A}} \frac{\mathbf{b}(a')}{\sigma(a')^2} \lambda_{\min} \left(\sum_{a=1}^{A} \mathbf{w}(a)\mathbf{w}(a)^\top \right) \right)^2} \\
&\quad + \frac{\lambda_1(\mathbf{V}) H_U^2}{\sigma^2(a) \left(\min_{a' \in \mathcal{A}} \frac{\mathbf{b}'(a')}{\sigma(a')^2} \lambda_{\min} \left(\sum_{a=1}^{A} \mathbf{w}(a)\mathbf{w}(a)^\top \right) \right)^2}.
\end{aligned}$$

The claim of the lemma follows. □

Kiefer-Wolfowitz Equivalence

We now introduce a Kiefer-Wolfowitz type equivalence (Kiefer and Wolfowitz, 1960) for the quantity $\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*})$ for optimal $\mathbf{b}_* \in \Delta(\mathcal{A})$ and covariance matrix $\boldsymbol{\Sigma}_*$ in Proposition 4.

Proposition 4. (***Kiefer-Wolfowitz for PE-Optimal***) *Define the heteroscedas-*

tic design matrix as $\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} = \sum_{a=1}^{A} \mathbf{b}(a)\widetilde{\mathbf{x}}(a)\widetilde{\mathbf{x}}(a)^\top$. *Assume that* $\mathcal{A} \subset \mathbb{R}^d$ *is compact and* $\operatorname{span}(\mathcal{A}) = \mathbb{R}^d$. *Then the following are equivalent:*

(*a*) $\mathbf{b}_*$ *is a minimiser of* $\widetilde{\mathbf{g}}(\mathbf{b}, \boldsymbol{\Sigma}_*) = \mathbf{Tr}\left(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\right)$.

(*b*) $\mathbf{b}_*$ *is a maximiser of* $f(\mathbf{b}, \boldsymbol{\Sigma}_*) = \log\det\left(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}\right)$.

(*c*) $\widetilde{\mathbf{g}}\left(\mathbf{b}_*, \boldsymbol{\Sigma}_*\right) = d$.

Furthermore, there exists a minimiser $\mathbf{b}_*$ *of* $\widetilde{\mathbf{g}}(\mathbf{b}, \boldsymbol{\Sigma}_*)$ *such that* $|\operatorname{Supp}\left(\mathbf{b}_*\right)| \leqslant d(d+1)/2$.

Proof. We follow the proof technique of Lattimore and Szepesvári (2020a). Let $\mathbf{b} : \mathcal{A} \to [0,1]$ be a distribution on $\mathcal{A}$ so that $\sum_{a\in\mathcal{A}} \mathbf{b}(a) = 1$ and $\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} \in \mathbb{R}^{d\times d}$ and $g(\mathbf{b}) \in \mathbb{R}$ be given by

$$\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*} = \sum_{a=1}^{A} \mathbf{b}(a)\pi^2(a)\sigma^{-2}(a)\, \mathbf{x}(a)\mathbf{x}(a)^\top = \sum_{a=1}^{A} \mathbf{b}(a)\frac{\pi(a)\mathbf{x}(a)}{\sigma(a)}\left(\frac{\pi(a)\mathbf{x}(a)}{\sigma(a)}\right)^\top$$

where, (a) follows by setting $\widetilde{\mathbf{x}}(a) = \mathbf{x}(a)/\sigma(a)$. First recall that for a square matrix $\mathbf{A}$ let $\operatorname{adj}(\mathbf{A})$ be the transpose of the cofactor matrix of $\mathbf{A}$. Use the facts that the inverse of a matrix $\mathbf{A}$ is $\mathbf{A}^{-1} = \operatorname{adj}(\mathbf{A})^\top/\det(\mathbf{A})$ and that if $\mathbf{A} : \mathbb{R} \to \mathbb{R}^{d\times d}$, then

$$\frac{d}{dt}\det(\mathbf{A}(t)) = \mathbf{Tr}\left(\operatorname{adj}(\mathbf{A})\frac{d}{dt}\mathbf{A}(t)\right).$$

It follows then that

$$\begin{aligned}\nabla f(\mathbf{b}, \boldsymbol{\Sigma}_*)_{b(a)} &\overset{(a)}{=} \frac{\mathbf{Tr}\left(\operatorname{adj}(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})\widetilde{\mathbf{x}}(a)\widetilde{\mathbf{x}}(a')^\top\right)}{\det(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})} \\ &= \frac{\widetilde{\mathbf{x}}(a)^\top \operatorname{adj}(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})\widetilde{\mathbf{x}}(a')}{\det(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})} \overset{(b)}{=} \widetilde{\mathbf{x}}(a)^\top \mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}^{-1}\widetilde{\mathbf{x}}(a') = \widetilde{\mathbf{g}}(\mathbf{b})\end{aligned}$$

where, in (a) we show the a-th component of $f(\mathbf{b})$ when we differentiate w.r.t to $\mathbf{b}(a)$, and (b) follows as $\frac{\text{adj}(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})}{\det(\mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*})} = \mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*}$. Also observe that

$$\left(\sum_{a=1}^{A} \mathbf{b}(a)\|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*}}\right) = \mathbf{Tr}\left(\sum_{a=1}^{A} \mathbf{b}(a)\widetilde{\mathbf{x}}(a)\widetilde{\mathbf{x}}(a')^\top \mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*}\right) = d. \tag{B.1}$$

Hence, $\max_{\mathbf{b}} \log\det \mathbf{A}_{\mathbf{b},\boldsymbol{\Sigma}_*}$ is lower bounded by d as in average we have that $\left(\sum_{a=1}^{A} \mathbf{b}(a)\|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*}}\right) = d$.

$(b) \Rightarrow (a)$: Suppose that $\mathbf{b}_*$ is a maximiser of f. By the first-order optimality criterion, for any $\mathbf{b}$ distribution on $\mathcal{A}$,

$$\begin{aligned}
0 &\geqslant \langle \nabla f(\mathbf{b}_*, \boldsymbol{\Sigma}_*), \mathbf{b} - \mathbf{b}_* \rangle \\
&\geqslant \left(\sum_{a=1}^{A} \mathbf{b}(a)\|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}} - \sum_{a=1}^{A} \mathbf{b}_*(a)\|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}}\right) \\
&\geqslant \left(\sum_{a=1}^{A} \mathbf{b}(a)\|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}} - d\right).
\end{aligned}$$

For an arbitrary $a \in \mathcal{A}$, choosing $\mathbf{b}$ to be the Dirac at $a \in \mathcal{A}$ proves that $\sum_{a=1}^{A} \|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}} \leqslant d$. Since $\widetilde{\mathbf{g}}(\mathbf{b}) \geqslant d$ for all $\mathbf{b}$ by (B.1), it follows that $\mathbf{b}_*$ is a minimiser of $\widetilde{\mathbf{g}}$ and that $\min_{\mathbf{b}} \widetilde{\mathbf{g}}(\mathbf{b}) = d$.

$(c) \Longrightarrow (b)$: Suppose that $\widetilde{\mathbf{g}}(\mathbf{b}_*) = d$. Then, for any $\mathbf{b}$,

$$\langle \nabla f(\mathbf{b}_*, \boldsymbol{\Sigma}_*), \mathbf{b} - \mathbf{b}_* \rangle = \left(\sum_{a=1}^{A} \mathbf{b}(a)\|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}} - d\right) \leqslant 0.$$

And it follows that $\mathbf{b}_*$ is a maximiser of f by the first-order optimality conditions and the concavity of f. This can be shown as follows:

Let $\mathbf{b}$ be a Dirac at a and $\mathbf{b}(t) = \mathbf{b}_* + t(\mathbf{b}_* - \mathbf{b})$. Since $\mathbf{b}_*(a) > 0$ it follows for sufficiently small $t > 0$ that $\mathbf{b}(t)$ is a distribution over $\mathcal{A}$.

Because $\mathbf{b}_*$ is a minimiser of f,

$$0 \geqslant \frac{d}{dt} f(\mathbf{b}(t), \boldsymbol{\Sigma}_*)\Big|_{t=0} = \langle \nabla f(\mathbf{b}_*, \boldsymbol{\Sigma}_*), \mathbf{b}_* - \mathbf{b} \rangle = d - \sum_{a=1}^{A} \|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*}}.$$

We now show (a) $\Longrightarrow$ (c). To prove the second part of the theorem, let $\mathbf{b}_*$ be a minimiser of $\widetilde{\mathbf{g}}$, which by the previous part is a maximiser of f. Let $S = \text{Supp}(\mathbf{b}_*)$, and suppose that $|S| > d(d+1)/2$. Since the dimension of the subspace of $d \times d$ symmetric matrices is $d(d+1)/2$, there must be a non-zero function $\nu : \mathcal{A} \to \mathbb{R}$ with $\text{Supp}(\nu) \subseteq S$ such that

$$\sum_{a \in S} \nu(a) \widetilde{\mathbf{x}}(a) \widetilde{\mathbf{x}}(a)^\top = \mathbf{0}. \tag{B.2}$$

Notice that for any $\widetilde{\mathbf{x}}(a) \in S$, the first-order optimality conditions ensure that $\sum_{a=1}^{A} \|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}} = d$. Hence

$$d \sum_{a \in S} \nu(a) = \sum_{a \in S} \nu(a) \|\widetilde{\mathbf{x}}(a)\|^2_{\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}} = 0,$$

where the last equality follows from (B.2). Let $\mathbf{b}(t) = \mathbf{b}_* + t\nu$ and let $\tau = \max\{t > 0 : \mathbf{b}(t) \in \mathcal{P}_\mathcal{A}\}$, which exists since $\nu \neq 0$ and $\sum_{a \in S} \nu(a) = 0$ and $\text{Supp}(\nu) \subseteq S$. By (B.2), $\mathbf{A}_{\mathbf{b}(t),\boldsymbol{\Sigma}_*} = \mathbf{A}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}$, and hence $f(\mathbf{b}(\tau), \boldsymbol{\Sigma}_*) = f(\mathbf{b}_*, \boldsymbol{\Sigma}_*)$, which means that $\mathbf{b}(\tau)$ also maximises f. The claim follows by checking that $|\text{Supp}(\mathbf{b}(T))| < |\text{Supp}(\mathbf{b}_*)|$ and then using induction. □

Corollary B.5. *1 From Proposition 4 we know that $\mathbf{b}_*$ is a minimizer for $\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b},\boldsymbol{\Sigma}_*})$ and $\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}) = d$. This implies that the loss is bounded at $\mathbf{b}_*$ as $\frac{\lambda_d(\mathbf{V})\, d}{n} \leqslant \mathcal{L}_n(\pi, \mathbf{b}_*, \boldsymbol{\Sigma}_*) \leqslant \frac{\lambda_1(\mathbf{V})\, d}{n}$ where $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$.*

Proof. First recall that we can rewrite the loss for any arbitrary proportion

$\mathbf{b}$ and co-variance $\mathbf{\Sigma}_*$ as

$$\mathcal{L}_n(\pi, \mathbf{b}, \mathbf{\Sigma}_*) = \frac{1}{n}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \mathbf{w}(a')\right) = \frac{1}{n}\left(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top\right)$$
$$= \frac{1}{n}\left(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*} \mathbf{V}\right).$$

From (Fang et al., 1994) we know that for any positive semi-definite matrices $\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*}$ and $\mathbf{V}$ we have that

$$\lambda_d(\mathbf{V})\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*}) \leqslant \mathbf{Tr}(\mathbf{V}\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*}) \leqslant \lambda_1(\mathbf{V})\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*})$$

where $\lambda_i(\mathbf{V})$ is the i th largest eigenvalue of $\mathbf{V}$. Now from Proposition 4 we know that for $\mathbf{b}_*$ is a minimizer for $\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b},\mathbf{\Sigma}_*})$ and $\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*}) = d$. This implies that the loss is bounded at $\mathbf{b}_*$ as

$$\lambda_d(\mathbf{V})\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*}) \leqslant \mathbf{Tr}(\mathbf{V}\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*}) \leqslant \lambda_1(\mathbf{V})\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*})$$
$$\implies \frac{\lambda_d(\mathbf{V})d}{n} \leqslant \mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \leqslant \frac{\lambda_1(\mathbf{V})d}{n}.$$

The claim of the corollary follows. □

Remark B.6. *Note that the estimator $\widehat{\theta}_n$ is an unbiased estimator of θ_*. Recall that*

$$\widehat{\theta}_n \coloneqq \arg\min_{\theta} \sum_{t=1}^{n} \tfrac{1}{\sigma^2(a_t)}(r_t - \mathbf{x}(a_t)^\top \theta)^2$$

where, a_t is the action sampled at timestep t. Define the

$$\mathbf{diag}(\mathbf{\Sigma}_n) = [\sigma^2(a_1), \sigma^2(a_2), \ldots, \sigma^2(a_n)],$$

$\mathbf{R}_n = [r_1, r_2, \ldots, r_n]^\top \in \mathbb{R}^{n\times 1}$ *be the n rewards observed and $\mathbf{J} \in \mathbb{R}^{n\times 1}$ is the noise vector, where $a_1, a_2, \ldots, a_n$ are the actions pulled at time $t = 1, 2, \ldots, n$.*

Then it can be shown that

$$\begin{aligned}
\mathbb{E}\left[\widehat{\theta}_n\right] - \theta_* &= \mathbb{E}\left[\left(\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n\right)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{R}_n\right] - \theta_* \\
&= \mathbb{E}\left[\left(\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n\right)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \left(\mathbf{X}_n \theta_* + \mathbf{J}\right)\right] - \theta_* \\
&= \mathbb{E}\left[\left(\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n\right)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n \theta_*\right] \\
&\qquad + \mathbb{E}\left[\left(\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n\right)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{J}\right] - \theta_* \\
&= \theta_* + \left(\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n\right)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbb{E}\left[\mathbf{J}\right] - \theta_* \overset{(a)}{=} 0
\end{aligned}$$

where, (a) *follows as noise is zero mean.*

B.1 Bandit Regret Proofs

Loss of Bandit Oracle

Proposition 5. **(*Bandit Oracle MSE*)** *Let the oracle sample each action* a *for* $\lceil n\mathbf{b}_*(a)\rceil$ *times, where* $\mathbf{b}_*$ *is the solution to* (3.3)*. Define* $\lambda_1(\mathbf{V})$ *as the maximum eigenvalue of* $\mathbf{V} := \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$*. Then the loss satisfies*

$$\mathcal{L}_n^*(\pi, \mathbf{b}_*, \boldsymbol{\Sigma}_*) \leqslant O_{\kappa^2, H_U^2}\left(\frac{d\lambda_1(V)\log n}{n}\right) + O_{\kappa^2, H_U^2}\left(\frac{1}{n}\right).$$

Proof. Recall the matrix $\mathbf{X}_n = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$ are the observed features for the n samples taken. Let $\mathbf{R}_n = [r_1, r_2, \ldots, r_n]^\top \in \mathbb{R}^{n\times 1}$ be the n rewards observed and $\mathbf{J} \in \mathbb{R}^{n\times 1}$ is the noise vector. Then using weighted least square estimates we have

$$\widehat{\theta}_n := \arg\min_{\theta} \sum_{t=1}^{n} \frac{1}{\sigma^2(a_t)}(r_t - \mathbf{x}(a_t)^\top \theta)^2$$

where, in (a) we a_t is the action sampled at timestep t. Recall that the $\mathbf{diag}(\boldsymbol{\Sigma}_n) = [\sigma^2(a_1), \sigma^2(a_2), \ldots, \sigma^2(a_n)]$, where $a_1, a_2, \ldots, a_n$ are the ac-

tions pulled at time $t = 1, 2, \ldots, n$. We have that:

$$\widehat{\theta}_n - \theta_* = (\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{J}$$

where the noise vector $\eta \sim \mathcal{SG}(0, \boldsymbol{\Sigma}_n)$ where $\boldsymbol{\Sigma}_n \in \mathbb{R}^{n \times n}$. For any $\mathbf{z} \in \mathbb{R}^d$ we have

$$\mathbf{z}^\top (\widehat{\theta}_n - \theta_*) = \mathbf{z}^\top (\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n)^{-1} \mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{J}.$$

Let $\mathbf{b}_*$ be the PE-Optimal design for $\mathcal{A}$ defined in (3.3). Then the oracle pulls action $a \in \mathcal{A}$ exactly $\lceil n\mathbf{b}_* \rceil$ times for some $n > d(d+1)/2$ and computes the least square estimator $\widehat{\theta}_n$. Observe that

$$\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_n - \theta_*) \sim \mathcal{SG}\left(0, \sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n)^{-1} \mathbf{w}(a')\right).$$

So $\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_n - \theta_*)\right)^2 \sim \mathcal{SE}\left(0, \sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n)^{-1} \mathbf{w}(a')\right)$ where $\mathcal{SE}$ denotes the sub-exponential distribution. Denote the quantity

$$t := \sqrt{2 \sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n)^{-1} \mathbf{w}(a') \log(1/\delta)}.$$

Now using sub-exponential concentration inequality in Theorem B.2, setting

$$\nu^2 = \sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{X}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{X}_n)^{-1} \mathbf{w}(a'),$$

and $\alpha = \nu$, we can show that

$$\mathbb{P}\left(\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\right)^2 > t\right) \leqslant \delta, \qquad \text{if } t \in (0,1]$$
$$\mathbb{P}\left(\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\right)^2 > t^2\right) \leqslant \delta, \qquad \text{if } t > 1.$$

Combining the above two we can show that

$$\mathbb{P}\left(\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\right)^2 > \min\{t, t^2\}\right) \leqslant \delta, \forall t > 0.$$

Further define matrix $\overline{\mathbf{\Sigma}}_n \in \mathbb{R}^{d\times d}$ as $\overline{\mathbf{\Sigma}}_n^{-1} \coloneqq (\mathbf{X}_n^\top \mathbf{\Sigma}_n^{-1} \mathbf{X}_n)^{-1}$. This means that we have with probability $(1-\delta)$ that

$$\begin{aligned}
&\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\right)^2 \\
&\leqslant \min\left\{\sqrt{2\sum_{a,a'} \mathbf{w}(a)^\top \overline{\mathbf{\Sigma}}_n^{-1} \mathbf{w}(a') \log(1/\delta)}, 2\sum_{a,a'} \mathbf{w}(a)^\top \overline{\mathbf{\Sigma}}_n^{-1} \mathbf{w}(a') \log(1/\delta)\right\} \\
&\overset{(a)}{=} \min\left\{\sqrt{\frac{2}{n}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a') \log(1/\delta)}, \right. \\
&\qquad \left. \frac{2}{n}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a') \log(1/\delta)\right\} \\
&\overset{(b)}{\leqslant} \min\left\{\sqrt{\frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}}, \frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}\right\}
\end{aligned}$$

and we have taken at most n pulls such that $n > \frac{d(d+1)}{2}$ pulls. Here (a) follows as $n\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*} = \overline{\mathbf{\Sigma}}_n$ and observing that oracle has access to $\mathbf{\Sigma}_*$, and optimal proportion $\mathbf{b}_*$. The (b) follows from applying Theorem B.5 such

that $\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*} \mathbf{w}(a') \leqslant d\lambda_1(\mathbf{V})$ where $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$. Thus, for any $\delta \in (0,1)$ we have

$$\mathbb{P}\left(\left\{\left(\sum_{a=1}^{A} \widetilde{\mathbf{x}}(a)^\top(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*)\right)^2 \right. \tag{B.3}$$

$$\left. > \min\left\{\sqrt{\frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}}, \frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}\right\}\right\}\right) \leqslant \delta. \tag{B.4}$$

Define the good event $\xi_\delta(n)$ as follows:

$$\xi_\delta(n) \coloneqq \left\{\left(\sum_{a=1}^{A} \widetilde{\mathbf{x}}(a)^\top(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*)\right)^2 \leqslant \min\left\{\sqrt{\frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}}, \frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}\right\}\right\}.$$

Then the loss of the oracle following PE-Optimal $\mathbf{b}_*$ is given by

$$\begin{aligned}
\mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) &= \mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_n - \theta_*\right)\right)^2\right] \\
&\leqslant \mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_n - \theta_*\right)\right)^2 \xi_\delta(n)\right] \\
&\qquad + \mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_n - \theta_*\right)\right)^2 \xi_\delta^c(n)\right] \\
&\overset{(a)}{\leqslant} \mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_n - \theta_*\right)\right)^2 \xi_\delta(n)\right] + \sum_{t=1}^{n} A H_U^2 \kappa^2 \mathbb{P}(\xi_\delta^c(n)) \\
&\overset{(b)}{\leqslant} \min\left\{\sqrt{\frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}}, \frac{8d\lambda_1(\mathbf{V})\log(1/\delta)}{n}\right\} + \sum_{t=1}^{n} A H_U^2 \kappa^2 \mathbb{P}(\xi_\delta^c(n)) \\
&\overset{(c)}{\leqslant} \min\left\{\sqrt{\frac{16d\lambda_1(\mathbf{V})\log n}{n}}, \frac{16d\lambda_1(\mathbf{V})\log n}{n}\right\} + O_{\kappa^2, H_U^2}\left(\frac{1}{n}\right) \\
&\leqslant \frac{48d\lambda_1(\mathbf{V})\log n}{n} + O_{\kappa^2, H_U^2}\left(\frac{1}{n}\right)
\end{aligned}$$

where, (a) follows as the noise $\eta^2 \leqslant \kappa^2$ and $\sum_a \|\mathbf{x}(a)\|^2 \leqslant A H_U^2$ which implies

$$\mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_n - \theta_*\right)\right)^2\right] \leqslant n A H_U^2 \kappa^2.$$

The (b) follows from (B.4), and (c) follows by setting $\delta = 1/n^3$, and noting that $n > A$. □

OLS-WLS Concentration Lemma

Lemma B.7. (***Concentration Lemma***) *After* Γ *samples of exploration, we can show that* $\mathbb{P}\left(\xi_\delta^{var}(\Gamma)\right) \geqslant 1 - 8\delta$ *where,* $C > 0$ *is a constant.*

Proof. We observed $(\mathbf{x}_t, r_t) \in \mathbb{R}^d \times \mathbb{R}, i = 1, \ldots, \Gamma$ from the model

$$r_t = \mathbf{x}_t^\top \theta_* + \eta_t, \tag{B.5}$$

$$\eta_t \sim \mathcal{SG}(0, \mathbf{x}_t^\top \boldsymbol{\Sigma}_* \mathbf{x}_t), \tag{B.6}$$

where $\theta_* \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_* \in \mathbb{R}^{d\times d}$ are unknown.

Given an initial estimate $\widehat{\theta}_\Gamma$ of θ_*, we first compute the squared residual $y_t := \left(\mathbf{x}_t^\top \widehat{\theta}_\Gamma - r_t\right)^2$, and then obtain an estimate of $\boldsymbol{\Sigma}_*$ via

$$\min_{\mathbf{S}\in\mathbb{R}^{d\times d}} \sum_{t=1}^{\Gamma} \left(\left\langle \mathbf{x}_t\mathbf{x}_t^\top, \mathbf{S}\right\rangle - y_t\right)^2. \tag{B.7}$$

Observe that if $\widehat{\theta}_\Gamma = \theta_*$, then the expectation of the squared residual y_t is

$$\mathbb{E}\left[y_t\right] = \mathbb{E}\left[\left(\mathbf{x}_t^\top \theta_* - r_t\right)^2\right] = \mathbb{E}\left[\eta_t^2\right] = \mathbf{x}_t^\top \boldsymbol{\Sigma}_* \mathbf{x}_t = \left\langle \mathbf{x}_t\mathbf{x}_t^\top, \boldsymbol{\Sigma}_*\right\rangle,$$

which is a linear function of $\boldsymbol{\Sigma}_*$. The program (B.7) is thus a least square formulation for estimating $\boldsymbol{\Sigma}_*$.

Let $\mathbf{X}_t := \mathbf{x}_t\mathbf{x}_t^\top$. Below we abuse notation and view $\boldsymbol{\Sigma}_*, \widehat{\boldsymbol{\Sigma}}_\Gamma, \mathbf{X}_t, \mathbf{S}$ as vectors in $\mathbb{R}^{d^2}$ endowed with the trace inner product $\langle\cdot,\cdot\rangle$. Let $\mathbf{X} \in \mathbb{R}^{\Gamma\times d^2}$ have rows $\{\mathbf{X}_t\}$, and $y = (y_1, \ldots, y_\Gamma)^\top \in \mathbb{R}^\Gamma$. Suppose $\mathbf{x}_t$ can only take on M possible values from $\{\phi_1, \ldots, \phi_M\}$, so $\mathbf{X}_t \in \{\Phi_1, \ldots, \Phi_M\}$, where $\Phi_m := \phi_m\phi_m^\top$. Note that for the forced exploration setting we have $M = d < A$. Moreover, each value appears exactly Γ/M times. Then (B.7) can

be rewritten as

$$\min_{\mathbf{S}\in\mathbb{R}^{d^2}} \sum_{m=1}^{M} \sum_{t:\mathbf{X}_t=\Phi_m} (\langle \Phi_m, \mathbf{S}\rangle - y_t)^2 = \min_{\mathbf{S}\in\mathbb{R}^{d^2}} \sum_{m=1}^{M} \left(\langle \Phi_m, \mathbf{S}\rangle - \frac{1}{\Gamma/M} \sum_{t:\mathbf{X}_t=\Phi_m} y_t \right)^2 .$$

Let $z_m := \frac{1}{\Gamma/M} \sum_{t:\mathbf{X}_t=\Phi_m} y_t$. Then it becomes

$$\min_{\mathbf{S}\in\mathbb{R}^{d^2}} \sum_{m=1}^{M} (\langle \Phi_m, \mathbf{S}\rangle - z_m)^2 = \min_{\mathbf{S}\in\mathbb{R}^{d^2}} \|\Phi\mathbf{S} - z\|_2^2,$$

where $\Phi \in \mathbb{R}^{m\times d^2}$ has rows $\{\Phi_m\}$, and $z := (z_1, \ldots, z_m)^\top \in \mathbb{R}/M$. Note that $\{\Phi_m\}$ may or may not span $\mathbb{R}^{d^2}$. Observe that $\widehat{\mathbf{\Sigma}}_\Gamma$ be an optimal solution to the above problem. Then

$$\begin{aligned}
&\left\|\Phi(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\right\|_2^2 + \|\Phi\mathbf{\Sigma}_* - z\|_2^2 + 2\left\langle \Phi(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*), \Phi\mathbf{\Sigma}_* - z\right\rangle \\
&= \left\|\Phi\widehat{\mathbf{\Sigma}}_\Gamma - \Phi\mathbf{\Sigma}_* + \Phi\mathbf{\Sigma}_* - z\right\|_2^2 \\
&= \left\|\Phi\widehat{\mathbf{\Sigma}}_\Gamma - z\right\|_2^2 \leqslant \|\Phi\mathbf{\Sigma}_* - z\|_2^2 .
\end{aligned}$$

Hence, we can show that

$$\begin{aligned}
\left\|\Phi(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\right\|_2^2 &\leqslant -2\left\langle \Phi(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*), \Phi\mathbf{\Sigma}_* - z\right\rangle \\
&\overset{(a)}{\leqslant} 2\left\|\Phi(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\right\|_2 \|\Phi\mathbf{\Sigma}_* - z\|_2 .
\end{aligned}$$

where, (a) follows from Cauchy Schwarz inequality. So

$$\left\|\Phi(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\right\|_2 \leqslant 2\|\Phi\mathbf{\Sigma}_* - z\|_2 .$$

Observe that the RHS does not contain the $\widehat{\mathbf{\Sigma}}_\Gamma$ anymore. Note that the

m-th entry of $\Phi\Sigma_* - z$ is

$$\langle \Phi_m, \Sigma_* \rangle - z_m = \phi_m^\top \Sigma_* \phi_m - \frac{1}{\Gamma/M} \sum_{t: X_t = \Phi_m} y_t.$$

Let $\zeta_\Gamma \coloneqq \widehat{\theta}_\Gamma - \theta_*$ where $\widehat{\theta}_\Gamma$ is the estimation of θ_* after $\Gamma = \sqrt{n}$ rounds of exploration. The noise η_t is σ_t^2 sub-Gaussian. Then

$$\begin{aligned} y_t &= \left(x_t^\top \widehat{\theta}_t - r_t \right)^2 \\ &= (\eta_t + x_t^\top \zeta_\Gamma)^2 \\ &= \eta_t^2 + 2\eta_t x_t^\top \zeta_\Gamma + \left(x_t^\top \zeta_\Gamma \right)^2 \\ &= x_t^\top \Sigma_* x_t + \epsilon_t = \langle \Sigma_*, x_t x_t^\top \rangle + \epsilon_t \overset{(a)}{=} \langle \widetilde{\theta}_*, z_t \rangle + \epsilon_t \end{aligned}$$

where, in (a) we denote the $\widetilde{\theta}_* \in \mathbb{R}^{d^2}$ as the vector reshaping Σ_* and $z_t \in \mathbb{R}^{d^2}$ is the vector reshaping $x_t x_t^\top$. This shows that the feedback y_t is linear. Now we need to show that ϵ_t is sub-exponential. We proceed as follows: We have that

$$\epsilon_t \coloneqq y_t - x_t^\top \Sigma_* x_t = \underbrace{\eta_t^2 - \mathbb{E}[\eta_t^2]}_{\textbf{Part A}} + \underbrace{2\eta_t x_t^\top \zeta_\Gamma}_{\textbf{Part B}} + \underbrace{\left(x_t^\top \zeta_\Gamma \right)^2}_{\textbf{Part C}}$$

The goal is to prove that $\mathbb{P}(\epsilon_t > \mathbb{E}[\epsilon_t] + s) \leqslant \exp(-s/2\sigma_{\max}^2)$ for some $s \in \mathbb{R}$.

For part A, we know that the η_t^2 is a sub-exponential random variable with $\eta_t^2 \sim \mathcal{SE}(\nu, \alpha)$ where $\nu = 4\sigma_t^2\sqrt{2}, \alpha = 4\sigma_t^2$, and $\sigma_t^2 = x_t^\top \Sigma_* x_t$. This follows from Equation 37 in Appendix B of Honorio and Jaakkola (2014). It shows that if X is a centered sub-Gaussian random variable with sub-Gaussian parameter σ^2 then X^2 is sub-exponential with parameters $\nu = 4\sigma^2\sqrt{2}, \alpha = 4\sigma^2$.

From Theorem B.2 we know that

$$\begin{aligned}
&\mathbb{P}\left(\eta_t^2 \geqslant \mathbb{E}[\eta_t^2] + 8\sigma_t^2 \log(A/\delta)\right) \\
&\quad \leqslant \exp\left(-\frac{1}{2}\min\left\{\frac{8\sigma_t^2\log(A/\delta)}{4\sigma_t^2}, \frac{64\sigma_t^4\log^2(A/\delta)}{32\sigma_t^4}\right\}\right) \\
&\quad = \exp\left(-\log(A/\delta)\right).
\end{aligned}$$

Hence, $\eta_t^2 \leqslant 4\sigma_t^2\sqrt{2} + 8\sigma_t^2\log(A/\delta) \leqslant 16\sigma_{max}^2\log(A/\delta)$ with probability $(1-\delta)$ as $\mathbb{E}[\eta_t^2] = \nu$. Equivalently we can write that

$$\mathbb{P}\left(\eta_t^2 \geqslant \mathbb{E}[\eta_t^2] + 16\sigma_t^2\log(A/\delta)\right) \leqslant \exp\left(-\frac{s_1^2}{16\sigma_{max}^2}\right) = \exp\left(-\frac{s_1^2}{2c'\sigma_{max}^2}\right). \tag{B.8}$$

where $s_1 = \sqrt{2c'\sigma_{max}^2\log(A/\delta)}$ and $c' > 0$ is a constant.

For part C we proceed as follows:

$$\begin{aligned}
(\mathbf{x}_t^\top(\widehat{\theta}_\Gamma - \theta_*))^2 \leqslant (\mathbf{x}^\top\mathbf{x})\|\widehat{\theta}_\Gamma - \theta_*\|^2 \leqslant (\mathbf{x}^\top\mathbf{x})\frac{MSE(\mathbf{X}(\widehat{\theta}_\Gamma - \theta_*))}{\lambda_{min}} \\
\leqslant \frac{H_U^2}{\lambda_{min}\Gamma}\left(8\log(6)\sigma_{max}^2 r + 8\sigma_{max}^2\log(1/\delta)\right) \leqslant \frac{2c''\sigma_{max}^2 d^2\log(A/\delta)}{\Gamma}
\end{aligned}$$

The first inequality follows by Cauchy Schwarz and the second by Remark 2.3 of Rigollet and Hütter (2015). Therefore it follows that

$$\mathbb{P}\left((\mathbf{x}_t^\top(\widehat{\theta}_\Gamma - \theta_*))^2 \geqslant \frac{2c''\sigma_{max}^2 d^2\log(A/\delta)}{\Gamma}\right) \leqslant \delta.$$

Assuming $\Gamma > d^2$ we can also show that

$$\mathbb{P}\left((\mathbf{x}_t^\top(\widehat{\theta}_\Gamma - \theta_*))^2 \geqslant 2c''\sigma_{max}^2\log(A/\delta)\right) \leqslant \delta$$

which drops the dependence on Γ and d. Equivalently we can write that

$$\mathbb{P}\left((\mathbf{x}_t^\top(\widehat{\theta}_\Gamma - \theta_*))^2 \geqslant 2c'' \sigma_{\max}^2 \log(A/\delta)\right) \leqslant \exp\left(-\frac{s_3^2}{2c''\sigma_{\max}^2}\right). \tag{B.9}$$

where $s_3 = \sqrt{2c''\sigma_{\max}^2 \log(A/\delta)}$.

For part B we proceed as follows:

$$2\underbrace{\eta_t}_{a}\underbrace{\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right)}_{b} \overset{(a)}{\leqslant} 2\eta_t^2 + \frac{1}{2}\left(\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right)\right)^2$$

where, (a) follows as $2ab \leqslant 2a^2 + \frac{1}{2}b^2$. It follows then that

$$\begin{aligned}
&\mathbb{P}\left(2\eta_t \mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right) \geqslant s_1^2 + s_3^3\right) \\
&\overset{(a)}{\leqslant} \mathbb{P}\left(2\eta_t^2 + \frac{1}{2}(\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right))^2 > s_1^2 + s_3^2\right) \\
&\leqslant \mathbb{P}\left(2\eta_t^2 > s_1^2 + s_3^2\right) + \mathbb{P}\left(\frac{1}{2}(\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right))^2 > s_1^2 + s_3^2\right) \\
&= \mathbb{P}\left(\eta_t^2 > \frac{s_1^2 + s_3^2}{2}\right) + \mathbb{P}\left((\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right))^2 > 2(s_1^2 + s_3^2)\right) \\
&\overset{(b)}{\leqslant} \mathbb{P}\left(\eta_t^2 > \frac{s_1^2 + s_3^2}{2}\right) + \mathbb{P}\left((\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right))^2 > \frac{s_1^2 + s_3^2}{2}\right) \\
&\overset{(c)}{\leqslant} \mathbb{P}\left(\eta_t^2 > \frac{s_1^2}{2}\right) + \mathbb{P}\left((\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right))^2 > \frac{s_3^2}{2}\right) \\
&\overset{(d)}{\leqslant} \exp\left(-\frac{s_1^2}{c'\sigma_{\max}^2}\right) + \exp\left(-\frac{s_3^2}{c''\sigma_{\max}^2}\right)
\end{aligned}$$

where, (a) follows as LHS $2\eta_t^2 + \frac{1}{2}(\mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right))^2 > 2\eta_t \mathbf{x}_t^\top\left(\widehat{\theta}_\Gamma - \theta_*\right)$ (that is LHS is larger). The (b) follows as RHS $\frac{s_1^2+s_3^2}{2} < 2(s_1^2 + s_3^2)$ and (c) follows as RHS $\frac{s_1^2+s_3^2}{2} < \frac{s_1^2}{2}$ (that is RHS is smaller). The (d) follows from (B.8), and (B.9).

We now estimate the expectation of ϵ_t. Observe that for $\Gamma > d^2$ we

have that

$$\begin{aligned}\mathbb{E}_{\eta,\zeta}[\epsilon_t] &= \mathbb{E}_{\eta,\zeta}\left[\eta_t^2 - \mathbb{E}_\eta[\eta_t^2] + 2\eta_t \mathbf{x}_t^\top \zeta_\Gamma + \left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2\right] \\ &= \mathbb{E}_\eta[\eta_t^2] - \mathbb{E}_{\eta,\zeta}[\mathbb{E}_\eta[\eta_t^2]] + 2\mathbb{E}_{\eta,\zeta}[\eta_t \mathbf{x}_t^\top \zeta_\Gamma] + \mathbb{E}_\zeta[\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2] \\ &\geqslant 2\mathbb{E}_{\eta,\zeta}[\eta_t \mathbf{x}_t^\top \zeta_\Gamma] + \mathbb{E}_\zeta[\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2] \geqslant \frac{\sigma_{\max}^2 d}{\Gamma}.\end{aligned}$$

Similarly we can get an upper bound to $\mathbb{E}[\epsilon_t]$ for $\Gamma > d^2$ as follows:

$$\begin{aligned}\mathbb{E}_{\eta,\zeta}[\epsilon_t] &= \mathbb{E}_{\eta,\zeta}\left[\eta_t^2 - \mathbb{E}_\eta[\eta_t^2] + 2\eta_t \mathbf{x}_t^\top \zeta_\Gamma + \left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2\right] \\ &\overset{(a)}{\leqslant} 2\mathbb{E}_\eta[\eta_t^2] + \frac{1}{2}\mathbb{E}_\zeta[\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2] + \mathbb{E}_\zeta[\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2] \\ &\leqslant 2\mathbb{E}_\eta[\eta_t^2] + \frac{2c'' \sigma_{\max}^2 d^2 \log(A/\delta)}{\Gamma} \overset{(b)}{\leqslant} 16\sigma_{\max}^2 + \frac{2c'' \sigma_{\max}^2 d^2 \log(A/\delta)}{\Gamma}\end{aligned}$$

where, (a) follows for $2ab \leqslant 2a^2 + \frac{1}{2}b^2$, (b) follows as $\mathbb{E}[\eta_t^2] = \nu \leqslant 8\sigma_{\max}^2$.

Define $s^2 = (s_1^2 + s_3^2)$. Then combining Part A, B and C it follows that

$$\begin{aligned}\mathbb{P}(\epsilon_t \geqslant s^2) &= \mathbb{P}(\eta_t^2 + 2\eta_t \mathbf{x}_t^\top \zeta_\Gamma + \left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2 \geqslant s^2) \\ &\leqslant \mathbb{P}(\eta_t^2 \geqslant s^2) + \mathbb{P}(2\eta_t \mathbf{x}_t^\top \zeta_\Gamma \geqslant s^2) + \mathbb{P}(\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2 \geqslant s^2) \\ &\overset{(a)}{\leqslant} \mathbb{P}(\eta_t^2 \geqslant s_1^2) + \mathbb{P}(2\eta_t \mathbf{x}_t^\top \zeta_\Gamma \geqslant s_1^2 + s_3^2) + \mathbb{P}(\left(\mathbf{x}_t^\top \zeta_\Gamma\right)^2 \geqslant s_3^2) \\ &\leqslant 2\exp\left(-\frac{s_1^2}{c' \sigma_{\max}^2}\right) + 2\exp\left(-\frac{s_3^2}{c'' \sigma_{\max}^2}\right) \qquad \text{(B.10)}\end{aligned}$$

where (a) follows as RHS $s_1^2 < s^2$, and $s_3^2 < s^2$ (that is the RHS is smaller). Let there be some constant $C > 0$ such that $s^2/C < \max\{s_1^2/c', s_3^2/c''\}$. Then it follows that

$$\mathbb{P}(\epsilon_t \geqslant \mathbb{E}[\epsilon_t] + s^2) \overset{(a)}{\leqslant} \mathbb{P}\left(\epsilon_t \geqslant \frac{\sigma_{\max}^2 d}{\Gamma} + s^2\right) \leqslant \mathbb{P}(\epsilon_t \geqslant s^2) \leqslant 4\exp\left(-\frac{s^2}{C\sigma_{\max}^2}\right).$$

where, (a) as the RHS $\mathbb{E}[\epsilon_t] \geqslant \frac{\sigma_{\max}^2 d}{\Gamma}$ is smaller. This shows that ϵ_t is a sub-

exponential random variable using Theorem B.2. Then using Theorem B.2 and $\Gamma > d^2$ we can show that

$$\mathbb{P}\left(\epsilon_t \geqslant \mathbb{E}[\epsilon_t] + \frac{Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right) \leqslant 4\exp\left(-\frac{Cd^2\sigma_{\max}^2 \log(A/\delta)}{C\Gamma\sigma_{\max}^2}\right) \leqslant 4\frac{\delta}{A}.$$

This implies that $\epsilon_t \leqslant \mathbb{E}[\epsilon_t] + \frac{Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma} \leqslant 16\sigma_{\max}^2 + \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}$ with probability greater than $1 - 4\frac{\delta}{A}$.

Combining all of the steps above we can show that

$$\begin{aligned}
&\mathbb{P}\left(\langle \Phi_m, \boldsymbol{\Sigma}_* \rangle - z_m > \frac{d}{\sqrt{n}} \sum_{t:X_t = \Phi_m} \left(16\sigma_{\max}^2 + \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right)\right) \\
&\overset{(a)}{=} \mathbb{P}\left(\langle \Phi_m, \boldsymbol{\Sigma}_* \rangle - z_m > \left(16\sigma_{\max}^2 + \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right)\right) \\
&\overset{(b)}{\leqslant} \mathbb{P}\left(\langle \Phi_m, \boldsymbol{\Sigma}_* \rangle - z_m > \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right) \leqslant 4\delta/A,
\end{aligned}$$

where, (a) follows by setting $\Gamma = \sqrt{n}$ and $M = d < A$ and noting that the m-th row consist of $\sqrt{n}/d$ entries. The (b) follows as the Hence the above implies that

$$\mathbb{P}\left(\mathbf{x}(a)^\top \widehat{\boldsymbol{\Sigma}}_\Gamma \mathbf{x}(a) - \mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a) \geqslant \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right) \leqslant 4\delta/A.$$

Similarly, we can bound the other tail inequality as

$$\mathbb{P}\left(\mathbf{x}(a)^\top \widehat{\boldsymbol{\Sigma}}_\Gamma \mathbf{x}(a) - \mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a) \leqslant -\frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right) \leqslant 4\delta/A.$$

Hence we can show by union bounding over all actions $A > d$ that

$$\mathbb{P}\left(\forall a, \left|\mathbf{x}(a)^\top \left(\widehat{\boldsymbol{\Sigma}}_\Gamma - \boldsymbol{\Sigma}_*\right) \mathbf{x}(a)\right| \geqslant \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\Gamma}\right) \leqslant 2A\frac{4\delta}{A} = 8\delta.$$

The claim of the lemma follows. □

Lemma B.8. (***Operator Norm Concentration Lemma***) *We have that*

$$\mathbb{P}\left(\|\widehat{\boldsymbol{\Sigma}}_\Gamma - \boldsymbol{\Sigma}_*\| \geqslant \frac{2Cd^2\sigma_{\max}^2\lambda_{\min}^{-1}(\mathbf{Y})\log(A/\delta)}{\Gamma}\right) \leqslant 8\delta$$

for a constant $C > 0$.

Proof. Define the set of actions $\mathcal{Z}$ such that it has a span over $\mathbf{X}$ and $\mathbf{X}\mathbf{X}^\top$. Define the vector $\mathbf{y}(a) = \mathbf{x}(a)\mathbf{x}(a)^\top \in \mathbb{R}^{d^2}$. Also observe that $|\mathcal{Z}| = d^2$. Without loss of generosity, we assume that $\mathcal{Z} = \{1, 2, \ldots, d^2\}$. Now define the matrix $\mathbf{Y} \in \mathbb{R}^{d^2 \times d^2}$ such that

$$\mathbf{Y} = [\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(|\mathcal{Z}|)]$$

We further assume that the $\lambda_{\min}(\mathbf{Y}) > 0$. We already have from Theorem A.4 that

$$\begin{aligned}
&\mathbb{P}\left(\forall a \in \mathcal{A}, \left|\mathbf{x}(a)^\top(\widehat{\boldsymbol{\Sigma}}_\Gamma - \boldsymbol{\Sigma}_*)\mathbf{x}(a)\right| \leqslant \frac{2Cd^2\sigma_{\max}^2\log(A/\delta)}{\Gamma}\right) \geqslant 1 - 8\delta \\
\overset{(a)}{\implies} &\mathbb{P}\left(\forall a \in \mathcal{Z}, \left|\langle\widehat{\boldsymbol{\Sigma}}_\Gamma, \mathbf{y}(a)\rangle - \langle\boldsymbol{\Sigma}_*, \mathbf{y}(a)\rangle\right| \leqslant \frac{2Cd^2\sigma_{\max}^2\log(A/\delta)}{\Gamma}\right) \geqslant 1 - 8\delta.
\end{aligned}$$

where, (a) follows by the fact that $\mathcal{Z} \subset \mathcal{A}$. Now take an arbitrary vector $\mathbf{x}$ in unit ball such that $\|\mathbf{x}\|_2 \leqslant 1$. Now we define the vector $\mathbf{y} = \mathbf{x}\mathbf{x}^\top$ such that $\mathbf{y} \in \mathbb{R}^{d^2}$. Then following Assumption 4 we have that

$$\mathbf{x}\mathbf{x}^\top = \mathbf{y} = \sum_{a \in \mathcal{Z}} \alpha(a)\mathbf{y}(a) = \alpha\mathbf{Y} \overset{(a)}{\implies} \alpha = \mathbf{Y}^{-1}\mathbf{y}$$

where, in (a) we can take the inverse because $\lambda_{\min}(\mathbf{Y}) > 0$. Now we want

to bound

$$\begin{aligned}
\|\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*\| &= \left|\mathbf{x}^\top \left(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*\right)\mathbf{x}\right| = \left|\langle \widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*, \mathbf{y}\rangle\right| \\
&\leqslant \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \left\| \underbrace{\sum_a \alpha(a)}_{\alpha} \right\| \\
&= \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \|\mathbf{Y}^{-1}\mathbf{y}\| \\
&\leqslant \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \|\mathbf{Y}^{-1}\| \|\mathbf{x}\|^2 \\
&\leqslant \frac{2Cd^2\sigma_{max}^2 \lambda_{min}^{-1}(\mathbf{Y}) \log(A/\delta)}{\Gamma}.
\end{aligned}$$

The claim of the lemma follows. □

Corollary B.9. *For,* $n \geqslant 4C^2d^2\sigma_{max}^2 \log^2(A/\delta)/\sigma_{min}^2$*, we have that with probability at least* $1 - 8\delta$*, the following holds: for all action* a*,* $\dfrac{\sigma^2(a)}{\widehat{\sigma}_\Gamma^2(a)} \leqslant 1 + \frac{4Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 \Gamma}$.

Proof. From the Theorem A.4, we know that $\left|\mathbf{x}(a)^\top (\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\mathbf{x}(a)\right| \leqslant \frac{2Cd^2 \log(A/\delta)}{\Gamma}$ with probability at least $1 - 8\delta$. Hence we can show that

$$\begin{aligned}
&|\widehat{\sigma}_\Gamma^2(a) - \sigma^2(a)| \leqslant \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \\
\implies &\sigma^2(a) - \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \leqslant \widehat{\sigma}_\Gamma^2(a) \leqslant \sigma^2(a) + \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \\
\implies &1 - \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma^2(a)\Gamma} \leqslant \frac{\widehat{\sigma}_\Gamma^2(a)}{\sigma^2(a)} \leqslant 1 + \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma^2(a)\Gamma} \\
\implies &1 - \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2\Gamma} \leqslant \frac{\widehat{\sigma}_\Gamma^2(a)}{\sigma^2(a)} \leqslant 1 + \frac{2Cd^2\sigma_{max}^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2\Gamma} \\
\implies &\frac{1}{1 + \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2\Gamma}} \leqslant \frac{\sigma^2(a)}{\widehat{\sigma}_\Gamma^2(a)} \leqslant \frac{1}{1 - \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2\Gamma}}.
\end{aligned}$$

It follows then that

$$\frac{\sigma^2(a)}{\widehat{\sigma}^2_\Gamma(a)} \leqslant \frac{1}{1-\frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\sigma^2_{\min}\Gamma}} \overset{(a)}{\leqslant} 1+\frac{4Cd^2\sigma^2_{\max}\log(A/\delta)}{\sigma^2_{\min}\Gamma}$$

where, (a) follows for $x = \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\sigma^2_{\min}\Gamma}$ and

$$\frac{1}{1-x} \leqslant 1+2x \implies 1 \leqslant 1+x-2x^2 \implies x(1-2x) \geqslant 0$$

which holds for $x = \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\sigma^2_{\min}\Gamma} \leqslant \frac{1}{2}$. For $n \geqslant 4C^2d^2\sigma^2_{\max}\log^2(A/\delta)/\sigma^2_{\min}$ we can show that $x \leqslant \frac{1}{2}$. The claim of the corollary follows. □

Bounding the Loss of Algorithm 2

Proposition 6. (***Loss of Algorithm 2, formal***) *Let $\widehat{b}$ be the empirical PE-Optimal design followed by Algorithm 2 and it samples each action a as $\lceil n\widehat{b}(a)\rceil$ times. Then the MSE of Algorithm 2 for for $n \geqslant \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\sigma^2_{\min}\Gamma}$ is given by*

$$\overline{\mathcal{L}}_n(\pi,\widehat{b},\widehat{\Sigma}_\Gamma) \leqslant \underbrace{O_{\kappa^2,H_U^2}\left(\frac{d^3\lambda_1(\mathbf{V})\log n}{\sigma^2_{\min}n}\right)}_{\substack{\textit{PE-Optimal MSE}\\ \textit{and exploration error}}} + \underbrace{O_{\kappa^2,H_U^2}\left(\frac{d^2\lambda_1(\mathbf{V})\log n}{n^{3/2}}\right)}_{\textit{Approximation error}} + \underbrace{O_{\kappa^2,H_U^2}\left(\frac{1}{n}\right)}_{\textit{Failure event MSE}}.$$

Proof. Recall that the $\widehat{\Sigma}_\Gamma$ be the empirical co-variance after Γ timesteps. Then Algorithm 2 pulls each action $a \in \mathcal{A}$ exactly $\left\lceil (n-\Gamma)\widehat{b}(a)\right\rceil$ times for some $\sqrt{n} > A$ and computes the least squares estimator $\widehat{\theta}_n$. Recall that the estimate $\widehat{\theta}_n$ only uses the $(n-\Gamma)$ data sampled under $\widehat{b}$. Also recall we actually use $\widehat{\Sigma}_\Gamma$ as input for optimization problem (3.3), where $\Gamma = \sqrt{n}$.

We first define the good event $\xi_\delta(n-\Gamma)$ as follows:

$$\begin{aligned}\xi_\delta(n-\Gamma) &:= \Bigg\{ \left(\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2 \\ &\leqslant \min\left\{ \sqrt{\frac{(8d\lambda_1(\mathbf{V}) + \alpha_0 + \alpha)\log(1/\delta)}{n-\Gamma}}, \right. \\ &\qquad \left. \frac{(8d\lambda_1(\mathbf{V}) + \alpha_0 + \alpha)\log(1/\delta)}{n-\Gamma} \right\}\Bigg\}\end{aligned}$$

where, α_0, and α will be defined later. Also, define the good variance event as follows:

$$\xi_\delta^{var}(\Gamma) := \left\{ \forall a, \left| \mathbf{x}(a)^\top \left(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_* \right) \mathbf{x}(a) \right| < \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \right\}. \quad \text{(B.11)}$$

Then we can bound the loss of the SPEED as follows:

$$\begin{aligned}\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) &= \mathbb{E}_{\mathcal{D}} \left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_{n-\Gamma} - \theta_* \right) \right)^2 \right] \\ &= \mathbb{E}_{\mathcal{D}} \left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_{n-\Gamma} - \theta_* \right) \right)^2 \mathbb{I}\{\xi_\delta(n-\Gamma)\}\mathbb{I}\{\xi_\delta^{var}(\Gamma)\} \right] \\ &+ \mathbb{E}_{\mathcal{D}} \left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_{n-\Gamma} - \theta_* \right) \right)^2 \mathbb{I}\{\xi_\delta^c(n-\Gamma)\} \right] \\ &\quad + \mathbb{E}_{\mathcal{D}} \left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_{n-\Gamma} - \theta_* \right) \right)^2 \mathbb{I}\{(\xi_\delta^{var}(\Gamma))^c\} \right]. \quad \text{(B.12)}\end{aligned}$$

Now we bound the first term of the (B.12). Note that using weighted least

square estimates we have

$$\widehat{\theta}_{n-\Gamma} \overset{(a)}{=} \widehat{\theta}_n \coloneqq \arg\min_{\theta} \sum_{t=\Gamma+1}^{n} \frac{1}{\sigma^2(a_t)}(r_t - \mathbf{x}(a_t)^\top \theta)^2$$

where, in (a) we a_t is the action sampled at timestep t. Recall that the $\mathbf{diag}(\widehat{\mathbf{\Sigma}}_\Gamma) = [\widehat{\sigma}^2_\Gamma(a_1), \widehat{\sigma}^2_\Gamma(a_2), \ldots, \widehat{\sigma}^2_\Gamma(a_n)]$, where $a_1, a_2, \ldots, a_{n-\Gamma}$ are the actions pulled at time $t = \Gamma + 1, 2, \ldots, n$. We have that:

$$\begin{aligned}
\widehat{\theta}_{n-\Gamma} &= (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{R}_n \\
&= (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} (\mathbf{X}_{n-\Gamma}\theta_* + \eta) \\
\widehat{\theta}_{n-\Gamma} - \theta_* &= (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{J}
\end{aligned}$$

where the noise vector $\eta \sim \mathcal{SG}(0, \mathbf{\Sigma}_{n-\Gamma})$ where $\mathbf{diag}(\mathbf{\Sigma}_n) = [\sigma^2(a_1), \sigma^2(a_2), \ldots, \sigma^2(a_{n-\Gamma})]$. For any $\mathbf{z} \coloneqq \sum_a \mathbf{w}(a) \in \mathbb{R}^d$ we have

$$\mathbf{z}^\top(\widehat{\theta}_{n-\Gamma} - \theta_*) = \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{J}. \tag{B.13}$$

It implies from (B.13) that

$$\begin{aligned}
&\left(\mathbf{z}^\top(\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2 \\
&\sim \mathcal{SE}\left(0, \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbb{E}\left[\mathbf{J}\mathbf{J}^\top\right] \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma} (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z}\right)
\end{aligned} \tag{B.14}$$

where $\mathcal{SE}$ denotes the sub-exponential distribution. Hence to bound the quantity $\left(\mathbf{z}^\top(\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2$ we need to bound the variance. We first begin

by rewriting the loss function for $n \geqslant \frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\sigma_{\min}^2 \Gamma}$ as follows

$$
\begin{aligned}
&\mathbb{E}\left[\left(\mathbf{z}^\top(\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2\right] \\
&= \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbb{E}\left[\mathbf{y}\mathbf{y}^\top\right] \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma} (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z} \\
&\overset{(a)}{=} \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{\Sigma}_n \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma} (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z} \\
&= \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \mathbf{\Sigma}_n \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \mathbf{X}_{n-\Gamma} (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z} \\
&\overset{(b)}{=} \underbrace{\mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}}}_{\mathbf{m}^\top \in \mathbb{R}^{n-\Gamma}} \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \mathbf{\Sigma}_n \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \underbrace{\widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \mathbf{X}_{n-\Gamma} (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z}}_{\mathbf{m} \in \mathbb{R}^{n-\Gamma}} \\
&\overset{(c)}{\leqslant} \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1/2} \left(\left(1 + 2C_{\Gamma,\sigma_{\min}^2}(\delta)\right) \mathbf{I}_n\right) \cdot \\
&\qquad \widehat{\mathbf{\Sigma}}_\Gamma^{-1/2} \mathbf{X}_{n-\Gamma} (\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z} \\
&\overset{(d)}{=} \left(1 + 2C_{\Gamma,\sigma_{\min}^2}(\delta)\right) \mathbf{z}^\top(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{X}_{n-\Gamma})^{-1} \mathbf{z} \qquad \text{(B.15)}
\end{aligned}
$$

where, (a) follows as $\mathbb{E}\left[\mathbf{y}\mathbf{y}^\top\right] = \mathbf{\Sigma}_n$, in (b) $\mathbf{m}$ is a vector in $\mathbb{R}^{n-\Gamma}$. The (c) follows by first observing that

$$
\widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} \mathbf{\Sigma}_n \widehat{\mathbf{\Sigma}}_\Gamma^{-\frac{1}{2}} = \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{\Sigma}_n = \mathbf{diag}(\widehat{\mathbf{\Sigma}}_\Gamma^{-1} \mathbf{\Sigma}_n) = \left[\frac{\sigma^2(I_1)}{\widehat{\sigma}_\Gamma^2(I_1)}, \frac{\sigma^2(I_2)}{\widehat{\sigma}_\Gamma^2(I_2)}, \ldots, \frac{\sigma^2(I_n)}{\widehat{\sigma}_\Gamma^2(I_n)}\right].
$$

Then note that using Theorem B.9 we have

$$
\frac{\sigma^2(I_t)}{\widehat{\sigma}_\Gamma^2(I_t)} \leqslant 1 + 2 \cdot \underbrace{\frac{2Cd^2\sigma_{\max}^2 \log(A/\delta)}{\sigma_{\min}^2 \Gamma}}_{:= C_{\Gamma,\sigma_{\min}^2}(\delta)}
$$

for each $t \in [n]$, and (d) follows as $1 + 2C_{\Gamma,\sigma_{\min}^2}(\delta)$ is not a random variable. Let $\widehat{b}^*$ be the empirical PE-Optimal design returned by the approximator after it is supplied with $\widehat{\mathbf{\Sigma}}_\Gamma$. Now observe that the quantity of the samples

collected (following $\widehat{b}^*$) after exploration is as follows:

$$\left(\widetilde{\mathbf{X}}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \widetilde{\mathbf{X}}_{n-\Gamma}\right)^{-1} = \left(\sum_a \left\lceil (n-\Gamma)\widehat{b}^*(a)\widehat{\sigma}_\Gamma^{-2}(a)\right\rceil \mathbf{w}(a)\mathbf{w}(a)^\top\right)^{-1} = \frac{1}{n-\Gamma}\mathbf{A}_{\widehat{b}^*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1}.$$

Hence we use the loss function

$$\begin{aligned}\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) &:= \left(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta)\right) \mathbf{z}^\top (\widetilde{\mathbf{X}}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1} \widetilde{\mathbf{X}}_{n-\Gamma})^{-1}\mathbf{z} \\ &= \frac{\left(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta)\right)}{n-\Gamma} \sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\widehat{b}^*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a'). \qquad \text{(B.16)}\end{aligned}$$

Also recall that we define

$$\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) = \frac{1}{n}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a').$$

So to minimize the quantity $\mathbb{E}\left[\left(\sum_a \mathbf{w}(a)^\top(\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2\right]$ is minimizing the quantity $\frac{\left(1+2C_{\Gamma,\sigma^2_{\min}}(\delta)\right)}{n-\Gamma}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\widehat{b}^*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1}\mathbf{w}(a')$. Further recall that we can show that from Assumption 5 (approximation oracle) and Kiefer-Wolfowitz theorem in Theorem B.5 that for the proportion $\mathbf{b}_*$ and any arbitrary positive semi-definite matrix $\widehat{\mathbf{\Sigma}}_\Gamma$ the following holds

$$\begin{aligned}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a') &= \mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a')\right) \\ &= \mathbf{Tr}\Bigg(\mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \underbrace{\sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top}_{\mathbf{V}}\Bigg) \\ &= \mathbf{Tr}\left(\mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1}\mathbf{V}\right) \leqslant d\lambda_1(\mathbf{V}). \qquad \text{(B.17)}\end{aligned}$$

Then we can decompose the loss as follows:

$$\begin{aligned}\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) &= \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) + \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) \\ &= \underbrace{\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma)}_{\textbf{Approximation error}} + \underbrace{\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)}_{\textbf{Comparing two diff loss}} \\ &\quad + \mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma). \end{aligned} \tag{B.18}$$

For the approximation error we need access to an oracle (see Assumption 5) that gives ϵ approximation error. Then setting $\epsilon = \frac{1}{\sqrt{n}}$ we have that

$$\begin{aligned}&\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) \qquad \text{(B.19)} \\ &= \frac{\left(1 + 2C_{\Gamma, \sigma^2_{\min}}(\delta)\right)}{n - \Gamma} \underbrace{\mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a') - \sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right)}_{\epsilon} \\ &\overset{(a)}{\leqslant} O_{\kappa^2, H^2_U}\left(\frac{d^2 \sigma^2_{\max} \log(A/\delta)}{n^{3/2}}\right) \qquad \text{(B.20)}\end{aligned}$$

where, (a) follows by setting $\Gamma = \sqrt{n}$, $\epsilon = 1/\sqrt{n}$ and $C_{\Gamma, \sigma^2_{\min}}(\delta) = \frac{2C d^2 \sigma^2_{\max} \log(A/\delta)}{\sigma^2_{\min} \Gamma} = \frac{2C d^2 \sigma^2_{\max} \log(A/\delta)}{\sigma^2_{\min} \sqrt{n}}$. Let us define $\mathbf{K}_1 := \mathbf{Tr}(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a'))$, and $\mathbf{K}_2 := \mathbf{Tr}(\mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a'))$. For the second part of comparing the two losses we can show that

$$
\begin{aligned}
&\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) \\
&\quad = \frac{1}{(n-\Gamma)} \mathbf{Tr}\left(\left(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta)\right) K_1\right) - \frac{1}{n} K_2 \\
&\quad = \frac{(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta))\mathbf{K}_1}{n-\Gamma} - \frac{(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta))\mathbf{K}_2}{n-\Gamma} + \frac{(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta))\mathbf{K}_2}{n-\Gamma} - \frac{1}{n}\mathbf{K}_2 \\
&\quad = \frac{(1 + 2C_{\Gamma,\sigma^2_{\min}}(\delta))}{n-\Gamma}(\mathbf{K}_1 - \mathbf{K}_2) + \frac{2C_{\Gamma,\sigma^2_{\min}}(\delta)\mathbf{K}_2}{n-\Gamma} + \frac{1}{n-\Gamma}\mathbf{K}_2 - \frac{1}{n}\mathbf{K}_2 \\
&\quad \overset{(a)}{=} \frac{\Gamma}{n(n-\Gamma)} \underbrace{\mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b}^*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a') - \sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right)}_{\leqslant 0} \\
&\qquad + \frac{2C_{\Gamma,\sigma^2_{\min}}(\delta)}{n-\Gamma} \mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right) \qquad \text{(B.21)} \\
&\qquad + \frac{\Gamma}{n(n-\Gamma)} \mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right) \\
&\quad \overset{(b)}{\leqslant} O_{\kappa^2, H_U^2}\left(\frac{d^3 \sigma^2_{\max} \lambda_1(\mathbf{V}) \log(A/\delta)}{\sigma^2_{\min} n^{3/2}}\right) \qquad \text{(B.22)}
\end{aligned}
$$

where, (a) follows by substituting the definition of $\mathbf{K}_1$ and $\mathbf{K}_2$. The (b) follows by setting $\Gamma = \sqrt{n}$, $C_{\Gamma,\sigma^2_{\min}}(\delta) = \frac{2C d^2 \sigma^2_{\max} \log(A/\delta)}{\sigma^2_{\min}\Gamma} = \frac{2C d^2 \sigma^2_{\max} \log(A/\delta)}{\sigma^2_{\min}\sqrt{n}}$, and $\mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right) \leqslant d\lambda_1(\mathbf{V})$.

Now we combine all parts together in (B.18) using (B.17), (B.20) and (B.22). First we define the quantity

$$
\alpha := 2C_{\Gamma,\sigma^2_{\min}}(\delta) \mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right) + \frac{\Gamma}{n} \mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a')\right).
$$

It follows then that (B.18) can be written as

$$\begin{aligned}
&\frac{1+2C_{\Gamma,\sigma^2_{\min}}(\delta)}{n-\Gamma}\sum_{a,a'}\mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b},\widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a') \\
&\qquad\leqslant \underbrace{\frac{(1+2C_\Gamma(\delta))\epsilon}{(n-\Gamma)}}_{\textbf{Approximation error}} + \frac{\alpha}{n-\Gamma} + \frac{1}{n}\sum_{a,a'}\mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a') \\
&\implies (1+2C_{\Gamma,\sigma^2_{\min}}(\delta))\sum_{a,a'}\mathbf{w}(a)^\top \mathbf{A}^{-1}_{\widehat{b},\widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a') \\
&\qquad\leqslant \underbrace{(1+2C_\Gamma(\delta))\epsilon}_{\alpha_0} + \alpha + \frac{n-\Gamma}{n}\sum_{a,a'}\mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a') \\
&\qquad\overset{(a)}{\leqslant} \alpha_0 + \alpha + d\lambda_1(\mathbf{V})
\end{aligned} \tag{B.23}$$

where, (a) follows from Assumption 5, Theorem B.5 and (B.17). Also observe that from (B.14) we have that $\left(\sum_{a=1}^{A}\mathbf{w}(a)^\top(\widehat{\theta}_n - \theta_*)\right)^2$ is a sub-exponential random variable. Then using the sub-exponential concentration inequality we have with probability at least $1-\delta$

$$
\begin{aligned}
&\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2 \\
&\leqslant \min\left\{\sqrt{(1+2C_{\Gamma,\sigma^2_{\min}}(\delta))\sum_{a,a'} \mathbf{w}(a)\left(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1}\mathbf{X}_{n-\Gamma}\right)^{-1}\mathbf{w}(a')2\log(1/\delta)},\right. \\
&\qquad \left.(1+2C_\Gamma(\delta))\sum_{a,a'} \mathbf{w}(a)^\top \left(\mathbf{X}_{n-\Gamma}^\top \widehat{\mathbf{\Sigma}}_\Gamma^{-1}\mathbf{X}_{n-\Gamma}\right)^{-1}\mathbf{w}(a')2\log(1/\delta)\right\} \\
&= \min\left\{\frac{1}{\sqrt{n-\Gamma}}\sqrt{(1+2C_{\Gamma,\sigma^2_{\min}}(\delta))\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a')2\log(1/\delta)},\right. \\
&\qquad \left.\frac{(1+2C_{\Gamma,\sigma^2_{\min}}(\delta))}{n-\Gamma}\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b},\widehat{\mathbf{\Sigma}}_\Gamma}\mathbf{w}(a')2\log(1/\delta)\right\} \\
&\overset{(a)}{\leqslant} \min\left\{\sqrt{\frac{(8d\lambda_1(\mathbf{V})+\alpha_0+\alpha)\log(1/\delta)}{n-\Gamma}}, \frac{(8d\lambda_1(\mathbf{V})+\alpha_0+\alpha)\log(1/\delta)}{n-\Gamma}\right\}
\end{aligned}
$$

where, (a) follows from (B.23), and we have taken at most $n-\Gamma$ pulls to estimate $\widehat{\theta}_n$ after forced exploration and $\sqrt{n} > d$. Thus, for any $\delta \in (0,1)$ we have

$$
\begin{aligned}
\mathbb{P}\Bigg(\Bigg\{\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top (\widehat{\theta}_n - \theta_*)\right)^2 > \min\Bigg\{&\sqrt{\frac{(8d\lambda_1(\mathbf{V})+\alpha_0+\alpha)\log(1/\delta)}{n-\Gamma}}, \\
&\frac{(8d\lambda_1(\mathbf{V})+\alpha_0+\alpha)\log(1/\delta)}{n-\Gamma}\Bigg\}\Bigg\}\Bigg) \leqslant \delta. \qquad \text{(B.24)}
\end{aligned}
$$

This gives us a bound on the first term of (B.12). Combining everything in (B.12) we can bound the loss of the SPEED as follows:

$$
\begin{aligned}
\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) &\leqslant \mathbb{E}_{\mathcal{D}}\left[\left(\sum_{a=1}^{A} \mathbf{w}(a)^\top \left(\widehat{\theta}_{n-\Gamma} - \theta_*\right)\right)^2 \mathbb{I}\{\xi_\delta(n-\Gamma)\}\mathbb{I}\{\xi_\delta^{var}(\Gamma)\}\right] \\
&\quad + \sum_{t=1}^{n} A H_U^2 \eta^2 \mathbb{P}(\xi_\delta^c(n-\Gamma)) + \sum_{t=1}^{n} A H_U^2 \eta^2 \mathbb{P}\left((\xi_\delta^{var}(\Gamma))^c\right) \\
&\leqslant \min\left\{\frac{2Cd^2\log(A/\delta)}{\Gamma}, \sqrt{\frac{(8d\lambda_1(\mathbf{V}) + \alpha_0 + \alpha)\log(A/\delta)}{n-\Gamma}},\right. \\
&\quad \left.\frac{(8d\lambda_1(\mathbf{V}) + \alpha_0 + \alpha)\log(A/\delta)}{n-\Gamma}\right\} + \sum_{t=1}^{n} A H_U^2 \eta^2 \mathbb{P}(\xi_\delta^c(n-\Gamma)) \\
&\quad + \sum_{t=1}^{n} A H_U^2 \eta^2 \mathbb{P}\left((\xi_\delta^{var}(\Gamma))^c\right) \\
&\overset{(a)}{\leqslant} \min\left\{\frac{8Cd^2\sigma_{max}^2\log(nA)}{\sqrt{n}}, \sqrt{\frac{48(d\lambda_1(\mathbf{V}) + \alpha_0 + \alpha)\log(nA)}{n}},\right. \\
&\quad \left.\frac{48(d\lambda_1(\mathbf{V}) + \alpha_0 + \alpha)\log(nA)}{n}\right\} + O\left(\frac{1}{n}\right) \\
&\leqslant \frac{48d^2\sigma_{max}^2\lambda_1(\mathbf{V})\log(nA)}{n} + \frac{48\alpha\log(nA)}{n} + \frac{48\alpha_0\log(nA)}{n} + O\left(\frac{1}{n}\right) \\
&\overset{(b)}{\leqslant} \frac{48d^2\sigma_{max}^2\lambda_1(\mathbf{V})\log(nA)}{n} + \frac{144d\lambda_1(\mathbf{V})C_{\Gamma,\sigma_{min}^2}(\delta)\log(nA)}{n} \\
&\quad + \frac{48d\lambda_1(\mathbf{V})\Gamma\log(nA)}{n^{3/2}} + \frac{48\epsilon\log(nA)}{n} + O\left(\frac{1}{n}\right)
\end{aligned}
$$

where (a) follows as Proposition 6 and setting $\delta = 1/n^3$ and noting that $\sqrt{n} > d$. The (b) follows by setting $(1 + 2C_\Gamma(\delta))\epsilon$ and the definition of α. Recall that for $\Gamma = \sqrt{n}$ we have that $C_{\Gamma,\sigma_{min}^2}(\delta) = \frac{2Cd^2\sigma_{max}^2\log(A/\delta)}{\sigma_{min}^2\Gamma} = \frac{2Cd^2\sigma_{max}^2\log(A/\delta)}{\sigma_{min}^2\sqrt{n}}$. Then setting $\epsilon = 1/\sqrt{n}$ we can bound the loss of the

following PE-Optimal $\widehat{b}$ as

$$\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) \leqslant O_{\kappa^2, H_U^2}\left(\frac{d^3\sigma_{max}^2\lambda_1(\mathbf{V})\log(nA)}{\sigma_{min}^2 n}\right) + O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma_{max}^2\lambda_1(\mathbf{V})\log(nA)}{n^{3/2}}\right) + O_{\kappa^2, H_U^2}\left(\frac{1}{n}\right).$$

The claim of the proposition follows. □

Remark B.10. (***Discussion on loss***) *Observe that from Proposition 6 that the MSE for policy evaluation setting scales as* $O(\frac{d^3\log(n)}{n})$. *We contrast this result with Chaudhuri et al. (2017) who obtain a bound on the MSE* $\mathbb{E}_{\mathcal{D}}[\|\theta_* - \widehat{\theta}_n\|^2] \leqslant O(\frac{d\log(n)}{n})$ *in a related setting. Note that Chaudhuri et al. (2017) only considers the setting when* $\mathbf{\Sigma}_*$ *is rank* 1. *We make no such assumption and get an additional factor of* d *in our result due to exploration in* d^2 *dimension to estimate* $\mathbf{\Sigma}_*$. *Finally we get the scaling as* d^3 *due to* $\sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a') \leqslant d\lambda_1(\mathbf{V})$ *from Theorem B.5. Also observe that we estimate* $\mathbb{E}_{\mathcal{D}}[\sum_a \mathbf{w}(a)^\top(\theta_* - \widehat{\theta}_n)^2]$ *as opposed to* $\mathbb{E}_{\mathcal{D}}[\|\theta_* - \widehat{\theta}_n\|^2]$ *in Chaudhuri et al. (2017).*

Regret of Algorithm 2

Corollary B.11. *For,* $n \geqslant 16C^2d^4\sigma_{max}^4\log^2(A/\delta)/\sigma_{min}^4$ *we have that for all action* a, $|\widehat{\sigma}^2_\Gamma(a) - \sigma^2(a)| \leqslant \sigma^2_{min}/2$.

Proof. From the Theorem A.4, we know that $\left|\mathbf{x}(a)^\top(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_*)\mathbf{x}(a)\right| \leqslant \frac{2Cd^2\sigma^2_{max}\log(A/\delta)}{\Gamma}$ with probability $1 - 8\delta$. Hence we can show that

$$|\widehat{\sigma}^2_\Gamma(a) - \sigma^2(a)| \leqslant \frac{2Cd^2\sigma^2_{max}\log(A/\delta)}{\Gamma} = \frac{2Cd^2\sigma^2_{max}\log(A/\delta)}{\sqrt{n}} \overset{(a)}{\leqslant} \frac{2Cd^2\sigma^2_{max}\log(A/\delta)}{\sqrt{16C^2d^4\sigma^4_{max}\log^2(A/\delta)/\sigma^4_{min}}} = \frac{\sigma^2_{min}}{2},$$

where (a) follows for $n \geqslant 16C^2 d^4 \sigma_{\max}^4 \log^2(A/\delta)/\sigma_{\min}^4$. The claim of the corollary follows. □

Lemma B.12. (***Loss Concentration of design matrix***) *Let $\widehat{\boldsymbol{\Sigma}}_\Gamma$ be the empirical estimate of $\boldsymbol{\Sigma}_*$. Define $\mathbf{V} = \sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top$. We have that for any arbitrary proportion $\mathbf{b}$ the following*

$$\mathbb{P}\left(\left| \sum_{a,a'} \mathbf{w}(a)^\top (\mathbf{A}^{-1}_{\mathbf{b}_*, \widehat{\boldsymbol{\Sigma}}_\Gamma} - \mathbf{A}^{-1}_{\mathbf{b}_*, \boldsymbol{\Sigma}_*}) \mathbf{w}(a') \right| \leqslant \frac{2CB^* d^3 \log(A/\delta)}{\Gamma} \right) \geqslant 1 - \delta$$

where B^* is a problem-dependent quantity such that

$$B^* = \left(\left\| \mathbf{A}^{-1}_{\mathbf{b}_*, \boldsymbol{\Sigma}_*} \mathbf{w} \right\|^2 \left\| \sum_{a=1}^{A} \mathbf{b}_*(a) \mathbf{w}(a) \mathbf{w}(a)^\top H_U^2 \right\| \cdot \left\| \left(\sum_{a=1}^{A} \frac{\mathbf{b}_*(a) \mathbf{w}(a) \mathbf{w}(a)^\top}{\sigma^2(a) + \frac{2C d^2 \sigma_{\max}^2 \log(9H_U^2/\delta)}{\sqrt{n}}} \right)^{-1} \mathbf{w} \right\| \right)$$

and $C > 0$ is a universal constant.

Proof. We have the following

$$
\begin{aligned}
&\left| \sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \mathbf{w}(a') - \sum_{a,a'} \mathbf{w}(a)^\top \mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*} \mathbf{w}(a') \right| \\
&= \left| \underbrace{\sum_{a} \mathbf{w}(a)^\top}_{\mathbf{w}} \left(\mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} - \mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*} \right) \underbrace{\sum_{a} \mathbf{w}(a)}_{\mathbf{w}} \right| \\
&= \left| \mathbf{w}^\top \left(\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*} \left(\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*} - \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \right) \mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \right) \mathbf{w} \right| \\
&= \left| \underbrace{\mathbf{w}^\top (\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_*}}_{\mathbf{u}} \left(\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*} - \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \right) \underbrace{\mathbf{A}^{-1}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}) \mathbf{w}}_{\mathbf{v}} \right| \\
&= \left| \mathbf{u} \left(\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*} - \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \right) \mathbf{v} \right| \overset{(a)}{\leqslant} \|\mathbf{u}\| \underbrace{\left\| \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*} - \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma} \right\|}_{\Delta} \|\mathbf{v}\| \qquad \text{(B.25)}
\end{aligned}
$$

where, (a) follows by Cauchy-Schwarz inequality. Now observe that the vector $\mathbf{u} \in \mathbb{R}^d$ is a problem dependent quantity. We now bound the Δ in

(B.25) as follows

$$
\begin{aligned}
\Delta &= \left\| \sum_{a=1}^{A} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a)} - \sum_{a=1}^{A} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\mathbf{x}(a)^\top \widehat{\boldsymbol{\Sigma}}_\Gamma \mathbf{x}(a)} \right\| \\
&\overset{(a)}{=} \left\| \sum_{a} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a)} - \sum_{a} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\widehat{\sigma}^2_\Gamma(a)} \right\| \\
&= \left\| \sum_{a} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top \left(\frac{1}{\sigma^2(a)} - \frac{1}{\widehat{\sigma}^2_\Gamma(a)} \right) \right\| \\
&= \left\| \sum_{a} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top \left(\frac{\widehat{\sigma}^2_\Gamma(a) - \sigma^2(a)}{\widehat{\sigma}^2_\Gamma(a)\sigma^2(a)} \right) \right\| \\
&\overset{(b)}{\leqslant} \left\| \sum_{a} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top \left(\frac{\widehat{\sigma}^2_\Gamma(a) - \sigma^2(a)}{\sigma^4_{\min}} \right) \right\| \\
&= \left\| \frac{1}{\sigma^4_{\min}} \sum_{a} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top \left(\mathbf{x}(a)^\top \widehat{\boldsymbol{\Sigma}}_\Gamma \mathbf{x}(a) - \mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a) \right) \right\| \\
&= \frac{1}{\sigma^4_{\min}} \left\| \sum_{a=1}^{A} \underbrace{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}_{\textbf{Problem dependent quantity}} \underbrace{\left(\mathbf{x}(a)^\top \left(\widehat{\boldsymbol{\Sigma}}_\Gamma - \boldsymbol{\Sigma}_* \right) \mathbf{x}(a) \right)}_{\textbf{Random Quantity}} \right\|
\end{aligned}
$$

where, (a) follows $\widehat{\sigma}^2_\Gamma(a) = \mathbf{x}(a)^\top \widehat{\boldsymbol{\Sigma}}_\Gamma \mathbf{x}(a)$ and $\sigma^2(a) = \mathbf{x}(a)^\top \boldsymbol{\Sigma}_* \mathbf{x}(a)$, ad (b) follows from Theorem B.11. Now observe from Theorem B.8 that we can bound the quantity

$$
\|\widehat{\boldsymbol{\Sigma}}_\Gamma - \boldsymbol{\Sigma}_*\| \leqslant \frac{2Cd^2\sigma^2_{\max}\lambda^{-1}_{\min}(\mathbf{Y})\log(A/\delta)}{\Gamma}.
$$

Then we also have that the spread of maximum eigenvalue of $\|\widehat{\boldsymbol{\Sigma}}_\Gamma - \boldsymbol{\Sigma}_*\|_2$

is controlled which implies

$$\frac{1}{\sigma_{\min}^4}\left\|\sum_{a=1}^{A}\underbrace{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}_{\textbf{Problem dependent quantity}}\underbrace{\left(\mathbf{x}(a)^\top\left(\widehat{\boldsymbol{\Sigma}}_\Gamma-\boldsymbol{\Sigma}_*\right)\mathbf{x}(a)\right)}_{\textbf{Random Quantity}}\right\|$$
$$\overset{(a)}{\leqslant}\left\|\sum_{a=1}^{A}\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top(\mathbf{x}(a)^\top\mathbf{x}(a))\right\|\frac{2Cd^2\sigma_{\max}^2\lambda_{\min}^{-1}(\mathbf{Y})\log(A/\delta)}{\Gamma}$$

where, (a) follows by Theorem B.8. Next for the third quantity in (B.25) we can bound as follows

$$\|\mathbf{v}\|=\|\mathbf{A}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma}^{-1}\mathbf{w}\|=\left\|\left(\sum_{a=1}^{A}\frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\widehat{\sigma}_\Gamma^2(a)}\right)^{-1}\mathbf{w}\right\|$$
$$\overset{(a)}{\leqslant}\left\|\left(\sum_{a=1}^{A}\frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a)+\frac{2Cd^2\sigma_{\max}^2\log(A/\delta)}{\sqrt{n}}}\right)^{-1}\mathbf{w}\right\|$$

where, (a) follows as

$$\widehat{\sigma}^2(a)\leqslant\sigma^2(a)+\frac{2Cd^2\sigma_{\max}^2\log(A/\delta)}{\Gamma}$$

from Theorem A.4. Finally observe that the first part of (B.25) we have that $\mathbf{w}^\top\mathbf{A}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*}^{-1}$ is a problem dependent parameter. Finally, plugging back

everything in (B.25) we get

$$
\begin{aligned}
&\|\mathbf{u}\| \left\| \mathbf{A}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*} - \mathbf{A}_{\mathbf{b}_*,\widehat{\boldsymbol{\Sigma}}_\Gamma} \right\| \|\mathbf{v}\| \\
&\quad \leqslant \left\| \mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*} \mathbf{w} \right\| \left\| \sum_{a=1}^{A} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top (\mathbf{x}(a)^\top \mathbf{x}(a)) \right\| \cdot \\
&\qquad\qquad \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\sigma^4_{\min}\Gamma} \left\| \left(\sum_{a=1}^{A} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a) + \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\Gamma}} \right)^{-1} \mathbf{w} \right\| \\
&\quad \leqslant \underbrace{\left(\left\| \mathbf{A}^{-1}_{\mathbf{b}_*,\boldsymbol{\Sigma}_*} \mathbf{w} \right\|^2 \left\| \sum_{a=1}^{A} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top H_U^2 \right\| \cdot \right.}_{B^*} \\
&\qquad \underbrace{\left. \left\| \left(\sum_{a=1}^{A} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a) + \frac{2Cd^2\sigma^2_{\max}\log(A/\delta)}{\Gamma}} \right)^{-1} \mathbf{w} \right\| \right)}_{B^*} \frac{2Cd^3\log(A/\delta)}{\Gamma} \\
&\quad \overset{(a)}{=} \frac{2CB^* d^3 \sigma^2_{\max} \lambda^{-1}_{\min}(\mathbf{Y}) \log(A/\delta)}{\Gamma}
\end{aligned}
$$

where, (a) follows by substituting the value of B^*. □

Regret Bound of SPEED

Theorem 1. (***formal***) *Running Algorithm 2 with budget* $n \geqslant 16C^2 d^4 \log^2(A/\delta)/\sigma^4_{\min}$ *the resulting regret satisfies*

$$
\begin{aligned}
\mathcal{R}_n &\leqslant \frac{1}{n^{3/2}} + O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma^2_{\max}\log(n)}{\sigma^2_{\min} n^{3/2}} \right) + \frac{2B^* C d^3 \sigma^2_{\max}\log(n)}{\sigma^2_{\min} n^{3/2}} \\
&\quad + \frac{d^2}{n^2} \mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top \right) + \frac{2AH_U^2\kappa^2}{n^2} \\
&= O_{\kappa^2, H_U^2}\left(\frac{B^* d^3 \sigma^2_{\max}\log(n)}{\sigma^2_{\min} n^{3/2}} \right).
\end{aligned}
$$

Proof. We follow the same steps as in Proposition 6. Observe that $\frac{16C^2d^4\sigma_{max}^4\log^2(A/\delta)}{\sigma_{min}^4} > \frac{2Cd^2\sigma_{max}^2\log(A/\delta)}{\sigma_{min}^2\Gamma}$. Hence for $\mathbf{z} = \sum_a \mathbf{w}(a)$ the loss function for $n \geqslant \frac{2Cd^2\sigma_{max}^2\log(A/\delta)}{\sigma_{min}^2\Gamma}$ as follows

$$\overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) \coloneqq \mathbb{E}\left[\left(\mathbf{z}^\top(\widehat{\theta}_{n-\Gamma} - \theta_*)\right)^2\right] \overset{(a)}{\leqslant} \left(1 + 2C_{\Gamma,\sigma_{min}^2}(\delta)\right)\mathbf{z}^\top(\widetilde{\mathbf{X}}_{n-\Gamma}^\top\widehat{\mathbf{\Sigma}}_\Gamma^{-1}\widetilde{\mathbf{X}}_{n-\Gamma})^{-1}\mathbf{z}.$$

where, (a) follows from (B.15). Recall that the quantity of the samples collected (following $\widehat{b}^*$) after exploration is as follows:

$$\begin{aligned}\left(\widetilde{\mathbf{X}}_{n-\Gamma}^\top\widehat{\mathbf{\Sigma}}_\Gamma^{-1}\widetilde{\mathbf{X}}_{n-\Gamma}\right)^{-1} &= \left(\sum_a \left\lceil (n-\Gamma)\widehat{b}^*(a)\widehat{\sigma}_\Gamma^{-2}(a)\right\rceil \mathbf{w}(a)\mathbf{w}(a)^\top\right)^{-1} \\ &= \frac{1}{n-\Gamma}\mathbf{A}_{\widehat{b}^*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1}.\end{aligned}$$

Hence we use the loss function

$$\begin{aligned}\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) &\coloneqq (1 + 2C_\Gamma(\delta))\,\mathbf{z}^\top(\widetilde{\mathbf{X}}_{n-\Gamma}^\top\widehat{\mathbf{\Sigma}}_\Gamma^{-1}\widetilde{\mathbf{X}}_{n-\Gamma})^{-1}\mathbf{z} \\ &= \frac{\left(1 + 2C_{\Gamma,\sigma_{min}^2}(\delta)\right)}{n-\Gamma}\sum_{a,a'}\mathbf{w}(a)^\top\mathbf{A}_{\widehat{b}^*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1}\mathbf{w}(a').\end{aligned}$$

Also, recall that we define

$$\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) = \frac{1}{n}\sum_{a,a'}\mathbf{w}(a)^\top\mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1}\mathbf{w}(a').$$

Then we can decompose the regret as follows:

$$\begin{aligned}
\mathcal{R}_n &= \overline{\mathcal{L}}_n(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \\
&\leqslant \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) + \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \\
&= \underbrace{\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma)}_{\textbf{Approximation error}} + \underbrace{\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma)}_{\textbf{Comparing two diff loss}} \\
&+ \underbrace{\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n^*(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*)}_{\textbf{Estimation error of } \mathbf{\Sigma}_*}
\end{aligned}$$

First recall that the good variance event as follows:

$$\xi_\delta^{var}(\Gamma) \coloneqq \left\{ \forall a, \left| \mathbf{x}(a)^\top \left(\widehat{\mathbf{\Sigma}}_\Gamma - \mathbf{\Sigma}_* \right) \mathbf{x}(a) \right| < \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\Gamma} \right\}.$$

Now first observe that $n \geqslant 16C^2d^4\sigma_{max}^4 \log^2(A/\delta)/\sigma_{min}^4$ is a larger regime than $n \geqslant \frac{2Cd^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 \Gamma}$ required for Proposition 6. Then under the good variance event, following the same steps as Proposition 6 we can bound the approximation error setting $\delta = 1/n^3$ as follows

$$\begin{aligned}
&\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) \\
&\leqslant O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 n^{3/2}} \right) \mathbb{I}\{\xi_\delta^{var}(\Gamma)\} + \sum_{t=1}^{n} A H_U^2 \kappa^2 \mathbb{P}\left((\xi_\delta^{var}(\Gamma))^c \right) \\
&\leqslant O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 n^{3/2}} \right) + \frac{A H_U^2 \kappa^2}{n^2}
\end{aligned}$$

and the second part of comparing the two losses as

$$\begin{aligned}
&\mathcal{L}'_{n-\Gamma}(\pi, \widehat{b}^*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) \\
&\leqslant O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 n^{3/2}} \right) \mathbb{I}\{\xi_\delta^{var}(\Gamma)\} + \sum_{t=1}^{n} A H_U^2 \kappa^2 \mathbb{P}\left((\xi_\delta^{var}(\Gamma))^c \right) \\
&\leqslant O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 n^{3/2}} \right) + \frac{A H_U^2 \kappa^2}{n^2}
\end{aligned}$$

We define the good estimation event as follows:

$$\xi_{\delta}^{est}(\Gamma) := \left\{ \left| \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a') - \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a') \right| \right.$$
$$\left. \leqslant \frac{2CB^* d^3 \sigma_{max}^4 \log(9H_U^2/\delta)}{\sigma_{min}^4 \Gamma} \right\}$$

Under the good estimation event $\xi^{est}(\Gamma)$ and using Theorem 3.2 we can show that the estimation error is given by

$$\begin{aligned}
&\mathcal{L}_n(\pi, \mathbf{b}_*, \widehat{\mathbf{\Sigma}}_\Gamma) - \mathcal{L}_n(\pi, \mathbf{b}_*, \mathbf{\Sigma}_*) \\
&\leqslant \left(\frac{1}{n} \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a') - \frac{1}{n} \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a') \right) \mathbb{I}\{\xi_{\delta}^{est}(\Gamma)\} \\
&\quad + \left(\frac{1}{n} \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a') - \frac{1}{n} \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a') \right) \mathbb{I}\{\xi_{\delta}^{est}(\Gamma)^C\} \\
&= \left(\frac{1}{n} \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \mathbf{w}(a') - \frac{1}{n} \sum_{a,a'} \mathbf{w}(a)^{\top} \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \mathbf{w}(a') \right) \mathbb{I}\{\xi_{\delta}^{est}(\Gamma)\} \\
&\quad + \frac{1}{n} \mathbf{Tr} \left(\left(\mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} - \mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \right) \left(\sum_{a,a'} \mathbf{w}(a) \mathbf{w}(a')^{\top} \right) \right) \mathbb{I}\{\xi_{\delta}^{est}(\Gamma)^C\} \\
&\overset{(a)}{\leqslant} \frac{1}{n} 2B^* \frac{Cd^3 \sigma_{max}^2 \log(1/\delta)}{\sigma_{min}^2 \Gamma} + \frac{1}{n} \mathbf{Tr} \left(\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_*}^{-1} \right) \mathbf{Tr} \left(\mathbf{A}_{\mathbf{b}_*,\widehat{\mathbf{\Sigma}}_\Gamma}^{-1} \right) \mathbf{Tr} \left(\sum_{a,a'} \mathbf{w}(a) \mathbf{w}(a')^{\top} \right) \delta \\
&\overset{(b)}{\leqslant} \frac{1}{n} 2B^* \frac{Cd^3 \sigma_{max}^2 \log(n)}{\sigma_{min}^2 \sqrt{n}} + \frac{d^2}{n^2} \mathbf{Tr} \left(\sum_{a,a'} \mathbf{w}(a) \mathbf{w}(a')^{\top} \right) \\
&= \frac{2B^* Cd^3 \sigma_{max}^2 \log(n)}{\sigma_{min}^2 n^{3/2}} + \frac{d^2}{n^2} \mathbf{Tr} \left(\sum_{a,a'} \mathbf{w}(a) \mathbf{w}(a')^{\top} \right)
\end{aligned}$$

where, (a) follows from Theorem 3.2, (b) follows as $\Gamma = \sqrt{n}$ and setting

$\delta = \frac{1}{n^3}$. Combining everything we have the following regret as

$$\begin{aligned}\mathcal{R}_n &\leqslant \frac{1}{n^{3/2}} + O_{\kappa^2, H_U^2}\left(\frac{d^2\sigma_{max}^2 \log(n)}{\sigma_{min}^2 n^{3/2}}\right) + \frac{2B^* C d^3 \sigma_{max}^2 \log(n)}{\sigma_{min}^2 n^{3/2}} \\ &+ \frac{d^2}{n^2}\mathbf{Tr}\left(\sum_{a,a'} \mathbf{w}(a)\mathbf{w}(a')^\top\right) + \frac{2AH_U^2\kappa^2}{n^2} \\ &= O_{\kappa^2, H_U^2}\left(\frac{B^* d^3 \sigma_{max}^2 \log(n)}{\sigma_{min}^2 n^{3/2}}\right)\end{aligned}$$

where,

$$\begin{aligned}B^* = \Bigg(\left\|\mathbf{A}_{\mathbf{b}_*, \mathbf{\Sigma}_*}^{-1}\mathbf{w}\right\|^2 \left\|\sum_{a=1}^{A} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top H_U^2\right\| \cdot \\ \left\|\left(\sum_{a=1}^{A} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a) + \frac{2Cd^3\log(9H_U^2/\delta)}{\sqrt{n}}}\right)^{-1}\mathbf{w}\right\|\Bigg).\end{aligned}$$

The claim of the theorem follows. □

Remark B.13. (***Discussion on Sample regime and*** B_*)***:*** *Observe that combining Proposition 5 and Theorem 1 we can have a loss of SPEED that scales as*

$$O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d\log(n)}{n}\right) + O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{B^* d^3 \log(n)}{n^{3/2}}\right)$$

which seems to contradict the loss bound in Proposition 6.

However, this is not the case. Observe that the B_* *is a problem-dependent quantity that depends on a number of samples* n*. We define it as*

$$B_* := \left\|\mathbf{A}_{\mathbf{b}_*, \mathbf{\Sigma}_*}^{-1}\mathbf{w}\right\|^2 \left\|\sum_{a=1}^{A} \mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top H_U^2\right\| \left\|\left(\sum_{a=1}^{A} \frac{\mathbf{b}_*(a)\mathbf{w}(a)\mathbf{w}(a)^\top}{\sigma^2(a) + \frac{2Cd^2\log(A/\delta)}{\sqrt{n}}}\right)^{-1}\mathbf{w}\right\|.$$

However, there are two regimes when $n \leqslant \frac{16C^2 d^4 \sigma_{max}^4 \log^2(A/\delta)}{\sigma_{min}^4}$ *then* $B_* = \Theta(\sqrt{n})$ *and for* $n > \frac{16C^2 d^4 \sigma_{max}^4 \log^2(A/\delta)}{\sigma_{min}^4}$ *then* $B_* = o(\sqrt{n})$ *. In the first case when* $n \leqslant \frac{16C^2 d^4 \sigma_{max}^4 \log^2(A/\delta)}{\sigma_{min}^4}$ *with* $B_* = \Theta(\sqrt{n})$ *we have the loss that scales as*

$$O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d\log(n)}{n}\right) + O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{B_* d^3 \log(n)}{n^{3/2}}\right) = O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d^3\log(n)}{n}\right)$$

This is the regime of Proposition 6 as it holds for all $n \geqslant \frac{2C d^2 \sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 \Gamma}$ *for* $\Gamma \geqslant 1$*. Note that* $\frac{2C d^2 \sigma_{max}^2 \log(A/\delta)}{\sigma_{min}^2 \Gamma}$ *is less than* $\frac{16C^2 d^4 \sigma_{max}^4 \log^2(A/\delta)}{\sigma_{min}^4}$*.*

In the second case when $n > \frac{16C^2 d^4 \sigma_{max}^4 \log^2(A/\delta)}{\sigma_{min}^4}$ *with* $B_* = o(\sqrt{n})$ *we have a tighter bound as the first term dominates and we have the loss scaling as*

$$O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d\log(n)}{n}\right) + O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{B^\star d^3 \log(n)}{n^{3/2}}\right) = O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d\log(n)}{n}\right)$$

Intuitively this is a larger sample regime where the SPEED has a good estimation of $\tilde{}_*$ *and the design matrix estimation has also concentrated. Combining both the regimes we can show that for* $n \geqslant \frac{2C d^2 \log(A/\delta)}{\sigma\_min^2 \Gamma}$ *the loss of SPEED scales by*

$$\max\left\{O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d\log(n)}{n}\right), O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d^3\log(n)}{n}\right)\right\} = O_{\kappa^2, H_U^2, \sigma_{max}^2, \sigma_{min}^2}\left(\frac{d^3\log(n)}{n}\right)$$

which is the bound of Proposition 6. So in summary Proposition 6 is a more general bound for a larger regime size than Theorem 1 and does not contradict the theorem statement.

B.2 Regret Lower Bound

Theorem 2. (***Lower Bound***) *Let* $|\mathbf{\Theta}| = 2^d$ *and* $\theta_* \in \mathbf{\Theta}$. *Then any* δ*-PAC policy* π *following the design* $\mathbf{b} \in \triangle(\mathcal{A})$ *satisfies* $\mathcal{R}'_n = \mathcal{L}_n(\pi, \widehat{b}, \mathbf{\Sigma}_*) - \mathcal{L}_n(\pi, \mathbf{b}^*, \mathbf{\Sigma}_*) \geqslant \Omega\left(\frac{d^2 \lambda_d(\mathbf{V}) \log(n)}{n^{3/2}}\right)$ *for the environment in* (B.26).

Proof. **Step 1 (Define Environment):** We define an environment model B_j consisting of A actions and J hypotheses with true hypothesis $\theta_* = \theta_j$ (j-th column) as follows:

$$\begin{array}{cccccccc} \theta & = & \theta_1 & \theta_2 & \theta_3 & \dots & \theta_J \\ \hline \mu_1(\theta) & = & \beta & \beta - \frac{\beta}{J} & \beta - \frac{2\beta}{J} & \dots & \beta - \frac{(J-1)\beta}{J} \\ \mu_2(\theta) & = & \iota_{21} & \iota_{22} & \iota_{23} & \dots & \iota_{2J} \\ & \vdots & & & \vdots & & \\ \mu_A(\theta) & = & \iota_{A1} & \iota_{A2} & \iota_{A3} & \dots & \iota_{AJ} \end{array} \tag{B.26}$$

where, each ι_{ij} is distinct and satisfies $\iota_{ij} < \beta/4J$. θ_1 is the optimal hypothesis in B_1, θ_2 is the optimal hypothesis in B_2 and so on such that for each B_j and $j \in [J]$ we have column j as the optimal hypothesis.

Finally, assume that $\mathbf{\Sigma} = \theta\theta^\top$ is a rank one matrix. To distinguish between the covariance matrix between two distributions we denote $\mathbf{\Sigma}_\theta = \theta\theta^\top$. Therefore we have that $\sigma_i^2(\theta) = \mathbf{x}_i^\top \mathbf{\Sigma}_\theta \mathbf{x}_i = (\mathbf{x}_i^\top \theta)^2 = \mu_i^2(\theta)$. Hence for any algorithm, identifying the co-variance matrix $\mathbf{\Sigma}_{\theta_*}$ is the same as identifying the θ_*. Also assume that $\pi(a) = \frac{1}{A}$. Hence each action is equally weighted by the target policy.

This is a general hypothesis testing setting where the functions $\mu_a(\theta)$ can be thought of as linear functions of θ such that $\mu_a(\theta) = \mathbf{x}(a)^\top \theta$. Assume that $0 < \mu_a(\theta) \leqslant 1$, and $\log(\mu_a(\theta)/\mu_a(\theta')) > 1/4$.

Now observe that between any two hypothesis θ and θ' we have the

following

$$\begin{aligned}
&\mathrm{KL}\Big(\mathcal{N}(\mu_i(\theta), \mathbf{x}_i^\top \boldsymbol{\Sigma}_\theta \mathbf{x}_i))\big|\big|\mathcal{N}(\mu_i(\theta'), \mathbf{x}_i^\top \boldsymbol{\Sigma}_{\theta'} \mathbf{x}_i))\Big) \\
&\quad = 2\log(\frac{\sigma_i(\theta')}{\sigma_i(\theta)}) + \frac{\sigma_i^2(\theta) + (\mu_i(\theta) - \mu_i(\theta'))^2}{2\sigma_i^2(\theta')} - \frac{1}{2} \\
&\quad \overset{(a)}{=} 2\log(\frac{\mu_i(\theta')}{\mu_i(\theta)}) + \frac{\mu_i^2(\theta) + (\mu_i(\theta) - \mu_i(\theta'))^2}{2\mu_i^2(\theta')} - \frac{1}{2} \overset{(a)}{\geqslant} \frac{(\mu_i(\theta) - \mu_i(\theta'))^2}{8}
\end{aligned} \tag{B.27}$$

where, (a) follows from the condition that $0 < \mu_a(\theta) \leqslant 1$, and

$$\log(\mu_a(\theta)/\mu_a(\theta')) > 1/4.$$

Step 2 (Minimum samples to verify θ_*): Let, Λ_1 be the set of alternate models having a different optimal hypothesis than $\theta^* = \theta_1$ such that all models having different optimal hypothesis than θ_1 such as $B_2, B_3, \ldots B_J$ are in Λ_1. Let τ_δ be the stopping time for any δ-PAC policy $\mathbf{b}$. That is τ_δ is the time that any algorithm stops and outputs its estimate $\widehat{\theta}_{\tau_\delta}$. Let $T_t(a)$ denote the number of times the action a has been sampled till round t. Let $\widehat{\theta}_{\tau_\delta}$ be the predicted optimal hypothesis at round τ_δ. We first consider the model B_1. Define the event $\xi = \{\widehat{\theta}_{\tau_\delta} \neq \theta_*\}$ as the error event in model B_1. Let the event $\xi' = \{\widehat{\theta}_{\tau_\delta} \neq \theta'^*\}$ be the corresponding error event in model B_2. Note that $\xi^C \subset \xi'$. Now since $\mathbf{b}$ is δ-PAC policy we have $\mathbb{P}_{B_1,\mathbf{b}}(\xi) \leqslant \delta$ and $\mathbb{P}_{B_2,\mathbf{b}}(\xi^C) \leqslant \delta$. Hence we can show that,

$$2\delta \geqslant \mathbb{P}_{B_1,\mathbf{b}}(\xi) + \mathbb{P}_{B_2,\mathbf{b}}(\xi^{\complement}) \overset{(a)}{\geqslant} \frac{1}{2}\exp\left(-\mathrm{KL}\left(P_{B_1,\mathbf{b}} \| P_{B_2,\mathbf{b}}\right)\right)$$

$$\mathrm{KL}\left(P_{B_1,\mathbf{b}} \| P_{B_2,\mathbf{b}}\right) \geqslant \log\left(\frac{1}{4\delta}\right)$$

$$\frac{1}{8}\sum_{i=1}^{A} \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(i)] \cdot \left(\mu_i(\theta_*) - \mu_i(\theta_*')\right)^2 \overset{(b)}{\geqslant} \log\left(\frac{1}{4\delta}\right)$$

$$\frac{1}{8}\left(\beta - \beta + \frac{\beta}{J}\right)^2 \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(1)] + \frac{1}{8}\sum_{i=2}^{A} (\iota_{i1} - \iota_{i2})^2 \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(i)] \overset{(c)}{\geqslant} \log\left(\frac{1}{4\delta}\right)$$

$$\frac{1}{8}\left(\frac{1}{J}\right)^2 \beta^2 \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(1)] + \frac{1}{8}\sum_{i=2}^{A} (\iota_{i1} - \iota_{i2})^2 \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(i)] \geqslant \log\left(\frac{1}{4\delta}\right)$$

$$\frac{1}{8}\left(\frac{1}{J}\right)^2 \beta^2 \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(1)] + \frac{1}{8}\sum_{i=2}^{A} \frac{\beta^2}{4J^2} \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(i)] \overset{(d)}{\geqslant} \log\left(\frac{1}{4\delta}\right) \tag{B.28}$$

where, (a) follows from Theorem B.15, (b) follows from Theorem B.14, (c) follows from the construction of the bandit environments and (B.27), and (d) follows as $(\iota_{ij} - \iota_{ij'})^2 \leqslant \frac{\beta^2}{4J^2}$ for any i-th action and j-th hypothesis.

Now, we consider the alternate model B_3. Again define the event $\xi = \{\widehat{\theta}_{\tau_\delta} \neq \theta_*\}$ as the error event in model B_1 and the event $\xi' = \{\widehat{\theta}_{\tau_\delta} \neq \theta_*''\}$ be the corresponding error event in model B_3. Note that $\xi^{\complement} \subset \xi'$. Now since $\mathbf{b}$ is δ-PAC policy we have $\mathbb{P}_{B_1,\mathbf{b}}(\xi) \leqslant \delta$ and $\mathbb{P}_{B_3,\mathbf{b}}(\xi^{\complement}) \leqslant \delta$. Following the same way as before we can show that,

$$\frac{1}{8}\left(\frac{2}{J}\right)^2 \beta^2 \mathbb{E}_{B_3,\mathbf{b}}[T_{\tau_\delta}(1)] + \frac{1}{8}\sum_{i=2}^{A} \frac{\beta^2}{4J^2} \mathbb{E}_{B_3,\mathbf{b}}[T_{\tau_\delta}(i)] \overset{(d)}{\geqslant} \log\left(\frac{1}{4\delta}\right). \tag{B.29}$$

Similarly, we get the equations for all the other $(J-2)$ alternate models in Λ_1. Now consider an optimization problem (ignoring the constant factor

of $\frac{1}{8}$ across all the constraints)

$$\begin{aligned}
&\min_{t_i : i \in [A]} \sum t_i \\
\text{s.t.} \quad &\left(\frac{1}{J}\right)^2 \beta^2 t_1 + \frac{\beta^2}{4J^2} \sum_{i=2}^{A} t_i \geqslant \log(1/4\delta) \\
&\left(\frac{2}{J}\right)^2 \beta^2 t_1 + \frac{\beta^2}{4J^2} \sum_{i=2}^{A} t_i \geqslant \log(1/4\delta) \\
&\vdots \\
&\left(\frac{J-1}{J}\right)^2 \beta^2 t_1 + \frac{\beta^2}{4J^2} \sum_{i=2}^{A} t_i \geqslant \log(1/4\delta) \\
&t_i \geqslant 0, \forall i \in [A]
\end{aligned}$$

where the optimization variables are t_i. It can be seen that the optimum objective value is $J^2\beta^{-2}\log(1/4\delta)$. Interpreting $t_i = \mathbb{E}_{B_1,\mathbf{b}}[T_{\tau_\delta}(i)]$ for all i, we get that $\mathbb{E}_{B_1,\mathbf{b}}[\tau_\delta] = \sum_i t_i = t_1 \geqslant J^2\beta^{-2}\log(1/4\delta)$ which gives us the required lower bound to the number of pulls of action 1. Observe that the optimum objective value is reached by substituting $t_1 = J^2\beta^{-2}\log(1/4\delta)$ and $t_2 = \ldots = t_A = 0$. It follows that for verifying any hypothesis $\theta_j \neq \theta_*$ the verification proportion is given by $\mathbf{b}_{\theta_j} = (1, \underbrace{0, 0, \ldots, 0}_{\text{(A-1) zeros}})$. Observe setting $\beta = J\sqrt{\log(1/4\delta)/n}$ recovers $\tau_\delta = n$ which implies that a budget of n samples is required for verifying hypothesis $\theta_j = \theta_*$. For the remaining steps we take $\beta = J\sqrt{\log(1/4\delta)/n}$.

Step 3 (Lower Bounding Regret): Then we can show that the MSE of

any hypothesis $\theta_j = \theta_*$

$$\mathbb{E}_{\mathcal{D}}\left[\left(\sum_a \pi(a)\mathbf{x}(a)^\top(\theta_j - \widehat{\theta}_n)\right)^2\right] = \frac{1}{n}\sum_{a,a'}\mathbf{w}(a)\mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\Sigma_{\theta_*}}\mathbf{w}(a') = \frac{1}{n}\mathbf{Tr}\Bigg(\mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\Sigma_{\theta_*}}\underbrace{\sum_{a,a'}\mathbf{w}(a)\mathbf{w}(a')}_{V}\Bigg)$$

where, $\mathbf{b}_{\theta_j}(a)$ is the number of samples allocated to action a. First we will bound the loss of the oracle for this environment given by $\mathcal{L}_n(\pi, \mathbf{b}, \Sigma_{\theta_*}) = \frac{1}{n}\mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\Sigma_{\theta_*}}\mathbf{V})$. Note that the oracle has access to the Σ_{θ_*}, so it only need to verify whether $\theta_j = \theta_*$ by following $\mathbf{b}_{\theta_j}$. Then we have that

$$\mathbf{A}_{\mathbf{b}_{\theta_j},\Sigma_{\theta_*}} = \sum_a \mathbf{b}_{\theta_j}(a)\frac{\mathbf{x}(a)\mathbf{x}(a)^\top}{\sigma^2(a)} = \frac{\mathbf{w}(1)\mathbf{w}(1)^\top}{(\mathbf{x}(1)^\top\theta_j)^2} = \frac{\mathbf{w}(1)\mathbf{w}(1)^\top}{(\beta - \frac{j\beta}{J})^2}$$

$$\implies \mathbf{Tr}(\mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\Sigma_{\theta_*}}) = \frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}$$

Now we will bound the loss of the algorithm that uses $\widehat{\Sigma}_\beta$ to estimate $\widehat{b}$. It then collects the $\mathcal{D}$ and uses it to estimate θ_* following the WLS estimation using Σ_{θ_*}.

Denote the number of times the algorithm samples each action i be $T'_n(i)$. Let the algorithm allocate $T'_n(1) = J^2\beta^{-2}\log(1/4\delta) - d$ samples to action 1 and to any other action i' it allocates $T'_n(i') = d$ samples such that $d \geqslant 1$. WLOG let $i' = 2$. Finally let $T'_n(3) = \ldots = T'_n(A) = 0$. Hence the optimal action 1 is under-allocated and the sub-optimal action 2 is over-allocated. The loss of such an algorithm now is given by

$$\mathcal{L}_n(\pi, \widehat{b}, \Sigma_{\theta_*}) = \frac{1}{n}\mathbf{Tr}(\mathbf{A}^{-1}_{\widehat{b},\Sigma_{\theta_*}}\mathbf{V}).$$

Hence it follows by setting $\delta = 1/(nJ)$ that

$$\begin{aligned}
\mathbf{A}_{\widehat{b},\boldsymbol{\Sigma}_{\theta_*}} = \frac{1}{n}\sum_{a} n\widehat{b}(a)\frac{\mathbf{x}(a)\mathbf{x}(a)^\top}{\sigma^2(a)} &= \frac{1}{n}\sum_{a} T_n'(a)\frac{\mathbf{x}(a)\mathbf{x}(a)^\top}{\sigma^2(a)} \\
&= \frac{1}{n}T_n'(1)\frac{\mathbf{x}(1)\mathbf{x}(1)^\top}{\sigma^2(1)} + \underbrace{\frac{1}{n}T_n'(2)\frac{\mathbf{x}(2)\mathbf{x}(2)^\top}{\sigma^2(2)}}_{\geqslant 0} \\
&\geqslant \frac{1}{n}T_n'(1)\frac{\mathbf{w}(1)\mathbf{w}(1)^\top}{(\mathbf{x}(1)^\top\theta_j)^2} \\
&\overset{(a)}{=} \frac{J^2\beta^{-2}\log(nJ) - d}{n}\frac{\mathbf{w}(1)\mathbf{w}(1)^\top}{(\beta - \frac{j\beta}{J})^2}
\end{aligned}$$

where, (a) follows by substituting the value of T_n'. Then we have that

$$\begin{aligned}
\mathbf{Tr}(\mathbf{A}^{-1}_{\widehat{b},\boldsymbol{\Sigma}_{\theta_*}}) &\geqslant \frac{n}{J^2\beta^{-2}\log(nJ) - d}\frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)} \\
&= \frac{n}{J^2\beta^{-2}(\log(nJ) - \frac{d}{J^2\beta^{-2}})}\frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)} \\
&\overset{(a)}{\geqslant} \frac{\beta^2\log(nJ) + \frac{d}{J^2\beta^{-2}}}{J^2}\frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)} \\
&\geqslant \frac{\beta^2\log(nJ)}{J^2}\frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}
\end{aligned}$$

where, (a) follows as for $d \geqslant 1$ we have that

$$n - (\log(nJ))^2 \geqslant -\frac{d^2}{(J^2\beta^{-2})^2} \implies (\log(nJ) - \frac{d}{J^2\beta^{-2}})^{-1} \geqslant \log(nJ) + \frac{d}{J^2\beta^{-2}}.$$

Step 4 (Lower Bound regret): Hence we have the regret for verifying

any hypothesis $\theta_j = \theta_*$ as follows:

$$\begin{aligned}
\mathcal{R}'_n &= \mathcal{L}_n(\pi, \widehat{b}, \mathbf{\Sigma}_{\theta_*}) - \mathcal{L}_n(\pi, \mathbf{b}^*, \mathbf{\Sigma}_{\theta_*}) \\
&\geqslant \frac{1}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\widehat{b},\mathbf{\Sigma}_{\theta_*}}\mathbf{V}\right) - \frac{1}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\mathbf{\Sigma}_{\theta_*}}\mathbf{V}\right) = \frac{1}{n}\mathbf{Tr}\left(\left(\mathbf{A}^{-1}_{\widehat{b},\mathbf{\Sigma}_{\theta_*}} - \mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\mathbf{\Sigma}_{\theta_*}}\right)\mathbf{V}\right) \\
&\geqslant \frac{\lambda_d(\mathbf{V})}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\widehat{b},\mathbf{\Sigma}_{\theta_*}} - \mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\mathbf{\Sigma}_{\theta_*}}\right) \\
&= \frac{\lambda_d(\mathbf{V})}{n}\left[\mathbf{Tr}\left(\mathbf{A}^{-1}_{\widehat{b},\mathbf{\Sigma}_{\theta_*}}\right) - \mathbf{Tr}\left(\mathbf{A}^{-1}_{\mathbf{b}_{\theta_j},\mathbf{\Sigma}_{\theta_*}}\right)\right] \\
&= \frac{\lambda_d(\mathbf{V})}{n}\left[\frac{\beta^2\log(nJ)}{J^2}\frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)} - \frac{(\beta - \frac{j\beta}{J})^2}{\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}\right] \\
&= \frac{\lambda_d(\mathbf{V})\beta^2(\beta - \frac{j\beta}{J})^2}{n\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}\left[\frac{\log(nJ)}{J^2} - 1\right]
\end{aligned}$$

$$\begin{aligned}
&\overset{(a)}{\geqslant} \frac{\lambda_d(\mathbf{V})\beta^2(\beta - \frac{j\beta}{J})^2}{n\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}\left[\frac{\log(nJ)}{2J^2}\right] \\
&\overset{(b)}{\geqslant} \frac{d\lambda_d(\mathbf{V})\beta^2}{n^{3/2}\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}\left[\frac{\log(nJ)}{2J^2}\right] \\
&\overset{(c)}{\geqslant} \frac{d^2\lambda_d(\mathbf{V})\beta^2}{n^{3/2}\mathbf{Tr}(\mathbf{w}(1)\mathbf{w}(1)^\top)}\log(2n) \\
&= \Omega\left(\frac{d^2\lambda_d(\mathbf{V})\log(n)}{n^{3/2}}\right)
\end{aligned}$$

where, (a) follows as $\frac{\log(nJ)}{J^2} - 1 \geqslant \frac{\log(nJ)}{2J^2}$, (b) follows as gap $(\beta - \frac{j\beta}{J})^2 \geqslant \frac{d}{\sqrt{n}}$ for any θ_j, and (c) follows by substituting $|\mathbf{\Theta}| = J = 2^d$. □

Lemma B.14. (***Restatement of Lemma 15.1 in Lattimore and Szepesvári (2020b), Divergence Decomposition***) *Let* B *and* B' *be two bandit models having different optimal hypothesis* θ_* *and* θ'^* *respectively. Fix some policy* π *and round* n*. Let* $\mathbb{P}_{B,\pi}$ *and* $\mathbb{P}_{B',\pi}$ *be two probability measures induced by some*

n*-round interaction of* π *with* B *and* π *with* B' *respectively. Then*

$$\mathrm{KL}\left(\mathbb{P}_{B,\pi} \| \mathbb{P}_{B',\pi}\right) = \sum_{i=1}^{A} \mathbb{E}_{B,\pi}[T_n(i)] \cdot \mathrm{KL}(\mathcal{N}(\mu_i(\theta), 1) \| \mathcal{N}(\mu_i(\theta_*), 1))$$

where, $\mathrm{KL}(.\|.)$ *denotes the Kullback-Leibler divergence between two probability measures and* $T_n(i)$ *denotes the number of times action* i *has been sampled till round* n*.*

Lemma B.15. (***Restatement of Lemma 2.6 in Tsybakov (2008)***) *Let* $\mathbb{P}, \mathbb{Q}$ *be two probability measures on the same measurable space* $(\Omega, \mathcal{F})$ *and let* $\xi \subset \mathcal{F}$ *be any arbitrary event then*

$$\mathbb{P}(\xi) + \mathbb{Q}\left(\xi^{\complement}\right) \geqslant \frac{1}{2} \exp\left(-\mathrm{KL}(\mathbb{P} \| \mathbb{Q})\right)$$

where $\xi^{\complement}$ *denotes the complement of event* ξ *and* $\mathrm{KL}(\mathbb{P}\|\mathbb{Q})$ *denotes the Kullback-Leibler divergence between* $\mathbb{P}$ *and* $\mathbb{Q}$*.*

Environment $\mathcal{E}$**:** Consider the environment $\mathcal{E}$ which consist of 3 actions in $\mathbb{R}^2$ such that $\mathbf{x}(1) = [1, 0]$ is along x-axis, $\mathbf{x}(2) = [0, 1]$ is along y-axis and $\mathbf{x}(3) = [1/\sqrt{2}, 1/\sqrt{2}]$. Let $\theta_* = [1, 0]$ and so the optimal action is action 1. Let the target policy $\pi = [0.9, 0.1, 0.0]$. Finally, let the variances be $\sigma^2(1) = 5/100$, $\sigma^2(2) = 1.0$ and $\sigma^2(3) = 5/100$.

Proposition 8. (***Onpolicy regret***) *Let the Onpolicy algorithm have access to the variance in environment* $\mathcal{E}$*. Then the regret of Onpolicy scales as* $O\left(\frac{\lambda_1(\mathbf{V})}{n}\right)$.

Proof. Recall that in $\mathcal{E}$, there are 3 actions in $\mathbb{R}^2$ such that $\mathbf{x}(1) = [1, 0]$ is along x-axis, $\mathbf{x}(2) = [0, 1]$ is along y-axis and $\mathbf{x}(3) = [1/\sqrt{2}, 1/\sqrt{2}]$. The $\theta_* = [1, 0]$ and so the optimal action is action 1. The target policy $\pi = [0.9, 0.1, 0.0]$. Finally, let the variances be $\sigma^2(1) = 1.0$, $\sigma^2(2) = 1.0$ and

$\sigma^2(3) = 5/100$. Hence, PE-Optimal design results in $\mathbf{b}^* = [0.5, 0.5, 0.0]$.

$$\mathbf{A}_{\pi,\mathbf{\Sigma}_{\theta_*}} = \sum_a \pi(a)\frac{\mathbf{x}(a)\mathbf{x}(a)^\top}{\sigma^2(a)} = \frac{9}{10}\cdot\mathbf{x}(1)\mathbf{x}(1)^\top + \frac{1}{10}\mathbf{x}(2)\mathbf{x}(2)^\top$$
$$\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_{\theta_*}} = \sum_a \mathbf{b}_*(a)\frac{\mathbf{x}(a)\mathbf{x}(a)^\top}{\sigma^2(a)} = \frac{1}{2}\cdot\mathbf{x}(1)\mathbf{x}(1)^\top + \frac{1}{2}\mathbf{x}(2)\mathbf{x}(2)^\top$$

Recall that $\mathbf{V} = \sum_a \mathbf{w}(a)\mathbf{w}(a)^\top$. Hence, the regret scales as

$$\begin{aligned}\mathcal{R}_n &= \mathcal{L}_n(\pi,\pi,\mathbf{\Sigma}_{\theta_*}) - \mathcal{L}_n(\pi,\mathbf{b}^*,\mathbf{\Sigma}_{\theta_*})\\ &\leqslant \frac{1}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\pi,\mathbf{\Sigma}_{\theta_*}}\mathbf{V}\right) - \frac{1}{n}\mathbf{Tr}\left(\mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_{\theta_*}}\mathbf{V}\right) = \frac{1}{n}\mathbf{Tr}\left(\left(\mathbf{A}^{-1}_{\pi,\mathbf{\Sigma}_{\theta_*}} - \mathbf{A}^{-1}_{\mathbf{b}_*,\mathbf{\Sigma}_{\theta_*}}\right)\mathbf{V}\right)\\ &\overset{(a)}{\leqslant} O\left(\frac{\lambda_1(\mathbf{V})}{n}\right)\end{aligned}$$

where, (a) follows by substituting the value of $\mathbf{A}_{\pi,\mathbf{\Sigma}_{\theta_*}}$ and $\mathbf{A}_{\mathbf{b}_*,\mathbf{\Sigma}_{\theta_*}}$. □

B.3 Additional Experiments

In this section, we state additional experimental details.

Unit Ball: This experiment consists of a set of 4 actions that are arranged in a unit ball in $\mathbb{R}^2$, and $\|\mathbf{x}(a)\| = 1$ for all $a \in \mathcal{A}$. We consider three groups of actions: **a**) the reward-maximizing action in the direction of θ^*, **b**) the informative action (orthogonal to optimal action) that maximally reduces the uncertainty of $\widehat{\theta}_t$ and **c**) the less-informative actions as shown in Figure 3.1 (Top-Left). The variance of the most informative action is chosen to be high (0.35), but the target probability is set as low 0.1, which forces the on-policy algorithm to sample the high variance action less. Figure 3.1 (Top-Right) shows that SPEED outperforms Onpolicy, G-Optimal , and A-Optimal . Note that we experiment with A-Optimal design (Fontaine et al., 2021) because this criterion results in minimizing the average variance of the estimates of the regression coefficients and is most closely aligned

with our goal than G-, or, D-optimal designs (Jamieson and Jain, 2022).

Air Quality: We perform this experiment on real-world dataset Air Quality from UCI datasets. The Air quality dataset consists of 1500 samples each of which consists of 6 features. We first select 400 samples which are the actions in our setting. We then fit a weighted least square estimate to the original dataset and get an estimate of θ_* and Σ_*. The reward model is linear and given by $\mathbf{x}_{I_t}^\top \theta_* +$ noise where $\mathbf{x}_{I_t}$ is the observed action at round t, and the noise is a zero-mean additive noise with variance scaling as $\mathbf{x}_{I_t}^\top \Sigma_* \mathbf{x}_{I_t}$. Hence the variance of each action depends on their feature vectors and Σ_*. Finally, we set a level τ, such that 30 actions having variance crossing τ are set with low target probability, and the remaining probability mass is uniformly distributed among the rest 370 action. Hence, again high variance actions are set with a low target probability, which forces the on-policy algorithm to sample the high-variance action less number of times. We apply SPEED to this problem and compare it to baselines A-Optimal , G-Optimal , and the Onpolicy algorithm.

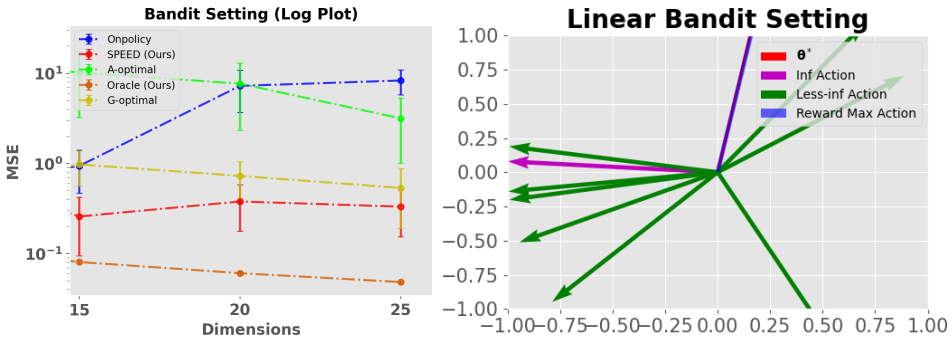


Figure B.1: 10 action unit ball environment

Red Wine Quality: The UCI Red Wine Quality dataset consist of 1600 samples of red wine with each sample i having feature $\mathbf{x}_i \in \mathbb{R}^{11}$. We first fit a weighted least square estimate to the original dataset and get an estimate of θ^* and Σ_*. The reward model is linear and given by $\mathbf{x}_{I_t}^\top \theta^* +$ noise where $\mathbf{x}_{I_t}$ is the observed action at round t, and the noise is a zero-mean

additive noise with variance scaling as $\mathbf{x}_{I_t}^\top \boldsymbol{\Sigma}_* \mathbf{x}_{I_t}$. Note that we consider the 1600 samples as actions. Then we run each of our benchmark algorithms on this dataset and reward model. Finally, we set a level τ, such that 40 actions having variance crossing τ are set with low target probability, and the remaining probability mass is uniformly distributed among the rest 1560 action. Hence, again high variance actions are set with a low target probability, which forces the on-policy algorithm to sample the high-variance action less number of times. We apply SPEED to this problem and compare it to baselines A-Optimal , G-Optimal , and the Onpolicy algorithm.

Movielens: We experiment with a movie recommendation problem on the MovieLens 1M dataset (Lam and Herlocker, 2016). This dataset contains one million ratings given by 6 040 users to 3 952 movies. We first apply a low-rank factorization to the rating matrix to obtain 5-dimensional representations: $\theta_j \in \mathbb{R}^5$ for user $j \in [6\,040]$ and $\mathbf{x}(a) \in \mathbb{R}^5$ for movie $a \in [3\,952]$. In each run, we choose one user θ_j and 100 movies $\mathbf{x}(a)$ randomly, and they represent the unknown model parameter and known feature vectors of actions, respectively.

Increasing Dimension: We perform this experiment to show how the MSE of SPEED scales with increasing dimensions and number of actions. We choose dimension $d \in \{15, 20, 25\}$. For each dimension $d \in \{15, 20, 25\}$ we choose the number of actions $|\mathcal{A}| = d^2 + 20$. Hence we ensure that the number of actions are greater than d^2 dimensions. We also choose the horizon as $T \in \{13000, 18000, 25000\}$ for each $d \in \{15, 20, 25\}$. We choose the same environment as the unit ball experiment. So the actions arranged in a unit ball in $\mathbb{R}^2$ and $\|\mathbf{x}(a)\| = 1$ for all $a \in \mathcal{A}$. Again we consider three groups of actions: **a)** the reward-maximizing action in the direction of θ^*, **b)** the informative action (orthogonal to optimal action) that maximally reduces the uncertainty of $\widehat{\theta}_t$ and **c)** the less-informative actions as shown in Figure B.1 but scaled to a larger set of actions. For each

case of dimension $d \in \{15, 20, 25\}$, the variance of the most informative actions along the directions orthogonal to the reward maximizing action are chosen to be high, but the target probability is set as low, which forces the on-policy algorithm to sample the high variance action less. We again show the performance in Figure 3.1 (Bottom-left). We observe that with increasing dimensions d the SPEED outperforms on-policy. Also, observe that the oracle with knowledge of $\mathbf{\Sigma}_*$ performs the best.

B.4 Table of Notations

Notations	**Definition**
$\pi(a)$	Target policy probability for action a
$\mathbf{b}(a)$	Behavior policy probability for action a
$\mathbf{x}(a)$	Feature of action a
θ_*	Optimal mean parameter
$\widehat{\theta}_n$	Estimate of θ_*
$\mu(a) = \mathbf{x}^\top \theta_*$	Mean of action a
$\widehat{\mu}_t(a) = \mathbf{x}^\top \widehat{\theta}_t$	Empirical mean of action a at time t
$R_t(a)$	Reward for action a at time t
$\mathbf{\Sigma}_*$	Optimal co-variance matrix
$\widehat{\mathbf{\Sigma}}_t$	Empirical co-variance matrix at time t
$\sigma^2(a) = \mathbf{x}(a)^\top \mathbf{\Sigma}_* \mathbf{x}(a)$	Variance of action a
$\widehat{\sigma}_t^2(a) = \mathbf{x}(a)^\top \widehat{\mathbf{\Sigma}}_t \mathbf{x}(a)$	Empirical variance of action a at time t
n	Total budget
$T_n(a)$	Total Samples of action a after n timesteps

Table B.1: Table of Notations for SPEED

C APPENDIX: SAVER: OPTIMAL DATA COLLECTION STRATEGY FOR SAFE POLICY EVALUATION IN TABULAR MDPS

Previous results and Probability Tools

Proposition 1. (***Restatement from Carpentier and Munos (2011)***) *In an* A*-action bandit setting, the estimated return of* π *after* n *action-reward samples is denoted by* Y_n*. Note that the expectation of* Y_n *after each action has been sampled once is given by* V^π*. Minimal MSE,* $\mathbb{E}_{\mathcal{D}}\left[(Y_n - V^\pi)^2\right]$*, is obtained by taking actions in the proportion:*

$$\mathbf{b}_*(a) \coloneqq \frac{\pi(a)\sigma(a)}{\sum_{a'=1}^{A} \pi(a')\sigma(a')}. \tag{C.1}$$

where $\mathbf{b}^*(a)$ *denotes the optimal sampling proportion.*

Lemma C.1. (***Wald's lemma for variance***) (*Resnick, 2019*) *Let* $\{\mathcal{F}_t\}$ *be a filtration and* R_t *be a* $\mathcal{F}_t$*-adapted sequence of i.i.d. random variables with variance* σ^2*. Assume that* $\mathcal{F}_t$ *and the* σ*-algebra generated by* $\{R_{t'} : t' \geqslant t+1\}$ *are independent and* T *is a stopping time w.r.t.* $\mathcal{F}_t$ *with a finite expected value. If* $\mathbb{E}\left[R_1^2\right] < \infty$ *then*

$$\mathbb{E}\left[\left(\sum_{t'=1}^{n} R_{t'} - n\mu\right)^2\right] = \mathbb{E}[n]\sigma^2$$

Lemma C.2. (***Restatement of Theorem 1 of Mukherjee et al. (2022a)***) *Assume the underlying MDP is an* L*-depth tree MDP as defined in Theorem 4.1. Let the estimated return of the starting state* s_1^1 *after* n *state-action-reward samples be defined as* $Y_n(s_1^1)$*. Let* $\mathcal{D}$ *be the observed data over* n *state-action-reward samples. To minimize MSE* $\mathbb{E}_{\mathcal{D}}[(Y_n(s_1^1) - V^\pi(s_1^1))^2]$ *the optimal sampling proportions for*

any arbitrary state is given by:

$$\mathbf{b}_*(a|s_i^\ell) \propto \left(\pi^2(a|s_i^\ell)\left[\sigma^2(s_i^\ell, a) + \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) M^2(s_j^{\ell+1})\right]\right)^{1/2},$$

where, $M(s_j^\ell)$ *is the normalization factor defined as follows:*

$$M(s_i^\ell) \coloneqq \sum_a \left(\pi^2(a|s_i^\ell)\big(\sigma^2(s_i^\ell, a) + \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) M^2(s_j^{\ell+1})\big)\right)^{1/2}$$

C.1 Intractable MDP

Proposition 1. *Fix an arbitrary* $n > 0$*. Then there exists an environment where no algorithm (including the safe oracle* $\mathbf{b}_*^k$*) can be run that will result in a regret* $\mathcal{R}_n = \mathcal{L}_n(\pi, \mathbf{b}) - \mathcal{L}_n^*(\pi, \mathbf{b}_*)$ *of* $\widetilde{O}(n^{-3/2})$ *while satisfying the safety constraint, where* $\mathbf{b}_*$ *is the unconstrained oracle.*

Proof. We first consider a bandit setting where there are 3 arms, action $\{0\}$ which is the safe action, and actions 1 and 2. Assume $\pi(a) = 1/A$ so that we can ignore its effect on optmal sampling policy $\mathbf{b}_*$

Case 1 (All actions safe): First consider an environment when all actions are safe. That is $\mu^c(0) = 0$ and $\mu^c(1) = 1$ and $\mu^c(2) = 1 - \epsilon$ and reward distributions are bounded between $[0, 1]$. Therefore at round $\ell \in [L]$ we can guarantee for any $\alpha \in (0, 1]$ that

$$\sum_{\ell'=1}^{\ell} \sum_{a=0}^{2} \pi(a) \widehat{\mu}_{c,\ell'}(a) \geqslant (1-\alpha)\ell \underbrace{\pi_0(0)\mu^c(0)}_{0}, \quad \forall \ell \in [L]$$

where, π_0 always samples safe action 0. Assume a safe oracle that knows the variances of the actions but does not know the means of the actions (both reward and cost means). Therefore from Carpentier and Munos (2011) we know that the optimal way to reduce the MSE $\min_{\mathbf{b}} \mathbb{E}_{\mathcal{D}}[(Y_n^\pi(s_1) - V^\pi(s_1))^2]$

is to run the policy $\mathbf{b}_*(a) \propto \pi(a)\sigma(a)$. We also know from Carpentier and Munos (2011) that there exists an algorithm $\mathcal{A}^{safe}$ (like MC-UCB that tracks $\mathbf{b}_*$) that achieves a regret after n rounds as $\mathcal{R}_n^{\text{safe}} = \widetilde{O}(\frac{K \log(n)}{n^{3/2}})$ where $\widetilde{O}$ hides logarithmic factors and problem dependent factors like $\mathbf{b}_{\min}$.

Case 2 (Some actions are unsafe): In this case, we now analyze a safe oracle algorithm $\mathbf{b}_*^k$. Consider an environment where $\mu^c(0) = 0.5$, $\mu^c(1) = 0.5 + \alpha$, and $\mu^c(2) = 0$. Let the rewards be bounded in $[0, 1]$ again. So action $\{2\}$ is unsafe. Therefore safe oracle policy which first runs action 1 for $C_1 n$ number of times for some $C_1 > 0$. Then it runs the safe action 0 for $C_0 n$ number of times (for some $C_0 > 0$) such that it has enough safety budget and then it runs action 2 for $n(1 - (C_0 + C_1))$ number of times. Let the variance of $\sigma^{r,(2)}(0) = 0.001$, $\sigma^{r,(2)}(1) = 0.001$ and $\sigma^{r,(2)}(2) = 0.25$.

The cost cumulative value over rounds for the algorithm for $\alpha = \frac{1}{4}$ is given by

$$\begin{aligned} V_{\mathcal{A}}^c &= (C_1 n)(0.5 + \alpha) + n(1 - C_0 - C_1)0 + (C_0 n)0.5 \\ &= (C_1 n) \cdot \frac{3}{4} + (C_0 n)\frac{2}{4} = \frac{n}{4}(3C_1 + 2C_0). \end{aligned}$$

Then to satisfy the safety budget we have to show that

$$\begin{aligned} & V_{\mathcal{A}}^c \geqslant n(1 - \alpha)0.5 \\ \overset{(a)}{\implies} & \frac{n}{4}(3C_1 + 2C_0) \geqslant \frac{3n}{8} \\ \implies & 3C_1 + 2C_0 \geqslant \frac{3}{2} \end{aligned}$$

Say we just want to satisfy the safety constraint, then setting $C_1 = \frac{1}{4}$ and $C_0 = \frac{3}{8}$ in the above equation we can achieve that. Therefore we have that $T_n(1) = \frac{n}{4}$ and $T_n(0) = \frac{3n}{8}$. This implies that $T_n(2) = n - \frac{n}{4} - \frac{3n}{8} = \frac{3n}{8}$.

Therefore we get that the loss of $\mathbf{b}_*^k$ is given by

$$\mathcal{L}_n(\pi, \mathbf{b}_*^k) = \sum_{a, T_n(a)>0} \frac{\sigma^{r,(2)}(a)}{T_n(a)} = \frac{8(0.001)^2}{3n} + \frac{4(0.001)^2}{n} + \frac{8(0.25)^2}{3n}$$

Now we calculate the loss of the optimal data collection algorithm following the unconstrained $\mathbf{b}_*$. Note that now $T_n^*(0) = \frac{0.001}{0.001+0.001+0.25} n = \frac{n}{252}$, $T_n^*(1) = \frac{n}{252}$ and $T_n^*(2) = \frac{250n}{252}$. Then the loss of the optimal data collection algorithm following $\mathbf{b}_*$ is given by

$$\begin{aligned}\mathcal{L}_n^*(\pi, \mathbf{b}_*) &= \sum_{a, T_n^*(a)>0} \frac{\sigma^{r,(2)}(a)}{T_n^*(a)} = \frac{252(0.001)^2}{n} + \frac{252(0.001)^2}{n} + \frac{252(0.25)^2}{250n} \\ &\approx \frac{2}{4000n} + \frac{15}{n}.\end{aligned}$$

It follows then that the regret scales as

$$\begin{aligned}\mathcal{R}_n &= \mathcal{L}_n(\pi, \mathbf{b}_*^k) - \mathcal{L}_n^*(\pi, \mathbf{b}_*) \\ &= \sum_{a, T_n(a)>0} \frac{\sigma^{r,(2)}(a)}{T_n(a)} - \sum_{a, T_n^*(a)>0} \frac{\sigma^{r,(2)}(a)}{T_n^*(a)} = O\left(\frac{K}{n}\right) \geqslant \mathcal{R}_n^{\text{safe}} = \widetilde{O}(\frac{K \log(n)}{n^{3/2}}).\end{aligned}$$

Note that this regret rate holds for any $C_1 < C_0$ and we cannot shift any more proportion to action $\{2\}$. Therefore the algorithm will choose the sub-optimal safe action $\{0\}$ more than the action that reduces the MSE (to satisfy safety constraint) most resulting in a regret that scales as n^{-1}. So any algorithm (including the safe oracle algorithm) will not be able to achieve the desired regret rate of $\widetilde{O}(n^{-3/2})$. The claim of the proposition follows. □

Remark C.3. (***Tractability condition***) *Let* $\mathbf{b}$ *be any behavior policy that minimizes MSE. However, running* $\mathbf{b}$ *only once is not enough to guarantee a regret of* $\widetilde{O}(n^{-3/2})$. *Let* $\mathbf{b}$ *be run for* K_b *episodes to guarantee a regret of* $\widetilde{O}(n^{-3/2})$. *Note that* K_b *is the number of rounds in the bandit setting. Observe that the*

number of rounds (or episodes in case of MDP) K_b *is behavior policy specific.*

Case 1 (Two action bandits): *Consider two action bandit setting such that* $A = 2$*. Further, let* $\pi(a) = 1/A$ *and the left action has a constraint-value of* C_1 *while the right action has a constraint-value of* C_2*. Let the deterministic baseline policy* π_0 *always choose the left action, while the behavior policy* $\mathbf{b}$ *chooses the right action. Note that* $\mathbf{b}$ *may or may not be* $\mathbf{b}_*$*. Then to satisfy the safety constraint (4.1) we need that*

$$
\begin{aligned}
(n - K_b)C_1 + K_b C_2 \geqslant (1-\alpha)nC_1 &\implies nC_1 - K_b C_1 + K_b C_2 \geqslant nC_1 - \alpha n C_1 \\
&\implies K_b(C_1 - C_2) \leqslant nC_1\alpha \\
&\implies 1 - \frac{C_2}{C_1} \leqslant \frac{n\alpha}{K_b} \\
&\implies \frac{K_b}{\alpha}(1 - \frac{C_2}{C_1}) \leqslant n \\
&\implies n \geqslant \frac{K_b}{\alpha}\left(1 - \frac{C_2}{C_1}\right)
\end{aligned}
$$

The above inequality shows two things, (1) the lower bound to the budget n *to run the behavior policy* $\mathbf{b}$ *for* K_b *rounds and satisfy the safety constraint; (2) The condition* $C_1 > C_2$ *has to be satisfied so that the RHS is positive.*

Case 2 (General multi-armed bandits): *Now generalizing this to* $A \geqslant 2$

we can show that the above condition can be modified into

$$
\begin{aligned}
&(n - K_b)\mu^c(0) + K_b \min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a) \geqslant (1-\alpha) n \mu^c(0) \\
\implies &n\mu^c(0) - K_b\mu^c(0) + K_b \min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a) \geqslant n\mu^c(0) - \alpha n \mu^c(0) \\
\implies &K_b(\mu^c(0) - \min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a)) \leqslant \alpha n \mu^c(0) \\
\implies &1 - \frac{\min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a)}{\mu^c(0)} \leqslant \frac{\alpha n}{K_b} \\
\implies &\frac{K_b}{\alpha}\left(1 - \frac{\min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a)}{\mu^c(0)}\right) \leqslant n \\
\implies &n \geqslant \frac{K_b}{\alpha}\left(1 - \frac{\min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a)}{\mu^c(0)}\right)
\end{aligned}
$$

The above inequality shows two things, (1) the lower bound to the budget n to run the behavior policy $\mathbf{b}$ for K_b rounds and satisfy the safety constraint for a general K_b armed bandit; (2) The condition $\min_{a \in \mathcal{A}\setminus\{0\}} \mu^c(a) < \mu^c(0)$ has to be satisfied so that the RHS is positive.

Case 3 (Tabular MDP): *Define $V_c^{\mathbf{b}^-}(s_1)$ as the value of the policy $\mathbf{b}^-$ starting from state s_1. So this policy $\mathbf{b}^-$ can be thought of as the worst possible policy that can be followed by the agent during an episode. Let this policy be run for K_{b^-} episodes. Also, recall that $V_c^{\pi_0}(s_1)$ is the value of the baseline policy π_0 starting from state s_1. It can easily shown following a similar line of argument as case 2 that we need a budget of*

$$
n \geqslant \frac{K_{b^-}}{\alpha}\left(1 - \frac{V_c^{\mathbf{b}^-}(s_1)}{V_c^{\pi_0}(s_1)}\right).
$$

Again the above inequality shows two things for a general Tree MDP: (1) the lower bound to the budget n to run the behavior policy $\mathbf{b}^-$ for K_{b^-} episodes and satisfy the safety constraint for a Tree MDP; (2) $V_c^{\mathbf{b}^-}(s_1) < V_c^{\pi_0}(s_1)$ so that the RHS is positive.

Now observe that in the first two cases of the bandit setting the $V_c^{\mathbf{b}^-}(s_1)$ yields

$\min_{a\in\mathcal{A}\setminus\{0\}}\mu^c(a)$. *Therefore combining all three cases we can state the budget* $n \geqslant \frac{K_{b^-}}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)$. *Now from (Carpentier and Munos, 2012; Mukherjee et al., 2022a) we know that* $K_{b^-} \geqslant C_\sigma(n-\sqrt{n})$ *where* $C_\sigma \in (0,1]$ *is an MDP dependent parameter that depends on the reward variance of state-action pairs to achieve a regret bound of* $\widetilde{O}(n^{-3/2})$. *We define the quantity* $C_\sigma = \max_{s,a}\frac{\mathbf{b}_*(a|s)}{M(s)}$ *where* $\mathbf{b}_*(a|s)$ *and* $M(s)$ *are defined in (4.4) and (4.5) respectively. Observe that* $C_\sigma \in (0,1)$. *Then we have that*

$$
\begin{aligned}
&n \geqslant \frac{K_{b^-}}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right) \implies n \geqslant \frac{C_\sigma(n-\sqrt{n})}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\\
&\implies n \geqslant \frac{C_\sigma n}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right) - \frac{\sqrt{n}}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\\
&\implies n\left(1-\frac{C_\sigma}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\right) + \frac{\sqrt{n}}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right) \geqslant 0\\
&\implies \sqrt{n}\left(\sqrt{n} - \frac{C_\sigma\sqrt{n}}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right) + \frac{1}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\right) \geqslant 0.
\end{aligned}
$$

This implies that

$$
\begin{aligned}
&\sqrt{n} - \frac{C_\sigma\sqrt{n}}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right) + \frac{1}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right) \geqslant 0\\
&\implies \sqrt{n}\left(1-\frac{C_\sigma}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\right) \geqslant -\frac{1}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\\
&\implies \sqrt{n} \geqslant \frac{-\frac{1}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)}{\left(1-\frac{C_\sigma}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)\right)}\\
&\implies \sqrt{n} \geqslant \frac{\frac{1}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)}{\frac{C_\sigma}{\alpha}\left(1-\frac{V_c^{b^-}(s_1)}{V_c^{\pi_0}(s_1)}\right)-1}.
\end{aligned}
$$

This yields the tractability condition.

C.2 Tractable MDP and Lower Bounds

Some Definitions for proving Lower Bound: These definitions follow similar definitions in Wagenmaker et al. (2022b). Define the Q-function that satisfies the Bellman equation as

$$Q_\ell^\pi(s,a) = R_\ell(s,a) + \sum_{s'} P_\ell\left(s' \mid s,a\right) V_{\ell+1}^\pi\left(s'\right)$$

and $Q_{L+1}^\pi(s,a) = 0$. Define the optimal Q-function as $Q_\ell^{\pi_*}(s,a) \coloneqq \sup_\pi Q_\ell^\pi(s,a), V_\ell^{\pi_*}(s) \coloneqq \sup_\pi V_\ell^\pi(s)$, and let $\pi^\star$ denote an optimal policy. A policy $\widehat{\pi}$ is called ϵ-optimal which satisfies the following

$$V^{\pi_*}(s_1) - V^{\widehat{\pi}}(s_1) \leqslant \epsilon$$

with probability greater than $1-\delta$ using as few episodes as possible. We further define a few more notations for proving the lower bound. Define the suboptimality gap as

$$\Delta_\ell(s,a) \coloneqq V_\ell^{\pi_*}(s) - Q_\ell^{\pi_*}(s,a).$$

such that $\Delta_\ell(s,a)$ denotes the suboptimality of taking action a in (s,h), and then playing the optimal policy henceforth. Define the state-action visitation distribution as:

$$w_\ell^\pi(s,a) \coloneqq \mathbb{P}_\pi\left[s_\ell = s, a_\ell = a\right], \quad w_\ell^\pi(s) \coloneqq \mathbb{P}_\pi\left[s_\ell = s\right].$$

Note that $w_\ell^\pi(s,a) = \pi_\ell(a|s) w_\ell^\pi(s)$. We denote the maximum reachability of (s,ℓ) by

$$W_\ell(s) \coloneqq \sup_\pi w_\ell^\pi(s).$$

This is the maximum probability with which we could hope to reach (s, ℓ). Define the best-policy gap-visitation complexity as $\mathcal{C}^\star(\mathbf{T})$. Finally, recall that tree MDP is a subset of general MDPs which let us restate the following lemmas on lower bound for unconstrained tree MDPs from Wagenmaker et al. (2022b).

Lemma C.4. (***Divergence Lemma, Restatement of Lemma 4.1 from Wagenmaker et al. (2022b)***) *Consider tree MDPs $\mathbf{T}$ and $\mathbf{T}'$ with the same state space $\mathcal{S}$, actions space $\mathcal{A}$, horizon L, and initial state distribution P_0. Fix some $(s, \ell) \in \mathcal{S} \times [L]$, and for any $a \in \mathcal{A}$ let $\nu_\ell(s, a)$ denote the law of the joint distribution of (s', R) where $s' \sim P_{\mathbf{T}}(\cdot \mid s, a)$ and $R \sim R_{\mathbf{T}}(s, a)$. Define the law $\nu'_\ell(s, a)$ analogously with respect to $\mathbf{T}'$. Fix some policy π and let $\mathbb{P}_{\mathbf{T}} = \mathbb{P}_{\nu\pi}$ and $\mathbb{P}_{\mathbf{T}'} = \mathbb{P}_{\nu'\pi}$ be the probability measures on $\mathbf{T}$ and $\mathbf{T}'$ induced by the τ-episode interconnection of π and ν (respectively by π' and ν'). For any almost-sure stopping time τ with respect to filtration $(\mathcal{F}_\tau)$,*

$$\sum_{s,a,h} \mathbb{E}_{\mathbf{T}}\left[N^\tau_\ell(s, a)\right] \mathrm{KL}\left(\nu_\ell(s, a), \nu'_\ell(s, a)\right) \geqslant \sup_{\xi \in \mathcal{F}_\tau} d\left(\mathbb{P}_{\mathbf{T}}(\xi), \mathbb{P}_{\mathbf{T}'}(\xi)\right)$$

where $d(x, y) = x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y}$ and $N^\tau_\ell(s, a)$ denotes the number of visits to (s, a, ℓ) in the τ episodes.

Lemma C.5. (***Proposition 12 from Wagenmaker et al. (2022b)***) *Fix some tree MDP $\mathbf{T}$. Then:*

1. *The set of valid state-action visitation distributions on $\mathbf{T}$ is convex.*
2. *For any valid state-action visitation distribution on $\mathbf{T}$, there exists some policy that realizes it.*

Lemma C.6. (***Restatement of Lemma F.3 from Wagenmaker et al. (2022b)***)

In the tree MDP $\mathbf{T}$*, fix some* $\bar{\ell} \in [L]$*. Then*

$$\mathcal{C}^\star(\mathbf{T}) \leqslant \inf_{\mathbf{b}} \max_{s,a} \frac{1}{w_{\bar{\ell}}^{\mathbf{b}}(s,a)\Delta_\ell(s,a)^2} + \max_{s,\ell} \frac{SAL}{W_\ell(s)}.$$

is the complexity of the Tree MDP $\mathbf{T}$*.*

Lemma C.7. (***Proposition 4 from Wagenmaker et al. (2022b)***) *The following bounds hold for any unconstrained tree MDP* $\mathbf{T}$*:*

1. $\mathcal{C}^\star(\mathbf{T}) \leqslant \frac{L^3SA}{\epsilon^2}$
2. $\mathcal{C}^\star(\mathbf{T}) \leqslant \sum_{\ell=1}^{L} \sum_{s,a} \min\left\{\frac{1}{W_\ell(s)\Delta_\ell(s,a)^2}, \frac{W_\ell(s)}{\epsilon^2}\right\} + \frac{L^2|\mathrm{OPT}(\epsilon)|}{\epsilon^2}$
3. $\mathcal{C}^\star(\mathbf{T}) \leqslant \sum_{\ell=1}^{L} \sum_{s,a} \frac{1}{\epsilon \max\{\Delta_\ell(s,a),\epsilon\}} + \frac{L^2|\mathrm{OPT}(\epsilon)|}{\epsilon^2}.$

where, $\mathcal{C}^\star(\mathbf{T})$ *is the complexity of the Tree MDP* $\mathbf{T}$*. The second term in* $\mathcal{C}^\star(\mathbf{T})$*,* $L^2|\mathrm{OPT}(\epsilon)|/\epsilon^2$*, captures the complexity of ensuring that after eliminating* $\epsilon/W_\ell(s)$*-suboptimal actions, sufficient exploration is performed to guarantee the returned policy is* ϵ*-optimal.*

Lemma C.8. (***Restatement of Theorem 5 in Carpentier and Munos (2012)***) *Let* $A \in \mathbb{N}$ *be a set of actions for a bandit setting. Let* inf *be the infimum taken over all online sampling algorithms that reduce the MSE and* sup *represent the supremum taken over all environments. Define the regret of the algorithm over the target policy* π *as* $\mathcal{R}_n \coloneqq \mathcal{L}_n(\pi) - \mathcal{L}_n^*(\pi)$ *where* $\mathcal{L}_n(\pi)$ *is the MSE of the target policy following the algorithm. Then:*

$$\inf \sup \mathbb{E}\left[\mathcal{R}_n\right] \geqslant C\frac{A^{1/3}}{n^{3/2}},$$

where C *is a numerical constant, and* n *is the total budget,*

Lemma C.9. *Define the regret of the algorithm over the target policy* π *as* $\mathcal{R}_n \coloneqq \mathcal{L}_n(\pi, \mathbf{b}) - \mathcal{L}_n^*(\pi, \mathbf{b}_*)$ *where* $\mathcal{L}_n(\pi, \mathbf{b})$ *is the MSE of the target policy following*

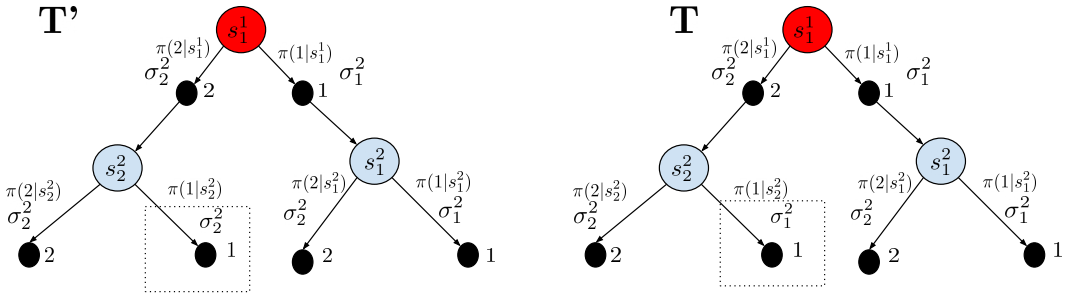


Figure C.4: Tractable Tree MDPs $\mathbf{T}$ and $\mathbf{T}'$. The difference between the two Tree MDPs is highlighted in the square box.

the algorithm and $\mathbf{b}_*$ *is the unconstrained oracle behavior policy. The reward regret in tree MDP* $\mathbf{T}$ *is lower bounded by*

$$\inf \sup \mathbb{E}\left[\mathcal{R}_n\right] \geqslant \Omega\left(\frac{\sqrt{SAL^2 \log(1/\delta)}}{n^{3/2}}\right).$$

Proof. We prove this lemma in two steps. In the first step, we prove the minimum number of episodes required by an ϵ-optimal policy $\mathbf{b}$ in tree MDP $\mathbf{T}$ (fig. C.4) such that $V^{\mathbf{b}_*}(s_1) - V^{\mathbf{b}}(s_1) \leqslant \epsilon$. Next in step 2 we show that given this minimum number of episodes, what is the loss suffered by $\mathbf{b}$ against $\mathbf{b}_*$ at the end of episode K.

Step 1 (Minimum episodes): We consider the two tree MDPs $\mathbf{T}$ and $\mathbf{T}'$ shown in Figure C.4. We will apply Theorem C.4 on our MDP, $\mathbf{T}$, and MDP $\mathbf{T}'$ which is identical to $\mathbf{T}$ except in state $(s_2^2, 1)$ where we have $\sigma^2(s_2^2, 1) = (\mu - \Delta)(1 - \mu + \Delta)$ in $\mathbf{T}'$ and $\sigma^2(s_1^2, 1) = (\mu + \alpha)(1 - \mu - \alpha)$ for $\mathbf{T}$ and some $\Delta > 0$. This yields a different $\mathbf{b}_*$ for MDP $\mathbf{T}$ than $\mathbf{b}_*$ for $\mathbf{T}'$.

Fix some $\bar{\ell} \in [L]$. Since $\mathbf{T}$ and $\mathbf{T}'$ are identical at all points but this one,

we have

$$\sum_{s,a,\ell} \mathbb{E}_{\mathbf{T}}\left[N_\ell^\tau(s,a)\right] \mathrm{KL}\left(\text{Bernoulli}(\mu-\Delta), \text{Bernoulli}(\mu+\alpha)\right)$$
$$= \mathbb{E}_{\mathbf{T}}\left[N_{\bar{\ell}}^\tau(s,a)\right] \mathrm{KL}\left(\text{Bernoulli}(\mu-\Delta), \text{Bernoulli}(\mu+\alpha)\right).$$

where, $\mathbb{E}_{\mathbf{T}}, \mathbb{E}_{\mathbf{T}'}$ denotes the expectation over the data collected in tree MDP $\mathbf{T}$ and $\mathbf{T}'$ respectively following policy $\mathbf{b}_*$.

Let $\mathbf{b}_*$ denote the optimal policy on $\mathbf{T}$, and $\mathbf{b}$ denote the ϵ -optimal policy by any other algorithm. Let the event $\xi = \{\mathbf{b} = \mathbf{b}_*\}$. Since we assume algorithm is δ-correct, and since the optimal policies on $\mathbf{T}$ and $\mathbf{T}'$ differ, we have $\mathbb{P}_{\mathbf{T}}(\xi) \geqslant 1-\delta$ and $\mathbb{P}_{\mathbf{T}'}(\xi) \leqslant \delta$. By Garivier and Kaufmann (2016), we can then lower bound

$$\mathrm{d}\left(\mathbb{P}_{\mathbf{T}}(\xi), \mathbb{P}_{\mathbf{T}'}(\xi)\right) \geqslant \log \frac{1}{2.4\delta}$$

Thus, by Theorem C.4, we have shown that, for any (s,a), $a \neq \mathbf{b}_{*,\bar{\ell}}(s)$,

$$\mathbb{E}_{\mathbf{T}}\left[N_{\bar{\ell}}^\tau(s,a)\right] \geqslant \frac{1}{\mathrm{KL}\left(\text{Bernoulli}(\mu-\Delta), \text{Bernoulli}(\mu+\alpha)\right)} \cdot \log \frac{1}{2.4\delta}$$

For small $\alpha > 0$, we can bound (see e.g. Lemma 2.7 of Tsybakov (2009))

$$\mathrm{KL}\left(\text{Bernoulli}(\mu-\Delta), \text{Bernoulli}(\mu+\alpha)\right) \leqslant 6(\Delta-\alpha)^2.$$

Taking $\alpha \to 0$, we have

$$\mathbb{E}_{\mathbf{T}}\left[N_{\bar{\ell}}^\tau(s,a)\right] \geqslant \frac{1}{6\Delta^2} \cdot \log \frac{1}{2.4\delta}.$$

We can write $\mathbb{E}_{\mathbf{T}}\left[N_{\bar{\ell}}^\tau(s,a)\right] = \mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^\tau w_{\bar{\ell}}^{\mathbf{b}^k}(s,a)\right]$ where $\mathbf{b}^k$ denotes the policy the algorithm played at episode k. Note that all state-visitation distributions lie in a convex set in $[0,1]^{SA}$ and that for any valid state-

visitation distribution, there exists some policy that realizes it, by Theorem C.5. By Caratheodory's Theorem, it follows that there exists some set of policies Π with $|\Pi| \leqslant SA + 1$ such that, for any $\mathbf{b}$ and all $s, a, w_{\bar{\ell}}^{\mathbf{b}}(s,a) = \sum_{\mathbf{b}' \in \Pi} \lambda_{\mathbf{b}'} w_{\bar{\ell}}^{\mathbf{b}'}(s,a)$, for some $\lambda \in \triangle_\Pi$. Note that λ is a distribution over the policies in Π. Letting λ^k denote this distribution satisfying the above inequality for $\mathbf{b}^k$, it follows that

$$\begin{aligned}
\mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau} w_{\bar{\ell}}^{\mathbf{b}^k}(s,a)\right] &= \mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau}\sum_{\mathbf{b}\in\Pi} \lambda_{\mathbf{b}}^k w_{\bar{\ell}}^{\mathbf{b}}(s,a)\right] \\
&= \sum_{\mathbf{b}\in\Pi} \mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau} \lambda_{\mathbf{b}}^k\right] w_{\bar{\ell}}^{\mathbf{b}}(s,a) \\
&= \mathbb{E}_{\mathbf{T}}[\tau] \sum_{\mathbf{b}\in\Pi} \frac{\mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau} \lambda_{\mathbf{b}}^k\right]}{\mathbb{E}_{\mathbf{T}}[\tau]} w_{\bar{\ell}}(s,a).
\end{aligned}$$

Note that $\sum_{\mathbf{b}\in\Pi} \mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau} \lambda_{\mathbf{b}}^k\right] = \mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau}\sum_{\mathbf{b}\in\Pi} \lambda_{\mathbf{b}}^k\right] = \mathbb{E}_{\mathbf{T}}[\tau]$ so it follows that $\left(\frac{\mathbb{E}_{\mathbf{T}}\left[\sum_{k=1}^{\tau} \lambda_{\mathbf{b}}^k\right]}{\mathbb{E}_{\mathbf{T}}[\tau]}\right)_{\mathbf{b}\in\Pi} \in \triangle_\Pi$. Thus, a δ-correct algorithm must satisfy, for all s, a and some $\lambda \in \triangle_\Pi$,

$$\mathbb{E}_{\mathbf{T}}[\tau] \geqslant \frac{1}{6\Delta^2 \cdot \sum_{\mathbf{b}\in\Pi} \lambda_{\mathbf{b}} w_{\bar{\ell}}^{\mathbf{b}}(s,a)} \cdot \log\frac{1}{2.4\delta}.$$

Since the set of state visitation distributions is convex, and since for any state-visitation distribution we can find some policy realizing that distribution, for any $\lambda \in \triangle_\Pi$, it follows that there exists some $\mathbf{b}'$ such that, for all s, a, $\sum_{\mathbf{b}\in\Pi} \lambda_{\mathbf{b}} w_{\bar{\ell}}^{\mathbf{b}}(s,a) = w_{\bar{\ell}}^{\mathbf{b}'}(s,a)$. So, we need, for all s, a

$$\mathbb{E}_{\mathbf{T}}[\tau] \geqslant \frac{1}{6\Delta^2 \cdot w_{\bar{\ell}}^{\mathbf{b}}(s,a)} \cdot \log\frac{1}{2.4\delta}.$$

It follows that every δ-correct algorithm must satisfy

$$\begin{aligned}\mathbb{E}_{\mathbf{T}}[\tau] &\geqslant \inf_{\mathbf{b}} \max_{s,a} \frac{1}{6\Delta^2 \cdot w_\ell^{\mathbf{b}}(s,a)} \cdot \log \frac{1}{2.4\delta}, \\ &\gtrsim \mathcal{C}^\star(\mathbf{T}) \cdot \log \frac{1}{2.4\delta} - \max_{s,\ell} \frac{SAL}{W_\ell(s)}\end{aligned}$$

from which the first inequality follows, and the second inequality follows from Theorem C.6.

The second term in $\mathcal{C}^\star(\mathbf{T})$, $L^2|\mathrm{OPT}(\epsilon)|/\epsilon^2$, captures the complexity of ensuring that after eliminating $\epsilon/W_\ell(s)$-suboptimal actions, sufficient exploration is performed to guarantee the returned policy is ϵ-optimal. Using Theorem C.7 we have that $\mathcal{C}^\star(\mathbf{T})$, $L^2|\mathrm{OPT}(\epsilon)|/\epsilon^2$ will be no worse than L^3SA/ϵ^2, it could be much better, if in the MDP the number of (s,a,ℓ) with $\Delta_\ell(s,a) \lesssim \epsilon/W_\ell(s)$ is small (note that since $\Delta_\ell(s,a) \geqslant \Delta_{\min}(s,\ell)$ by definition, $\mathrm{OPT}(\epsilon)$ will only contain states for which the minimum non-zero gap is less than $\epsilon/W_\ell(s)$). Wagenmaker et al. (2022b) obtains the bounds on $\mathcal{C}^\star(\mathbf{T})$ in Theorem C.7, providing an interpretation of $\mathcal{C}^\star(\mathbf{T})$ in terms of the maximum reachability, and illustrating $\mathcal{C}^\star(\mathbf{T})$ is no larger than the minimax optimal complexity. This implies that

$$\mathbb{E}_{\mathbf{T}}[\tau] \gtrsim \Omega\left(\frac{SAL^2}{\epsilon^2}\log(1/\delta)\right).$$

Hence the $V^{\mathbf{b}^K}(s_1^1) - V^{\mathbf{b}_*}(s_1^1) \leqslant \epsilon$ for $K \geqslant \frac{SAL^2}{\epsilon^2}\log(1/\delta)$.

Step 2 (Bound regret in T): In the $\mathbf{T}$ in Figure C.4 we now have

$$M(s_1^1) = \sqrt{2\sigma_1^2 + \sigma_2^2} + \sqrt{2\sigma_2^2 + \sigma_1^2}, \quad M(s_1^2) = M(s_2^2) = \sigma_1 + \sigma_2$$

Define confidence interval $\beta_L^K = L\sqrt{SA\log(SAL^2/\delta)/n}$. It can be shown

using pointwise uncertainty estimation from Corollary 3 that

$$|\widehat{\sigma}_{K,1} - \sigma_1| \leqslant \beta_L^K, \quad |\widehat{\sigma}_{K,2} - \sigma_2| \leqslant \beta_L^K \tag{C.2}$$

holds with probability greater than $1-\delta$, where the $\widehat{\sigma}_{K,1}$, $\widehat{\sigma}_{K,2}$ denote the estimated variances after K episodes. Then the loss of the agnostic algorithm at the end of the K-th episode is given by

$$\begin{aligned}
\mathcal{L}_n^K(\pi, \mathbf{b}) &= \frac{\sqrt{2\widehat{\sigma}_{K,1}^2 + \widehat{\sigma}_{K,2}^2} + \sqrt{2\widehat{\sigma}_{K,2}^2 + \widehat{\sigma}_{K,1}^2}}{n} \\
&\overset{(a)}{\geqslant} \frac{\sqrt{2(\sigma_1^2 - \beta_L^K) + \sigma_2^2 - \beta_L^K} + \sqrt{2(\sigma_2^2 - \beta_L^K) + \sigma_1^2 - \beta_L^K}}{n} \\
&= \frac{\sqrt{2\sigma_1^2 + \sigma_2^2 - 3\beta_L^K} + \sqrt{2\sigma_2^2 + \sigma_1^2 - 3\beta_L^K}}{n} \\
&\overset{(b)}{\geqslant} \frac{\sqrt{2\sigma_1^2 + \sigma_2^2} + \sqrt{2\sigma_2^2 + \sigma_1^2}}{n} - C\frac{\beta_L^K}{n}
\end{aligned}$$

where, (a) follows from concentration inequality in (C.2), and (b) follows for some appropriate constant $C > 0$. Then for $K \geqslant \frac{SAL^2}{\epsilon^2}\log(1/\delta)$ (from step 1) we have the total loss as

$$\begin{aligned}
\mathcal{L}_n(\pi, \mathbf{b}) = \mathcal{L}_n^K(\pi, \mathbf{b}) &\geqslant \left(\frac{\sqrt{2\sigma_1^2 + \sigma_2^2} + \sqrt{2\sigma_2^2 + \sigma_1^2}}{n} - \frac{\beta_L^K}{n}\right)\frac{SAL^2}{\epsilon^2}\log(1/\delta) \\
&\overset{(a)}{\geqslant} \underbrace{\frac{\sqrt{2\sigma_1^2 + \sigma_2^2} + \sqrt{2\sigma_2^2 + \sigma_1^2}}{n}}_{\mathcal{L}_n(\pi, \mathbf{b}_*)} + \frac{\beta_L^K}{n} \\
\mathcal{L}_n(\pi, \mathbf{b}) - \mathcal{L}_n^*(\pi, \mathbf{b}_*) &\overset{(b)}{\geqslant} \frac{\sqrt{SAL^2\log(SAL^2/\delta)}}{n\sqrt{K}} = \Omega\left(\frac{\sqrt{SAL^2\log(1/\delta)}}{n^{3/2}}\right)
\end{aligned}$$

where, (a) follows by first setting $\epsilon = 1/\sqrt{n}$ and then noting that

$$\left(\sqrt{2\sigma_1^2+\sigma_2^2}+\sqrt{2\sigma_2^2+\sigma_1^2}-\beta_L^K\right)(SAL^2)\log(1/\delta) \geqslant \frac{\sqrt{2\sigma_1^2+\sigma_2^2}+\sqrt{2\sigma_2^2+\sigma_1^2}}{n}+\frac{\beta_L^K}{n}.$$

Also note that $\mathcal{L}_n(\pi,\mathbf{b}_*) = \frac{\sqrt{2\sigma_1^2+\sigma_2^2}+\sqrt{2\sigma_2^2+\sigma_1^2}}{n}$. The (b) follows by substituting the value of β_L^K. The claim of the lemma follows. □

Theorem 1. (***Lower Bound, formal***) *Let $\pi(a|s) = \frac{1}{A}$ for each state $s \in \mathcal{S}$. Assume the MDP $\mathbf{M}$ is tractable under Assumption 6 and satisfies (4.7). Then the reward regret is lower bounded by*

$$\mathbb{E}[\mathcal{R}_n] = \mathcal{L}_n(\pi,\mathbf{b}) - \mathcal{L}_n^*(\pi,\mathbf{b}_*^k) \geqslant \begin{cases} \Omega\left(\max\left\{\frac{A^{1/3}}{n^{3/2}}, \left(\frac{H_{*,(1)}^2 A^{2/3}}{n^{3/2}}\right)\right\}\right), & (\textbf{\textit{MAB}}) \\ \Omega\left(\max\left\{\frac{\sqrt{SAL^2}}{n^{3/2}}, \left(\frac{H_{*,(1)}^2 SAL^2}{n^{3/2}}\right)\right\}\right) & (\textbf{\textit{Tabular MDP}}) \end{cases}$$

where, $\Delta_0 = |V_c^{\mathbf{b}_*^k}(s_1^1) - V_c^{\pi_0}(s_1^1)|$ and $H_{*,(1)} = \frac{1}{\alpha V_c^{\pi_0}(s_1^1)}(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)$ is the hardness parameter.

Proof. We follow a reduction-based proof technique to prove this lower bound (Yang et al., 2021b).

Step 1 (Reduction): First recall we have that the regret for any online algorithm **Alg** that minimizes the MSE $\mathcal{L}_n(\pi)$ is given by $\mathcal{R}_n(\textbf{Alg}) = \mathcal{L}_n(\pi,\mathbf{b}) - \mathcal{L}_n^*(\pi,\mathbf{b}_*^k)$, where $\mathcal{L}_n^*(\pi,\mathbf{b})$ is the MSE of the oracle algorithm. We also assume $\pi(a) = 1/A$ for all $a \in \mathcal{A}$, and $\sigma(a) \geqslant \frac{1}{16}$ for all a.

Now consider any sequential decision-making problem $\mathfrak{A}$ (for instance a multi-armed bandit problem) such that there exists $\xi \in \mathbb{R}$ (a constant solely depending on the sequential decision-making problem, e.g., the number of actions in bandits, or state-action-horizon in tabular RL), an instance of problem $\mathfrak{A}$ where for the budget n large enough and any

algorithm **Alg** we have from Theorem C.8 and Theorem C.9 that:

$$\mathbb{E}\left[\mathcal{R}_n^{\mathfrak{A}}(\mathbf{Alg})\right] \geqslant \frac{\xi}{n^{3/2}}, \tag{C.3}$$

For instance, in the MAB case $\xi = A^{1/3}$ with A the number of arms and in tabular RL $\xi = SAL^2$. Using this non-conservative (unconstraint) lower bound, we show our lower bound for the safe setting for the problem $\mathfrak{A}$ with a baseline policy π_0. We assume the MDP $\mathbf{T} \subset \mathbf{M}$ where we run the behavior policy $\mathbf{b}_*^k$ satisfies Assumption 6. This is required because otherwise we will not be able to run the behavior policy a sufficient number of times to reach a regret bound of $\widetilde{O}(n^{-3/2})$(see Proposition 1). To do so, let's consider any safe algorithm (that is to say it satisfies safety constraint) noted as $\mathbf{Alg}_c$. We assume this algorithms selects behavior policies $(\mathbf{b}^t)_{t\in[n]}$ and let $\mathcal{N}_0$ denotes the set of episodes in $\{1,\dots,K\}$ where $\mathbf{Alg}_c$ selects the safe policy π_0. Let $|\mathcal{N}_0| = N_0$ and $\Delta_0 \coloneqq |V^{\mathbf{b}_*^k} - V_c^{\pi_0}|$. Here we assume the budget n is large such that $n \geqslant SAL^2/\epsilon^2$ for some $\epsilon > 0$ (see Theorem C.9) and

$$\begin{aligned}
n &\geqslant \sqrt{\frac{\xi}{\alpha V_c^{\pi_0}(s_1^1)\cdot\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)} + \frac{\xi^2}{4\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)^2}} \\
\implies n^2 &\geqslant \frac{\xi}{\alpha V_c^{\pi_0}(s_1^1)\cdot\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)} + \frac{\xi^2}{4\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)^2}. \\
\implies n &\geqslant \frac{\xi}{n\alpha V_c^{\pi_0}(s_1^1)\cdot\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)} + \frac{\xi^2}{4n\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)^2}
\end{aligned}$$

Step 2 (Loss estimate): Let $\mathcal{L}(N_0)$ be the loss suffered in first N_0 episodes. We now distinguish two cases:

(a) If $\mathbb{E}\left[\mathcal{L}(N_0)\right] \geqslant \frac{\xi}{n\alpha V_c^{\pi_0}(s_1^1)\cdot\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)}$, then the definition of the

regret implies that:

$$\mathbb{E}\left[\mathcal{R}_n^{\mathfrak{A}}(\mathbf{Alg})\right] = \mathbb{E}\left[\mathcal{L}(N_0)\right] \cdot \Delta_0 \geqslant \frac{\xi \Delta_0}{n\alpha V_c^{\pi_0}(s_1^1) \cdot (\alpha V_c^{\pi_0}(s_1^1) + \Delta_0)}. \tag{C.4}$$

(b) If $\mathbb{E}\left[\mathcal{L}(N_0)\right] < \frac{\xi}{n\alpha V^{\pi 0} \cdot \left(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0\right)}$, then let's note $\mathcal{N}_0^C = \left\{i_1, i_2, \cdots, i_{|\mathcal{N}_0^c|}\right\}$ the set of episodes where $\mathbf{Alg}_c$ does not execute the baseline policy π_0. Now consider the safety budget (similar to Definition 1 of Yang et al. (2021b)) we have:

$$\begin{aligned}
B_{\mathcal{N}_0^c}\left(\mathbf{Alg}_c\right) &= \max_{t \in \mathcal{N}_0^c} \mathbb{E} \sum_{k=1}^{t} \left[(1-\alpha) V_c^{\pi_0}(s_1^1) - V^{\pi^t}(s_1^1)\right] \\
&= \max_{t \in \mathcal{N}_0^c} \mathbb{E} \sum_{k=1}^{t} \left[V^{\mathbf{b}_*^k}(s_1^1) - V^{\pi^t}(s_1^1) - \alpha V_c^{\pi_0}(s_1^1) - \left(V_c^{\mathbf{b}_*^k}(s_1^1) - V_c^{\pi_0}(s_1^1)\right)\right] \\
&= \max_{t \in \mathcal{N}_0^c} \mathbb{E}\left[\mathcal{R}_{\mathcal{N}_0^C}^{\mathfrak{A}}\left(\mathcal{A}_c\right)(t)\right] - \left(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0\right) t,
\end{aligned}$$

where $\Delta_0 = V_c^{\mathbf{b}_*^k}(s_1^1) - V_c^{\pi_0}(s_1^1)$ is the difference between the constraint value of the optimal policy and the baseline policy and $\mathbb{E}\left[R_{\mathfrak{A}}^{\mathcal{N}_0^C}\left(\mathcal{A}_c\right)(t)\right]$ is the regret incurred by the episodes $\{i_k\}_{k \in [t]}$. Therefore, for any $t \in [|\mathcal{T}_0^c|]$, by (C.3) we have that there exists an instance u (for instance in a bandit problem u is the means of each arm) of $\mathfrak{A}$ such that $\mathbb{E}\left[R_{\mathfrak{A}}^{\mathcal{N}_0^C}\left(\mathcal{A}_c\right)(t)\right] \geqslant \frac{\xi}{t^{3/2}}$. Let $t_0 = \frac{\xi^2}{4n\left(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0\right)^2}$, then there exists an instance such that

$$\begin{aligned}
B_{\mathcal{N}_0^C}\left(\mathbf{Alg}_c\right) &\geqslant \frac{\xi}{t_0^{3/2}} - \left(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0\right) t_0 \\
&= \frac{4\left(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0\right)^3 n^{3/2}}{\xi^2} - \frac{\xi^2}{4(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0)} \frac{1}{n^2} \\
&\overset{(a)}{\gtrsim} \frac{(\alpha V_c^{\pi_0}(s_1^1) + \Delta_0)^2 \xi^2}{n^{3/2}}.
\end{aligned}$$

where, (a) follows as $n^{3/2} - n^{-2} \geqslant n^{-3/2}$. Combining the safety condition

in eq. (4.1), we have

$$\mathbb{E}\left[\mathcal{L}(\mathcal{N}_0)\right] \geqslant \frac{B_{\mathcal{N}_0}\left(\mathbf{Alg}_c\right)}{\alpha V_c^{\pi_0}(s_1^1)} \gtrsim \frac{\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)^2 \xi^2}{\alpha V_c^{\pi_0}(s_1^1) n^{3/2}}.$$

By the same derivation of eq. (C.4), we have

$$\mathbb{E}\left[R_n^{\mathfrak{A}}(\mathbf{Alg})\right] \gtrsim \frac{\xi^2\Delta_0}{n\alpha V_c^{\pi_0}(s_1^1)\cdot(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)} \overset{(a)}{\geqslant} \frac{\xi^2}{n^{3/2}\alpha V_c^{\pi_0}(s_1^1)\cdot(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)}. \tag{C.5}$$

where, (a) follows for $\Delta_0 \geqslant 1/\sqrt{n}$. Combining eq. (C.3), C.4, and C.5 we can show that

$$\mathbb{E}\left[R_n^{\mathfrak{A}}(\mathbf{Alg})\right] \gtrsim \max\left\{\frac{\xi}{n^{3/2}}, \frac{(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)^2\xi^2}{(\alpha V_c^{\pi_0}(s_1^1))^2 n^{3/2}}\right\}.$$

Step 3 (Combine with MAB:) Now considering that safe oracle $\mathbf{b}_*^k$ is also an online algorithm **Alg**, we can drop the notation. Then for multi-armed bandits, by Theorem C.8, we choose $\xi = A^{1/3}$. Then we have

$$\begin{aligned}\mathbb{E}\left[\mathcal{R}_n\right] &\gtrsim \max\left\{\frac{A^{1/3}}{n^{3/2}}, \frac{\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)^2}{(\alpha V_c^{\pi_0}(s_1^1))^2}\left(\frac{A^{2/3}}{n^{3/2}}\right)\right\} \\ &\overset{(a)}{=} \min\left\{\frac{A^{1/3}}{n^{3/2}}, \left(\frac{H_{*,(1)}^2 A^{2/3}}{n^{3/2}}\right)\right\}.\end{aligned}$$

where, (a) follows from the problem complexity parameter $H_{*,(1)} = \frac{1}{\alpha V_c^{\pi_0}(s_1^1)}(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)$ when $\pi(a) = 1/A$ and $\sigma(a) \geqslant 1/16$ for the bandit setting.

Step 4 (Combine with tabular RL:) For tabular RL, by Theorem C.9,

we choose $\xi = \sqrt{SAL^2}$. Then we have

$$\begin{aligned}\mathbb{E}\left[\mathcal{R}_n\right] &\gtrsim \max\left\{\frac{\sqrt{SAL^2}}{n^{3/2}}, \frac{\left(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0\right)^2}{(\alpha V_c^{\pi_0}(s_1^1))^2}\left(\frac{SAL^2}{n^{3/2}}\right)\right\} \\ &\overset{(a)}{=} \min\left\{\frac{\sqrt{SAL^2}}{n^{3/2}}, \left(\frac{H_{*,(1)}^2 SAL^2}{n^{3/2}}\right)\right\}.\end{aligned}$$

where, (a) follows from the problem complexity parameter

$$H_{*,(1)} = \frac{1}{\alpha V_c^{\pi_0}(s_1^1)}(\alpha V_c^{\pi_0}(s_1^1)+\Delta_0)$$

when $\pi(a) = 1/A$ and $\sigma(a) \geqslant 1/16$. This concludes the proof. □

Remark C.10. (***Comparing regret***) *Observe that the regret lower bound is proved on $\mathcal{R}'_n = \mathcal{L}_n(\pi) - \mathcal{L}^*_n(\pi)$ which assumes that we can exactly solve for the oracle sampling solution. However, $\overline{\mathcal{L}}^*_n(\pi)$ in $\mathcal{R}_n$ is an upper bound to $\mathcal{L}^*_n(\pi)$ and so we cannot directly compare $\mathcal{R}_n$ with $\mathcal{R}'_n$. However, since $\mathcal{R}'_n$ gives a lower bound by directly solving for the oracle solution, we conjecture that this is the lower bound to $\mathcal{R}_n$. Proving this conjecture we leave it to future works.*

C.3 Proof of Tree Agnostic MSE

Theorem 2. (***formal***) *Let Assumption 6 hold. Then the MSE of the SaVeR for $\frac{n}{\log(SAn(n+1)/\delta)} \geqslant 32(LSA^2)^2 + \frac{SA}{\min_{s,a}\Delta^{c,(2)}(s,a)} + \frac{1}{4H^2_{*,(2)}}$ is bounded by*

$$\begin{aligned}\mathcal{L}_n(\pi, \widehat{b}^k) \leqslant &\frac{M^2(s_1^1)}{n} + \frac{8AM^2(s_1^1)}{n^2} + \frac{16A^2M^2(s_1^1)}{n^3} \\ &+ \frac{M^2(s_1^1)}{n}\left(32MLSA + H_{*,(2)}\right)^2 + 2\sum_{t=1}^{n}\frac{2\eta+4\eta^2}{n^2} \\ &+ O\left(\frac{(2\eta+4\eta^2)(LSA^2)^2H^2_{*,(2)}M^2\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}^{k,(3/2)}_{*,\min}(s)n^{3/2}}\right)\end{aligned}$$

with probability $(1-\delta)$. *The* $M = \sum_{\ell=1}^{L} \sum_{s_j^\ell} M(s_j^\ell)$, *and* $H_{*,(2)} = \sum_{\ell=1}^{L} \sum_{s_j^\ell} H_{*,(2)}(s_j^\ell)$ *is the problem complexity parameter. The total predicted constraint violations is bounded by*

$$\mathcal{C}_n(\pi, \widehat{b}^k) \leqslant \frac{H_{*,(2)}}{2} \frac{n}{M_{\min}} + 16LSA^2 + O\left(\frac{(2\eta + 4\eta^2)(LSA^2)^2 H_{*,(2)}^2 M^2 \sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n^{1/2}} \right)$$

with probability $(1-\delta)$, *where* $M_{\min} \coloneqq \min_s M(s)$.

Proof. **Step 1 (Sampling rule):** First note that agnostic SaVeR samples by the following rule

$$\text{Play } \mathbf{b}^k = \begin{cases} \pi_x & \text{if } \widehat{Z}^{k-1} \geqslant 0, k \leqslant \sqrt{K} \\ \widehat{b}^k & \text{if } \widehat{Z}^{k-1} \geqslant 0, k > \sqrt{K} \\ \pi_0 & \text{if } \widehat{Z}^{k-1} < 0 \end{cases} \tag{C.6}$$

where, $\widehat{Z}_L^{k-1} \coloneqq \sum_{k'=1}^{k-1} (Y_{c,L}^{\mathbf{b}^{k'}}(s_1^1) - \beta_L^{k'}(s,a)) - (1-\alpha)(k-1)V_c^{\pi_0}(s_1^1)$ is the safety budget till the k-th episode.

Step 2 (MSE Decomposition): Now recall that the agnostic algorithm does not know the variances and the means. We define the good cost event when the oracle has a good estimate of the cost mean. This is stated as follows:

$$\xi_{c,K} \coloneqq \bigcap_{\substack{1 \leqslant k \leqslant K, \\ 1 \leqslant a \leqslant A, 1 \leqslant s \leqslant S}} \left\{ \left| \widehat{\mu}_{c,L}^k(s,a) - \mu^c(s,a) \right| \leqslant (2\eta + 4\eta^2) L \sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_L^k(s,a)}} \right\} \tag{C.7}$$

where, $n = KL$ and K is the number of episodes and L is the length of horizon of each episode. The exploration policy π_x results in a good constraint estimate of state-action tuples. This is shown in Corollary 4. We

define the good variance event as

$$\xi_{\nu,K} \coloneqq \bigcap_{\substack{1\leqslant k\leqslant K,\\ 1\leqslant a\leqslant A, 1\leqslant s\leqslant S}} \left\{ |\widehat{\sigma}_L^k(s,a) - \sigma(s,a)| \leqslant (2\eta + 4\eta^2) L \sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_L^k(s,a)}} \right\}. \tag{C.8}$$

We define the safety budget event

$$\xi_{Z,K} \coloneqq \bigcap_{1\leqslant k\leqslant K} \left\{ \widehat{Z}^k \geqslant 0 \right\}. \tag{C.9}$$

Using the definition of MSE, and Theorem C.1 we can show that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \mathbb{I}\{\xi_{\nu,K}\} \right] \\
&\leqslant \sum_a \pi^2(a|s_1^1) \left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)} \right] \mathbb{E}[T_L^K(s_1^1,a) \mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \mathbb{I}\{\xi_{\nu,K}\}] \\
&\quad + \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{s_j^2} P(s_j^2|s_1^1,a) \mathrm{Var}[Y_n(s_j^2)] \mathbb{E}[T_L^K(s_j^2,a) \mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \mathbb{I}\{\xi_{\nu,K}\}] \\
&\leqslant \sum_a \pi^2(a|s_1^1) \left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)} \right] \mathbb{E}[T_L^K(s_1^1,a) \mathbb{I}\{\xi_{Z,K} \cap \mathbb{I}\{\xi_{\nu,K}\}\} \cap \mathbb{I}\{\xi_{c,K}\}] \\
&\quad + \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{\ell=2}^{L} \sum_{s_j^\ell} P(s_j^\ell|s_1^1,a) \sum_{a'} \pi^2(a'|s_j^\ell) \left[\frac{\sigma^2(s_j^\ell,a')}{\underline{T}_L^{(2),K}(s_j^\ell,a')} \right] \cdot \\
&\qquad \mathbb{E}[T_L^K(s_j^\ell,a') \mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \mathbb{I}\{\xi_{\nu,K}\}]
\end{aligned} \tag{C.10}$$

which implies that SaVeR does not need to know the reward means $\mu(s,a)$. Hence, the MSE of SaVeR is bounded by

$$
\begin{aligned}
\mathcal{L}_n(\pi) \leqslant & \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \xi_{\nu,K}\right]}_{\textbf{Part A, } \widehat{Z}_n \geqslant 0\textbf{, safety event holds}} \\
& + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{Z,K}^C\}\right]}_{\textbf{Part B, } \widehat{Z}_n < 0\textbf{, constraint violation}} \\
& \quad + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{c,K}^C\}\right]}_{\textbf{Part C, Safety event does not hold}} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{\nu,K}^C\}\right]}_{\textbf{Part D, Variance event does not hold}} \\
& \leqslant \sum_a \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1, a)}{\underline{T}_L^{(2),K}(s_1^1, a)}\right] \mathbb{E}[T_L^K(s_1^1, a)\mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \mathbb{I}\{\xi_{\nu,K}\}] \\
& \quad + \gamma^2 \sum_a \pi^2(a|s_1^1) \sum_{\ell=2}^{L} \sum_{s_j^\ell} P(s_j^\ell|s_1^1, a) \sum_{a'} \pi^2(a'|s_j^\ell)\left[\frac{\sigma^2(s_j^\ell, a')}{\underline{T}_L^{(2),K}(s_j^\ell, a')}\right] \cdot \\
& \qquad \mathbb{E}[T_L^K(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\}] \\
& \quad + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{Z,K}^C\}\right]}_{\textbf{Part B, } \widehat{Z}_n < 0\textbf{, constraint violation}} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{c,K}^C\}\right]}_{\textbf{Part C, Safety event does not hold}} \\
& \quad + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{\nu,K}^C\}\right]}_{\textbf{Part D, Variance event does not hold}} .
\end{aligned}
$$

Divide the total budget n into two parts, n_f when $\sum_{j=1}^{k} \mathbb{I}\{\widehat{Z}^j \geqslant 0\}$ is true, then $\mathbf{b}_*$ or π_x is run. Hence define

$$
n_f \coloneqq \sum_{k=1}^{K} \sum_{\ell=1}^{L} \sum_{s_j^\ell} \sum_{a'=1}^{A} \mathbb{E}[T_\ell^k(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap \mathbb{I}\{\xi_{\nu,K}\}].
$$

The other part consist of $n_u = n - n_f$ number of samples when $\sum_{j=1}^{k} \mathbb{I}\{\widehat{Z}^k <$

$0\}$ and only π_0 is run. Hence we define,

$$n_u = \sum_{k=1}^{K} \sum_{\ell=1}^{L} \sum_{s_j^\ell} \sum_{a'=1}^{A} \mathbb{E}[T_\ell^k(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}^C\}].$$

Step 3 (Sampling of SaVeR for $\widehat{Z}^k \geqslant 0$): First note that when $\widehat{Z}^k \geqslant 0$ the SaVeR samples at episode k and round $\ell+1$ the action $\arg\max_a U_{\ell+1}^k(s_i^{\ell+1}, a)$ where

$$U_\ell^k(s_i^\ell, a) \coloneqq \frac{\widehat{b}_\ell^k(a|s_i^\ell)}{T_\ell^k(s_i^\ell, a)} \leqslant \frac{\pi(a|s_i^\ell)}{T_\ell^k(s_i^\ell, a)} \Bigg(\sigma(s_i^\ell, a) + (2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_\ell^k(s_i^\ell, a)}} + \underbrace{\gamma^2 \sum_{a'} \pi(a'|s_i^\ell) \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell|a')\widehat{M}(s_j^{\ell+1})}_{\mathbf{B}(s_i^\ell)} \Bigg). \tag{C.11}$$

Let $\ell + 1 > 2SA$ be the time at which a given state-action (s_i^ℓ, p') is visited for the last time, i.e., $T_\ell^k(p') = T_L^K(p') - 1$ and $T_{\ell+1}^k(p') = T_L^K(p')$. Note that as $n = KL \geqslant 4SA$, there is at least one state-action pair (s_i^ℓ, p') such that this happens, i.e. such that it is visited after the initialization phase. Note that under Assumption 6 it is possible to visit each (s, a) atleast once. Since the SaVeR chooses to visit (s_i^ℓ, p') at time $\ell+1$, we have for any state-action pair (s_i^ℓ, p')

$$U_{\ell+1}^k(s_i^{\ell+1}, p) \leqslant U_{\ell+1}^k(s_i^{\ell+1}, p'). \tag{C.12}$$

From (C.11) and using the fact that $T_\ell^k(s_i^\ell, p') = T_L^K(s_i^\ell, p') - 1$, we can

show that

$$U^k_{\ell+1}(s_i^{\ell+1}, p') \leqslant \frac{\mathbf{b}_*(p'|s_i^{\ell+1})}{T^k_t(s_i^{\ell+1}, p')}\left((2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T^k_t(s_i^{\ell+1}, p') - 1}} + \mathbf{B}(s_i^{\ell+1})\right)$$
$$= \frac{\mathbf{b}_*(p'|s_i^{\ell+1})}{T^K_L(s_i^{\ell+1}, p') - 1}\left((2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T^K_L(s_i^{\ell+1}, p') - 1}} + \mathbf{B}(s_i^{\ell+1})\right). \tag{C.13}$$

Also note that

$$U^k_{\ell+1}(s_i^{\ell+1}, p) = \frac{\mathbf{b}_*(p|s_i^{\ell+1})}{T^k_t(s_i^{\ell+1}, p)}\left((2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T^k_t(s_i^{\ell+1}, p) - 1}} + B(s_i^{\ell+1})\right)$$
$$\overset{(a)}{\geqslant} \frac{\mathbf{b}_*(p|s_i^{\ell+1})}{T^K_L(s_i^{\ell+1}, p)}. \tag{C.14}$$

where, (a) follows as $T_t(p) \leqslant T^K_L(p, s_i^{\ell+1})$ (i.e., the number of times p has been visited can only increase after time ℓ). Combining (C.12), (C.13), (C.14) we can show that for any action p:

$$\frac{\mathbf{b}_*(p|s_i^{\ell+1})}{T^K_L(p, s_i^{\ell+1})} \leqslant \frac{\mathbf{b}_*(p'|s_i^{\ell+1})}{T^K_L(p', s_i^{\ell+1}) - 1}\left((2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T^K_L(s_i^{\ell+1}, p') - 1}} + \mathbf{B}(s_i^{\ell+1})\right). \tag{C.15}$$

Note that in the above equation, there is no dependency on ℓ, and thus, the probability that (C.15) holds for any $(s_i^{\ell+1}, p)$ and for any $(s_i^{\ell+1}, p')$ such that action $(s_i^{\ell+1}, p')$ is visited after the initialization phase, i.e., such that $T^K_L(s_i^{\ell+1}, p') > 2$ depends on the probability of event $\xi_{Z,n}$.

Step 4. ((Lower bound on $T^K_L(s_i^\ell, p)$ for $\widehat{Z}^k \geqslant 0$): If a state-action tuple (s_i^ℓ, p) is less visited compared to its optimal allocation without taking into account the initialization phase, i.e., $T^K_L(s_i^\ell, p) - 2 < \mathbf{b}(p|s_i^\ell)(n - 2A)$, then from the constraint $\sum_{p'}\left(T^K_L(s, p') - 2\right) = n - 2SA$ and the definition of the optimal allocation, we deduce that there exist at least another state-

action tuple s_i^ℓ, p' that is over-visited compared to its optimal allocation without taking into account the initialization phase, i.e., $T_L^K(s_i^\ell, p') - 2 > \mathbf{b}(s_i^\ell, p')(n - 2A)$. Note that for this action, $T_L^K(s_i^\ell, p') - 2 > \mathbf{b}_*(p'|s_i^\ell)(n - 2SA) \geqslant 0$, so we know that this specific action is taken at least once after the initialization phase and that it satisfies (C.15). Recall that we have defined $M(s_i^\ell) = \sum_a \pi(a|s_i^\ell)\sigma(s_i^\ell, a)$. Further define $M = \sum_{\ell=1}^L \sum_{s_i^\ell} M(s_i^\ell)$. Using the definition of the optimal allocation $T_L^{*,K}(s_i^\ell, p') = n_f \frac{\mathbf{b}_*(p'|s_i^\ell)}{M(s_i^\ell)}$, and the fact that $T_L^K(s_i^\ell, p') \geqslant \mathbf{b}_*(p'|s_i^\ell)(n_f - 2SA) + 2$, (C.15) may be written as for any state-action tuple (s_i^ℓ, p)

$$\begin{aligned}
&\frac{\mathbf{b}_*(p|s_i^\ell)}{T_L^K(s_i^\ell, p)} \\
&\leqslant \frac{\mathbf{b}_*(p'|s_i^\ell)}{T_L^{*,K}(p', s_i^\ell)} \frac{n_f}{(n_f - 2SA)} \left((2\eta + 4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_L^K(s_i^{\ell+1}, p') - 1}} + \mathbf{B}(s_i^{\ell+1}) \right) \\
&\leqslant \frac{M(s_i^\ell)}{n_f} + \frac{4SAM(s_i^\ell)}{n_f^2} + \frac{(2\eta + 4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\mathbf{b}_{*,\min}^{3/2}(s_i^\ell) n_f^{3/2}} \qquad \text{(C.16)}
\end{aligned}$$

because $n_f \geqslant 4SA$. By rearranging (C.16), we obtain the lower bound on $T_L^K(s_i^\ell, p)$:

$$\begin{aligned}
T_L^K(s_i^\ell, p) &\geqslant \frac{\mathbf{b}_*(p|s_i^\ell)}{\frac{M(s_i^\ell)}{n_f} + \frac{4SAM(s_i^\ell)}{n_f^2} + \frac{(2\eta + 4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\mathbf{b}_{*,\min}^{3/2}(s_i^\ell) n_f^{3/2}}} \\
&\overset{(a)}{\geqslant} T_L^{*,K}(s_i^\ell, p) - \frac{(2\eta + 4\eta^2)\mathbf{b}_*(p|s_i^\ell)\sqrt{\log(SAn(n+1)/\delta)}}{M(s_i^\ell)\mathbf{b}_{*,\min}^{3/2}(s_i^\ell) n_f^{3/2}} - 4A\mathbf{b}_*(p|s_i^\ell), \qquad \text{(C.17)}
\end{aligned}$$

where in (a) we use $1/(1+x) \geqslant 1 - x$ (for $x > -1$). Note that the lower bound holds on $\xi_{c,K}$ for any state-action (s_i^ℓ, p).

Step 5. (Upper bound on $T_L^K(s_i^\ell, p)$ for $\widehat{Z}^k \geqslant 0$): Now using (C.17) and the fact that n_f is given by $\sum_{\ell=1}^L \sum_{s_j^\ell} \sum_{a'=1}^A \mathbb{E}[T_L^K(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\} \cap$

$\mathbb{I}\{\xi_{v,K}\}] = n_f$, we obtain

$$T_L^K(s_i^\ell, p) = n_f - \sum_{p' \neq p} T_L^K(s_i^\ell, p') \leqslant \left(n_f - \sum_{p' \neq p} T_L^{*,K}(s_i^\ell, p') \right) + \sum_{p' \neq p} \left(\frac{(2\eta + 4\eta^2)\mathbf{b}_*(p'|s_i^\ell)\sqrt{\log(SAn(n+1)/\delta)}}{M(s_i^\ell)\mathbf{b}_{*,\min}^{3/2}(s_i^\ell)n_f^{3/2}} + 4A\mathbf{b}_*(p'|s_i^\ell) \right).$$

Now since $\sum_{p' \neq p} \mathbf{b}_*(p'|s_i^\ell) \leqslant 1$ we can show that

$$T_L^K(s_i^\ell, p) \leqslant T_L^{*,K}(s_i^\ell, p) + \frac{(2\eta + 4\eta^2)\mathbf{b}_*(p|s_i^\ell)\sqrt{\log(SAn(n+1)/\delta)}}{M(s_i^\ell)\mathbf{b}_{*,\min}^{3/2}(s_i^\ell)n_f^{3/2}} + 4A. \tag{C.18}$$

Step 6 (Bound part A): We now bound the part A using (C.16)

$$\sum_{a} \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}\cap\mathbb{I}\{\xi_{v,K}\}]$$
$$+\gamma^2\sum_{a}\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\sum_{a'}\pi^2(a'|s_j^\ell)\left[\frac{\sigma^2(s_j^\ell,a')}{\underline{T}_L^{(2),K}(s_j^\ell,a')}\right]\cdot$$
$$\mathbb{E}[T_L^K(s_j^\ell,a')\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]$$
$$\overset{(a)}{\leqslant}\left(\frac{M(s_1^1)}{n_f}+\frac{4SAM(s_1^1)}{n_f^2}+\frac{(2\eta+4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\mathbf{b}_{*,\min}^{3/2}(p|s_i^\ell)n_f^{3/2}}\right)^2 n_f$$
$$+\gamma^2\sum_{a}\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\cdot$$
$$\left(\frac{M(s_j^\ell)}{n_f}+\frac{4SAM(s_j^\ell)}{n_f^2}+\frac{(2\eta+4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\mathbf{b}_{*,\min}^{3/2}(p|s_i^\ell)n_f^{3/2}}\right)^2 n_f$$
$$=\frac{M^2(s_1^1)}{n_f}+\frac{8AM^2(s_1^1)}{n_f^2}+\frac{16A^2M^2(s_1^1)}{n_f^3}+O\left(\frac{(2\eta+4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\mathbf{b}_{*,\min}^{3/2}(p|s_i^\ell)n_f^{3/2}}\right)$$
$$+\gamma^2\sum_{a}\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\left(\frac{M^2(s_j^\ell)}{n_f}+\frac{8AM^2(s_j^\ell)}{n_f^2}+\frac{16A^2M^2(s_j^\ell)}{n_f^3}\right.$$
$$\left.+O\left(\frac{(2\eta+4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\mathbf{b}_{*,\min}^{3/2}(p|s_j^\ell)n_f^{3/2}}\right)\right)$$

where, in (a) follows from the definition of $M(s)$ and n_f.

Step 7 (Upper Bound to Constraint Violation): In this step we bound the quantity $\mathcal{C}_n(\pi)=\sum_{j=1}^{k}\mathbb{I}\{\widehat{Z}^j<0,\mathbf{b}^j\in\{\widehat{b}^k,\pi_0\}\}$. Define the number of times the policy $\mathbf{b}_*$ is played till episode k is $T^k(\mathbf{b}_*)$ and the number of times the baseline policy is played is given by $T^k(\pi_0)$. Observe that $\mathcal{C}_n(\pi)=\sum_{j=1}^{k}\mathbb{I}\{\widehat{Z}^j<0,\mathbf{b}^j\in\{\widehat{b}^k,\pi_0\}\}=T^K(\pi_0)\mathbb{I}\{\xi_{Z,K}^C\}$ as when the constraint are

violated policy π_0 is sampled. Let

$$\tau = \max \left\{ k \leqslant K \text{ and } n_f \geqslant \frac{\log(SAn(n+1)/\delta)}{\min_{s,a} \Delta^{c,\alpha,(2)}(s,a)} \mid \mathbf{b}^k = \pi_0 \right\}$$

be the last episode in which the baseline policy is played. We will define formally the gap $\Delta^{c,\alpha,(2)}(s,a)$ later. Observe that the constraint violation

can be re-stated as follows:

$$
\begin{aligned}
&\sum_{k=1}^{\tau} Y^{c}_{\mathbf{b}^k}(s_1^1) \coloneqq \sum_{k=1}^{\tau}\sum_{a} \mathbf{b}^k(a|s_1^1)\left(\widehat{\mu}^{c,k}_{L}(s_1,a)+\sum_{s_j^2} P(s_j^2|s_1^1,a)Y^{c}_{\mathbf{b}^k}(s_j^2)\right)\\
&\qquad\qquad < (1-\alpha)\tau V^{c}_{\pi_0}(s_1^1)\\
\implies &\sum_{k=1}^{\tau}\sum_{a} \mathbf{b}^k(a|s_1^1)\left(\underline{\widehat{\mu}}^{c,k}_{L}(s_1^1,a)+\sum_{s_j^2} P(s_j^2|s_1^1,a)\underline{Y}^{c}_{\mathbf{b}^k}(s_j^2)\right) < (1-\alpha)\tau V^{c}_{\pi_0}(s_1^1)\\
\overset{(a)}{\implies} &\sum_{k=1}^{\tau}\sum_{a} \mathbf{b}^k(a|s_1^1)\left(\underline{\widehat{\mu}}^{c,k}_{L}(s_1^1,a)+\sum_{s_j^2} P(s_j^2|s_1^1,a)\underline{Y}^{c}_{\mathbf{b}^k}(s_j^2)\right)\\
&\qquad < (1-\alpha)\sum_{k=1}^{\tau}\pi_0(0|s_1^1)\left(\mu^{c}(s_1^1,0)+\sum_{s_j^2} P(s_j^2|s_1^1,0)V^{c}_{\pi_0}(s_j^2)\right)\\
\implies &\sum_{k=1}^{\tau}\sum_{a} T^{k}_{L}(s_1^1,a)\left(\underline{\widehat{\mu}}^{c,k}_{L}(s_1^1,a)+\sum_{s_j^2} P(s_j^2|s_1^1,a)\underline{Y}^{c}_{\mathbf{b}^k}(s_j^2)\right)\\
&\qquad < (1-\alpha)\sum_{k=1}^{\tau}T^{k}_{L}(s_1^1,a)\left(\mu^{c}(s_1^1,0)+\sum_{s_j^2} P(s_j^2|s_1^1,0)V^{c}_{\pi_0}(s_j^2)\right)\\
\overset{(b)}{\implies} &\underbrace{\sum_{a} T^{\tau}_{L}(s_1^1,a)\underline{\widehat{\mu}}^{c,\tau}_{L}(s_1^1,a)}_{\textbf{Part A}}+\sum_{a} T^{\tau}_{L}(s_1^1,a)\sum_{s_j^2} P(s_j^2|s_1^1,a)\underline{Y}^{c}_{\mathbf{b}^k}(s_j^2)\\
&\qquad < \underbrace{(1-\alpha)\sum_{a} T^{\tau}_{L}(s_1^1,0)\mu^{c}(s_1^1,0)}_{\textbf{Part B}}+(1-\alpha)T^{\tau}_{L}(s_1^1,0)\sum_{s_j^2} P(s_j^2|s_1^1,0)V^{c}_{\pi_0}(s_j^2).
\end{aligned}
\tag{C.19}
$$

Comparing **Part A** and **Part B** for level $\ell = 1$ we observe that the constraint violation must satisfy

$$
\sum_{a} T^{\tau}_{L}(s_1^1,a)\underline{\widehat{\mu}}^{c,\tau}_{L}(s_1^1,a) < (1-\alpha)T^{\tau}_{L}(s_1^1,0)\mu^{c}(s_1^1,0)
$$

which can be reduced as follows

$$T_L^{\tau-1}(s_1^1, 0) \leqslant \frac{1}{\alpha\mu^c(s_1^1, 0)} \left(1 + \sum_{a=1}^{A} N(s_1^1, a)\right).$$

where $\Delta^{c,\alpha}(s_1^1, a) \coloneqq (1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a)$ and

$$\begin{aligned} N(s_1^1, a) &\coloneqq T_L^{\tau-1}(s_1^1, a) \cdot \big((1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) \\ &\qquad + c_1\sqrt{\log(An(n+1)/\delta)/T_L^{\tau-1}(s_1^1, a)}\Big) \\ &= \Delta^{c,\alpha}(s_1^1, a)T_L^{\tau-1}(s_1^1, a) + c_1\sqrt{\log(An(n+1)/\delta)T_L^{\tau-1}(s_1^1, a)} \end{aligned} \tag{C.20}$$

is a bound on the decrease in $\widehat{Z}_\tau$ in the first $\tau - 1$ rounds due to choosing action a in s_1^1. We will now bound $N(s_1^1, a)$ for each a. Now observe

$$\begin{aligned} &\Delta^{c,\alpha}(s_1^1, a) = (1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) \\ &= \mu^c(s_1^1, 0) - \alpha\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) \\ &= -(\mu^{*,c}(s_1^1) - \mu^c(s_1^1, 0)) - \alpha\mu^c(s_1^1, 0) + (\mu^{*,c}(s_1^1) - \mu^c(s_1^1, a)) \\ &= -\Delta^c(s_1^1, 0) - \alpha\mu^c(s_1^1, 0) + \Delta^c(s_1^1, a). \end{aligned}$$

where, $\mu^{*,c}(s_1^1) = \max_a \mu^c(s_1^1, a)$. Let $J(n_f) = \frac{(2\eta+4\eta^2)\mathbf{b}_*(p|s_i^\ell)\sqrt{\log(SAn(n+1)/\delta)}}{M(s_i^\ell)\mathbf{b}_{*,\min}^{3/2}(s_i^\ell)n_f^{3/2}}$. The first case is $\Delta^{c,\alpha}(s_1^1, a) > 0$, i.e. $\Delta^c(s_1^1, a) > \Delta^c(0) + \alpha\mu^c(0)$. These are the unsafe actions as $\Delta^{c,\alpha}(s_1^1, a) \coloneqq (1-\alpha)\mu^c(0) - \mu^c(s_1^1, a) > 0$ we have from (C.18)

$$T_n(s_1^1, a) \leqslant T_n^*(s_1^1, a) + J(n_f) + 4A = \frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f + J(n_f) + 4A$$

Plugging this back in $N(s_1^1, a)$ we get

$$\begin{aligned}
N(s_1^1, a) &= \Delta^{c,\alpha}(s_1^1, a) T_{\tau-1}(s_1^1, a) + c_1 \sqrt{\log(An(n+1)/\delta) T_{\tau-1}(s_1^1, a)} + J(n_f) \\
&\leqslant \frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f \Delta^{c,\alpha}(s_1^1, a) + 4A\Delta^{c,\alpha}(s_1^1, a) \\
&\quad + c_1 \sqrt{\log(An(n+1)/\delta) \left(\frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f + 4A \right)} + J(n_f) \\
&\overset{(a)}{\leqslant} \frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f \Delta^{c,\alpha}(s_1^1, a) + 4A\Delta^{c,\alpha}(s_1^1, a) \\
&\quad + c_1 \sqrt{\Delta^{c,\alpha,(2)}(s_1^1, a) \left(\frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f + 4A \right)} + J(n_f) \\
&\leqslant 2 \left(\frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f \Delta^{c,\alpha}(s_1^1, a) + 4A\Delta^{c,\alpha}(s_1^1, a) \right) + J(n_f). \qquad \text{(C.21)}
\end{aligned}$$

where, (a) follows for $n_f \geqslant \frac{\log(SAn(n+1)/\delta)}{\min_a \Delta^{c,\alpha,(2)}(s_1^1, a)}$. The other case is $\Delta^{c,\alpha}(s_1^1, a) < 0$, i.e. $\Delta^c(s_1^1, a) < \Delta^c(s_1^1, 0) + \alpha\mu^c(s_1^1, 0)$ then only safe actions are pulled. Then

$$\begin{aligned}
N(s_1^1, a) &\leqslant -\Delta^{c,\alpha}(s_1^1, a) T_{\tau-1}(s_1^1, a) + c_1 \sqrt{\log(An(n+1)/\delta) T_{\tau-1}(s_1^1, a)} + J(n_f) \\
&= \underbrace{-\Delta^{c,\alpha}(s_1^1, a)}_{a} T_{\tau-1}(s_1^1, a) + \underbrace{c_1 \sqrt{\log(An(n+1)/\delta)}}_{b} \sqrt{T_{\tau-1}(s_1^1, a)} + J(n_f) \\
&\overset{(a)}{\leqslant} -\frac{\log(An(n+1)/\delta)}{4\Delta^{c,\alpha}(s_1^1, a)} = \frac{\log(An(n+1)/\delta)}{4(\Delta^c(0) + \alpha\mu^c(0) - \Delta^c(s_1^1, a))} \\
&\overset{(b)}{\leqslant} 4 \left(\frac{\pi(s_1^1, a)\sigma(s_1^1, a)}{M} n_f (\Delta^c(0) + \alpha\mu^c(0) - \Delta^c(s_1^1, a)) \right) \qquad \text{(C.22)}
\end{aligned}$$

where, (a) follows by using $ax^2 + bx \leqslant -b^2/4a$ for $a < 0$, and (b) follows

as $n_f \geqslant \frac{\log(An(n+1)/\delta)}{\min^+_{a\in\mathcal{A}\setminus\{0\}} \pi(s_1^1,a)\sigma(s_1^1,a)\Delta^{c,\alpha,(2)}(s_1^1,a)}$ which implies

$$\frac{\log(An(n+1)/\delta)}{n_f} \leqslant 4\left(\sum_{\mathcal{A}\setminus\{0\}} \pi(s_1^1,a)\sigma(s_1^1,a)\overset{+}{\min}\{\Delta^c(s_1^1,a), \Delta^c(0)-\Delta^c(s_1^1,a)\}\right)^2$$
$$\implies \frac{\log(An(n+1)/\delta)}{\sum_{\mathcal{A}\setminus\{0\}} \pi(s_1^1,a)\sigma(s_1^1,a)\min^+\{\Delta^c(s_1^1,a), \Delta^c(0)-\Delta^c(s_1^1,a)\}}$$
$$\leqslant 4\left(\sum_{\mathcal{A}\setminus\{0\}} \pi(s_1^1,a)\sigma(s_1^1,a)\overset{+}{\min}\{\Delta^c(s_1^1,a), \Delta^c(0)-\Delta^c(s_1^1,a)\}\right) n_f.$$

Plugging everything back in (C.22), we get

$$\begin{aligned} n_u = T_{\tau-1}(s_1^1,0) &= \frac{1}{\alpha\mu^c(0)}\left(\sum_{a=1}^{A} N(s_1^1,a)\right) \\ &\leqslant \frac{2}{\alpha\mu^c(0)}\sum_{a\in\mathcal{A}_u}\Delta^c(s_1^1,a)\left(\frac{\pi(s_1^1,a)\sigma(s_1^1,a)}{M}n_f\right) \\ &\quad + \frac{4}{\alpha\mu^c(0)}\sum_{a\in\mathcal{A}_s\setminus\{0\}}\left(\Delta^c(0)-\Delta^c(s_1^1,a)\right)\left(\frac{\pi(s_1^1,a)\sigma(s_1^1,a)}{M}n_f\right) \\ &= \frac{6}{\alpha\mu^c(0)}\sum_{a\in\mathcal{A}\setminus\{0\}}\overset{+}{\min}\{\Delta^c(s_1^1,a), \Delta^c(0)-\Delta^c(s_1^1,a)\}\left(\frac{\pi(s_1^1,a)\sigma(s_1^1,a)}{M}(n-n_u)\right) \end{aligned} \tag{C.23}$$

It follows then that for the state s_1^1

$$n_u(s_1^1) \leqslant \frac{1}{\alpha\mu^c(s_1^1,0)}\left(1+\sum_{a=1}^{A} N(s_1^1,a)\right) \leqslant \frac{H_{*,(2)}(s_1^1)}{2}\frac{n}{M(s_1^1)}$$

where

$$H_{*,(2)}(s_i^\ell) \coloneqq \sum_a \mathbf{b}_*(a|s_i^\ell) \overset{+}{\min}\{\Delta^c(s_i^\ell, a), \Delta^c(s_i^\ell, 0) - \Delta^c(s_i^\ell, a)\},$$

$$M(s_i^\ell) \coloneqq \sum_a \sqrt{\pi^2(a|s_i^\ell)\left(\sigma^2(s_i^\ell, a) + \sum_{s_j^{\ell+1}} P(s_j^{\ell+1}|s_i^\ell, a) M^2(s_j^{\ell+1})\right)}. \quad \text{(C.24)}$$

For an arbitrary level $\ell \in [L]$, we can show using (C.19) that the constraint violation must satisfy

$$\begin{aligned}
&\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_a T_L^\tau(s_i^{\ell'}, a) \widehat{\underline{\mu}}_L^{c,\tau}(s_i^{\ell'}, a) < (1-\alpha) \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} T_L^\tau(s_i^{\ell'}, 0) \mu_0^c(s_i^{\ell'}, 0) \\
\overset{(a)}{\implies} &\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_a \Bigg(T_L^{*,K}(s_i^{\ell'}, a) - 4A\mathbf{b}_*(a|s_i^{\ell'}) \\
&\qquad - O\left(\frac{(2\eta + 4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\min_{s_i^{\ell'}} \mathbf{b}_{*,\min}^{k,(3/2)}(s_i^{\ell'}) n_f^{3/2}}\right)\Bigg) \widehat{\underline{\mu}}_L^{c,\tau}(s_i^{\ell'}, a) \\
&\qquad < (1-\alpha) \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \Bigg(T_L^{*,K}(s_i^{\ell'}, 0) + 4A \\
&\qquad\quad + O\left(\frac{(2\eta + 4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\min_{s_i^{\ell'}} \mathbf{b}_{*,\min}^{k,(3/2)}(s_i^{\ell'}) n_f^{3/2}}\right)\Bigg) \mu^c(s_i^{\ell'}, 0) \\
\implies &\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_a \left(T_L^{*,K}(s_i^{\ell'}, a)\right) \widehat{\underline{\mu}}_L^{c,\tau}(s_i^{\ell'}, a) < (1-\alpha) \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \left(T_L^{*,K}(s_i^{\ell'}, 0)\right) \mu^c(s_i^{\ell'}, 0) \\
&\quad + 8LSA^2(\mu^c(s_i^{\ell'}, 0) + \widehat{\underline{\mu}}_L^{c,\tau}(s_i^{\ell'}, a)) \\
&\quad + O\left(\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \frac{(2\eta + 4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\min_{s_i^{\ell'}} \mathbf{b}_{*,\min}^{k,(3/2)}(s_i^{\ell'}) n_f^{3/2}}\right)
\end{aligned}$$

$$\implies \sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\sum_{a}\left(T_L^{*,K}(s_i^{\ell'},a)\right)\underline{\hat{\mu}}_L^{c,\tau}(s_i^{\ell'},a) < (1-\alpha)\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\left(T_L^{*,K}(s_i^{\ell'},0)\right)\mu^c(s_i^{\ell'},0)$$
$$+ 8LSA^2(\mu_{0,L}^{c,\tau}(s_i^{\ell'},a) + \hat{\mu}_L^{c,\tau}(s_i^{\ell'},a) - \sqrt{\frac{\log((SAn(n+1)/\delta)}{2T_L^{\tau}(s_i^{\ell'},a)}})$$
$$+ O\left(\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\frac{(2\eta+4\eta^2)\sqrt{\log(SAn(n+1)/\delta)}}{\min_{s_i^{\ell'}}\mathbf{b}_{*,\min}^{k,(3/2)}(s_i^{\ell'})n_f^{3/2}}\right)$$
$$\implies \sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\sum_{a}\left(T_L^{*,K}(s_i^{\ell'},a)\right)\underline{\hat{\mu}}_L^{c,\tau}(s_i^{\ell'},a) < (1-\alpha)\max_s\mu^c(s,0)\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\left(T_L^{*,K}(s_i^{\ell'},0)\right)$$
$$+ 16LSA^2 + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s\mathbf{b}_{*,\min}^{k,(3/2)}(s)n_f^{3/2}}\right) \tag{C.25}$$

where, (a) follows as $\mu(s,a)\in(0,1]$ for all s,a and using (C.17) and (C.18). Summing over all states s_j^{ℓ} till level L we can show that

$$n_u = \sum_{\ell=1}^{L}\sum_{s_j^{\ell}}T_L^{*,K}(s_j^{\ell},0)$$
$$\leqslant \frac{n}{2}\sum_{\ell=1}^{L}\sum_{s_j^{\ell}}\frac{H_{*,(2)}(s_j^{\ell})}{M(s_j^{\ell})} + 16LSA^2 + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s\mathbf{b}_{*,\min}^{k,(3/2)}(s)n_f^{3/2}}\right)n_f$$
$$\overset{(a)}{\leqslant} \frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + 16LSA^2 + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s\mathbf{b}_{*,\min}^{k,(3/2)}(s)n_f^{1/2}}\right)$$
$$\overset{(b)}{\leqslant} \frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + 16LSA^2 + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s\mathbf{b}_{*,\min}^{k,(3/2)}(s)n^{1/2}}\right) \tag{C.26}$$

where, in (a) we define $M_{\min} = \min_s M(s)$, and $H_{*,(2)} = \sum_{\ell=1}^{L}\sum_{s_j^{\ell}}H_{*,(2)}(s_j^{\ell})$, and (b) follows by setting $n_f = n - n_u$. Finally, observe that $16LSA^2$ does not depend on the episode K, and the quantity $O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s\mathbf{b}_{*,\min}^{k,(3/2)}(s)n^{1/2}}\right)$

decreases with n.

Step 8 (Lower Bound to Constraint Violation): For the lower bound to the constraint we equate Equation (C.19) to 0 and show that

$$\underbrace{\sum_{a} T_L^{\tau}(s_1^1, a)\underline{\widehat{\mu}}_L^{c,\tau}(s_1^1, a)}_{\textbf{Part A}} + \sum_{a} T_L^{\tau}(s_1^1, a) \sum_{s_j^2} P(s_j^2|s_1^1, a)\underline{Y}_{\mathbf{b}^k}^{c}(s_j^2)$$
$$= \underbrace{(1-\alpha)\sum_{a} T_L^{\tau}(s_1^1, 0)\mu^c(s_1^1, 0)}_{\textbf{Part B}} + (1-\alpha)T_L^{\tau}(s_1^1, 0) \sum_{s_j^2} P(s_j^2|s_1^1, 0)V_{\pi_0}^{c}(s_j^2).$$

Again comparing **Part A** and **Part B** for level $\ell = 1$ we observe that the lower bound to constraint violation must satisfy

$$\sum_{a} T_L^{\tau}(s_1^1, a)\underline{\widehat{\mu}}_L^{c,\tau}(s_1^1, a) = (1-\alpha)T_L^{\tau}(s_1^1, 0)\mu^{c,}(s_1^1, 0)$$

which can be reduced as

$$\sum_{a} T_L^{\tau-1}(s_1^1, 0) \geqslant \frac{1}{\alpha\mu^c(s_1^1, 0)}\left(1 + \sum_{a=1}^{A} \underline{N}(s_1^1, a)\right).$$

where $\Delta^{c,\alpha}(s_1^1, a) \coloneqq (1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a)$ and

$$\begin{aligned}
&\underline{N}(s_1^1, a) \\
&\coloneqq T_L^{\tau-1}(s_1^1, a) \cdot \left((1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) + c_1\sqrt{\log(An(n+1)/\delta)/T_L^{\tau-1}(s_1^1, a)}\right) \\
&= \Delta^{c,\alpha}(s_1^1, a)T_L^{\tau-1}(s_1^1, a) + c_1\sqrt{\log(An(n+1)/\delta)T_L^{\tau-1}(s_1^1, a)} \\
&\overset{(a)}{\geqslant} \Delta^{c,\alpha}(s_1^1, a)\left(T_L^{*,K}(s_1^1, a) - 4A\mathbf{b}_*(a|s_1^1)\right) \\
&\quad + c_1\sqrt{\log(An(n+1)/\delta)\left(T_L^{*,K}(s_1^1, a) - 4A\mathbf{b}_*(a|s_1^1)\right)}
\end{aligned}$$

where, (a) follows from (C.17). Then we can show that

$$\begin{aligned}
T_L^{\tau-1}(s_1^1,0) &\geqslant \frac{1}{\alpha\mu^c(s_1^1,0)}\left(1+\sum_{a=1}^{A}\underline{N}(s_1^1,a)\right)\\
&\geqslant \frac{n_f}{M(s_1^1)}\left(\frac{H_{*,(2)}(s_1^1)}{8}-\frac{A}{2}\frac{H_{*,(2)}(s_1^1)}{M(s_1^1)}\right)-16SA
\end{aligned}$$

Similarly for any arbitrary level $\ell \in [L]$ following the same way as step 7 above it can be shown that

$$\begin{aligned}
&\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\sum_{a}\left(T_L^{*,K}(s_i^{\ell'},a)+4A\right)\underline{\widehat{\mu}}_L^{c,\tau}(s_i^{\ell'},a)\\
&\geqslant (1-\alpha)\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\left(T_L^{*,K}(s_i^{\ell'},0)-4A\mathbf{b}_*(0|s_i^{\ell'})\right)\mu^c(s_i^{\ell'},0)\\
\implies &\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\sum_{a}T_L^{*,K}(s_j^{\ell},a)\underline{\widehat{\mu}}_L^{c,\tau}(s_j^{\ell},a)\geqslant(1-\alpha)\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}T_L^{*,K}(s_i^{\ell},0)\mu^c(s_i^{\ell},0)-16LSA^2.
\end{aligned}$$

For the state s_j^{ℓ} we can show that

$$\begin{aligned}
\sum_{\ell'=1}^{\ell}\sum_{s_j^{\ell'}}T_L^{*,K}(s_j^{\ell'},0) &\geqslant \frac{1}{\alpha\max_s\mu^c(s_j^{\ell},0)}\left(1+\sum_{\ell'=1}^{\ell}\sum_{s_j^{\ell'}}\sum_{a=1}^{A}\underline{N}(s_j^{\ell'},a)\right)\\
&\geqslant \sum_{\ell=1}^{L}\sum_{s_j^{\ell}}\frac{n_f}{M(s_j^{\ell})}\left(\frac{H_{*,(2)}(s_j^{\ell})}{8}-\frac{A}{2}\frac{H_{*,(2)}(s_j^{\ell})}{M(s_j^{\ell})}\right)-16LSA^2\\
&\quad-O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s\mathbf{b}_{*,\min}^{k,(3/2)}(s)n_f^{3/2}}\right).
\end{aligned}$$

Finally summing over all states s_j^ℓ and level L we can show that

$$\sum_{\ell=1}^{L}\sum_{s_j^\ell} T_L^{*,K}(s_j^\ell,0) \geqslant \sum_{\ell=1}^{L}\sum_{s_j^\ell}\frac{n_f}{M(s_j^\ell)}\left(\frac{H_{*,(2)}(s_j^\ell)}{8}-\frac{A}{2}\frac{H_{*,(2)}(s_j^\ell)}{M(s_j^\ell)}\right) - 16LSA^2 - O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n_f^{3/2}}\right). \tag{C.27}$$

Again, observe that $16LSA^2$ does not depend on the episode K.

Step 9 (Bound Part B): Then from (C.27) we can show that

$$\begin{aligned}
&\frac{M(s_1^1)}{\sum_{\ell=1}^{L}\sum_{s_j^\ell} T_L^{*,K}(s_j^\ell,0)} \\
&\leqslant \frac{M(s_1^1)}{\sum_{\ell=1}^{L}\sum_{s_j^\ell}\frac{n_f}{M(s_j^\ell)}\left(\frac{H_{*,(2)}(s_j^\ell)}{8}-\frac{A}{2}\frac{H_{*,(2)}(s_j^\ell)}{M(s_j^\ell)}\right) - 16LSA^2 - O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n_f^{3/2}}\right)} \\
&\overset{(a)}{\leqslant} (M(s_1^1)+16LSA^2)\sum_{\ell=1}^{L}\sum_{s_j^\ell}\frac{M(s_j^\ell)}{n_f}\left(\frac{H_{*,(2)}(s_j^\ell)}{8}+\frac{A}{2}\frac{H_{*,(2)}(s_j^\ell)}{M(s_j^\ell)}\right) \\
&\quad + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n_f^{3/2}}\right) \\
&\leqslant (M(s_1^1)+16LSA^2)\sum_{\ell=1}^{L}\sum_{s_j^\ell}\frac{M(s_j^\ell)}{n_f}\left(2+H_{*,(2)}(s_j^\ell)\right) \\
&\quad + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n_f^{3/2}}\right) \\
&\overset{(b)}{\leqslant} (M(s_1^1)+16LSA^2)\frac{M}{n_f}\left(2+H_{*,(2)}\right) + O\left(\frac{(2\eta+4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n_f^{3/2}}\right)
\end{aligned}$$

where, (a) follows for $1/(x-c) \leqslant x+c$ for $x^2 \geqslant 1+c^2$ and $c>0$. The (b) follows for $H_{*,(2)} = \sum_{\ell=1}^{L}\sum_{s_j^\ell} H_{*,(2)}(s_j^\ell)$ and $M = \sum_{\ell=1}^{L}\sum_{s_j^\ell} M(s_j^\ell)$. It

follows then by setting $n_f = n - n_u$ that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{Z,K}^C\}\right] \\
&\overset{(a)}{\leqslant} \left(\frac{(M(s_1^1) + 16LSA^2)M\left(2 + H_{*,(2)}\right)}{n_f}\right. \\
&\qquad \left. + O\left(\frac{(2\eta + 4\eta^2)L\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n_f^{3/2}}\right)\right)^2 n_u \\
&\overset{(b)}{\leqslant} \frac{(M(s_1^1) + 16LSA^2)^2 n_u}{(n-n_u)^2}\left(2 + H_{*,(2)}\right)^2 \\
&\qquad + O\left(\frac{(2\eta + 4\eta^2)L^2S^2A^4H_{*,(2)}^2M^2\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s)(n-n_u)^{3/2}}\right) \\
&\overset{(c)}{\leqslant} \frac{(M(s_1^1) + 16LSA^2)^2 H_{*,(2)} n}{(n - H_{*,(2)}n)^2}\left(2 + H_{*,(2)}\right)^2 \\
&\qquad + O\left(\frac{(2\eta + 4\eta^2)L^2S^2A^4H_{*,(2)}^2M^2\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s)(n - H_{*,(2)}n)^{3/2}}\right) \\
&\leqslant \frac{M^2(s_1^1)}{n}\left(32MLSA^2 + H_{*,(2)}\right)^2 \\
&\qquad + O\left(\frac{(2\eta + 4\eta^2)L^2S^2A^4H_{*,(2)}^2M^2\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s) n^{3/2}}\right)
\end{aligned}$$

where, (a) follows from Theorem C.1, (b) follows from definition of $H_{*,(2)}$, and (c) follows from (C.26).

Step 10 (Combine everything): Combining everything from step 5,

step 8 and setting $\delta = 1/n^2$ we can show that the MSE of SaVeRscales as

$$\begin{aligned}\mathcal{L}_n(\pi, \widehat{b}^k) \leqslant & \frac{M^2(s_1^1)}{n} + \frac{8AM^2(s_1^1)}{n^2} + \frac{16A^2M^2(s_1^1)}{n^3} + \frac{M^2(s_1^1)}{n}\left(32MLSA + H_{*,(2)}\right)^2 \\ & + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{c,K}^C\}\right]}_{\textbf{Part C, Safety event does not hold}} + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{v,K}^C\}\right]}_{\textbf{Part D, Variance event does not hold}} \\ \overset{(a)}{\leqslant} & \frac{M^2(s_1^1)}{n} + \frac{8AM^2(s_1^1)}{n^2} + \frac{16A^2M^2(s_1^1)}{n^3} \\ & + \frac{M^2(s_1^1)}{n}\left(32MLSA + H_{*,(2)}\right)^2 + 2\sum_{t=1}^{n}\frac{2\eta + 4\eta^2}{n^2} \\ & + O\left(\frac{(2\eta + 4\eta^2)L^2S^2A^4H_{*,(2)}^2M^2\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}_{*,\min}^{k,(3/2)}(s)n^{3/2}}\right)\end{aligned} \tag{C.28}$$

where, (a) follows as $\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{c,K}^C\}\right] \leqslant 2\eta + 4\eta^2$ and using the low error probability of the cost event from Theorem C.14 and variance event from Corollary 3. The claim of the theorem follows.

□

C.4 Proof of Tree Oracle MSE

Proposition 2. (***formal***) *Let Assumption 6 hold. Then the MSE of the oracle for* $\frac{n}{\log(SAn(n+1)/\delta)} \geqslant 32(LSA^2)^2 + \frac{SA}{\min_{s,a}\Delta^{c,(2)}(s,a)} + \frac{1}{4H_{*,(2)}^2}$ *is bounded by*

$$\begin{aligned}\mathcal{L}_n(\pi, \mathbf{b}_*^k) \leqslant & \frac{M^2(s_1^1)}{n} + \frac{8AM^2(s_1^1)}{n^2} + \frac{16A^2M^2(s_1^1)}{n^3} \\ & + \frac{M^2(s_1^1)}{n}\left(32MLSA^2 + H_{*,(2)}\right)^2 + 2\sum_{t=1}^{n}\frac{2\eta + 4\eta^2}{n^2} + \frac{2}{n}\end{aligned}$$

with probability $(1-\delta)$. *The* $M = \sum_{\ell=1}^{L}\sum_{s_j^\ell} M(s_j^\ell)$, *and* $H_{*,(2)} = \sum_{\ell=1}^{L}\sum_{s_j^\ell} H_{*,(2)}(s_j^\ell)$ *is the problem complexity parameter. The total predicted constraint violations is*

bounded by

$$\mathcal{C}_n^*(\pi, \mathbf{b}_*^k) \leqslant \frac{H_{*,(2)}}{2} \frac{n}{M_{\min}} + 16LSA^2$$

with probability $(1-\delta)$*, where* $M_{\min} \coloneqq \min_s M(s)$.

Proof. **Step 1 (Sampling rule):** We follow the proof technique of Theorem 2. Note that the oracle tree algorithm knows the variances of reward and constraints values (but does not know the mean of either) and samples by the following rule

$$\mathbf{b}_*^k = \begin{cases} \pi_x, & \text{if } \widehat{Z}_L^{k-1} \geqslant 0, k \leqslant \sqrt{K} \\ \mathbf{b}_*, & \text{if } \widehat{Z}_L^{k-1} \geqslant 0, k > \sqrt{K} \\ \pi_0 & \text{if } \widehat{Z}_L^{k-1} < 0 \end{cases}. \tag{C.29}$$

where, $\widehat{Z}_L^{k-1} \coloneqq \sum_{k'=1}^{k-1} (Y_{c,L}^{\mathbf{b}^{k'}}(s_1^1) - \beta_L^{k'}(s,a)) - (1-\alpha)(k-1)V_c^{\pi_0}(s_1^1)$ is the safety budget till the k-th episode.

Step 2 (MSE Decomposition): Now recall that the oracle knows the variances but does not know the means (constraint and reward). We define the good constraint event when the oracle has a good estimate of the constraint mean. This is stated as follows:

$$\xi_{c,K} \coloneqq \bigcap_{\substack{1 \leqslant k \leqslant K, \\ 1 \leqslant a \leqslant A, 1 \leqslant s \leqslant S}} \left\{ \left| \widehat{\mu}_L^{c,k}(s,a) - \mu^c(s,a) \right| \leqslant (2\eta + 4\eta^2) \sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_L^k(s,a)}} \right\} \tag{C.30}$$

where, $n = KL$ and K is the number of episodes and L is the length of horizon of each episode. Define $c_1 = 2\eta + 4\eta^2$.

The exploration policy π_e results in a good constraint estimate of state-action tuples. This is shown in Corollary 4.

We also define the safety budget event $\xi_{Z,K} \coloneqq \bigcap_{1 \leqslant k \leqslant K} \left\{ \widehat{Z}^k \geqslant 0 \right\}$. Now

using Theorem C.1 we can show that

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-V_\pi(s_1^1)\right)^2\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}\right]\leqslant\sum_a\pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\cdot\\
&\quad\mathbb{E}[T_L^K(s_1^1,a)\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]\\
&\quad+\gamma^2\sum_a\pi^2(a|s_1^1)\sum_{s_j^2}P(s_j^2|s_1^1,a)\mathrm{Var}[Y_n(s_j^2)]\mathbb{E}[T_L^K(s_j^2,a)\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]\\
&\leqslant\sum_a\pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]\\
&\quad+\gamma^2\sum_a\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\sum_{a'}\pi^2(a'|s_j^\ell)\left[\frac{\sigma^2(s_j^\ell,a')}{\underline{T}_L^{(2),K}(s_j^\ell,a')}\right]\cdot\\
&\quad\mathbb{E}[T_L^K(s_j^\ell,a')\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]
\end{aligned}$$

which implies that the oracle does not need to know the reward means $\mu(a)$. Hence, Using the definition of MSE we can show that the MSE of oracle is bounded by

$$\begin{aligned}
\mathcal{L}_n(\pi)\leqslant&\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-V_\pi(s_1^1)\right)^2\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}\right]}_{\textbf{Part A, }\widehat{Z}_n\geqslant 0\textbf{, safety event holds}}+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-V_\pi(s_1^1)\right)^2\mathbb{I}\{\xi_{Z,K}^C\}\right]}_{\textbf{Part B, }\widehat{Z}_n<0\textbf{, constraint violation}}\\
&+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-V_\pi(s_1^1)\right)^2\mathbb{I}\{\xi_{c,K}^C\}\right]}_{\textbf{Part C, Safety event does not hold}}\\
&\leqslant\sum_a\pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]\\
&\quad+\gamma^2\sum_a\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\sum_{a'}\pi^2(a'|s_j^\ell)\left[\frac{\sigma^2(s_j^\ell,a')}{\underline{T}_L^{(2),K}(s_j^\ell,a')}\right]\cdot\\
&\quad\mathbb{E}[T_L^K(s_j^\ell,a')\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}]\\
&\quad+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-V_\pi(s_1^1)\right)^2\mathbb{I}\{\xi_{Z,K}^C\}\right]}_{\textbf{Part B, }\widehat{Z}_n<0\textbf{, constraint violation}}+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1)-V_\pi(s_1^1)\right)^2\mathbb{I}\{\xi_{c,K}^C\}\right]}_{\textbf{Part C, Safety event does not hold}}
\end{aligned}$$

Divide the total budget n into two parts, n_f when $\sum_{j=1}^{k} \mathbb{I}\{\widehat{Z}^j \geqslant 0\}$ is true, then $\mathbf{b}_*$ is run. Hence define

$$n_f \coloneqq \sum_{k=1}^{K} \sum_{\ell=1}^{L} \sum_{s_j^\ell} \sum_{a'=1}^{A} \mathbb{E}[T_\ell^k(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\}].$$

The other part consist of $n_u = n - n_f$ number of samples when $\sum_{j=1}^{k} \mathbb{I}\{\widehat{Z}^k < 0\}$ and only π_0 is run. Hence we define,

$$n_u = \sum_{k=1}^{K} \sum_{\ell=1}^{L} \sum_{s_j^\ell} \sum_{a'=1}^{A} \mathbb{E}[T_\ell^k(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}^C\}].$$

Step 3 (Sampling of oracle for an episode k when $\widehat{Z}^k \geqslant 0$): First note that when $\widehat{Z}^k \geqslant 0$ the oracle samples at episode k according to the policy $\mathbf{b}_*$. The following the same steps as in step 3 of Theorem 2 we can show that. At episode k, time $\ell + 1$, the $\mathbf{b}_*$ samples the state-action tuple, action $\arg\max_a U_{\ell+1}^k(s_i^{\ell+1}, a)$ where

$$U_\ell^k(s_i^\ell, a) \coloneqq \frac{\mathbf{b}_{*,\ell}(a|s_i^\ell)}{T_\ell^k(s_i^\ell, a)} \tag{C.31}$$

Let $\ell + 1 > 2SA$ be the time at which a given state-action (s_i^ℓ, p') is visited for the last time, i.e., $T_\ell^k(p') = T_L^K(p') - 1$ and $T_{\ell+1}^k(p') = T_L^K(p')$. Note that as $n = KL \geqslant 4SA$, there is at least one state-action pair (s_i^ℓ, p') such that this happens, i.e. such that it is visited after the initialization phase. Since the oracle chooses to pull visit (s_i^ℓ, p') at time $\ell + 1$, we have for any state-action pair (s_i^ℓ, p')

$$U_{\ell+1}^k(s_i^{\ell+1}, p) \leqslant U_{\ell+1}^k(s_i^{\ell+1}, p'). \tag{C.32}$$

From (C.31) and using the fact that $T_\ell^k(s_i^\ell, p') = T_L^K(s_i^\ell, p') - 1$, we can

show that

$$U^k_{\ell+1}(s_i^{\ell+1}, p') \leqslant \frac{\mathbf{b}_*(p'|s_i^{\ell+1})}{T^k_t(s_i^{\ell+1}, p')} = \frac{\mathbf{b}_*(p'|s_i^{\ell+1})}{T^K_L(s_i^{\ell+1}, p') - 1} \tag{C.33}$$

Also note that

$$U^k_{\ell+1}(s_i^{\ell+1}, p) = \frac{\mathbf{b}_*(p|s_i^{\ell+1})}{T^k_t(s_i^{\ell+1}, p)} \overset{(a)}{\geqslant} \frac{\mathbf{b}_*(p|s_i^{\ell+1})}{T^K_L(s_i^{\ell+1}, p)}. \tag{C.34}$$

where, (a) follows as $T_t(p) \leqslant T^K_L(p, s_i^{\ell+1})$ (i.e., the number of times p has been sampled can only increase after time ℓ). Combining (C.32), (C.33), (C.34) we can show that for any action p:

$$\frac{\mathbf{b}_*(p|s_i^{\ell+1})}{T^K_L(p, s_i^{\ell+1})} \leqslant \frac{\mathbf{b}_*(p'|s_i^{\ell+1})}{T^K_L(p', s_i^{\ell+1}) - 1} \tag{C.35}$$

Note that in the above equation, there is no dependency on ℓ, and thus, the probability that (C.35) holds for any $(s_i^{\ell+1}, p)$ and for any $(s_i^{\ell+1}, p')$ such that state-action $(s_i^{\ell+1}, p')$ is visited after the initialization phase, i.e., such that $T^K_L(s_i^{\ell+1}, p') > 2$ depends on the probability of event $\xi_{Z,n}$.

Step 4. (Lower bound on $T^K_L(s_i^\ell, p)$ for $\widehat{Z}^k \geqslant 0$): If a state-action tuple s_i^ℓ, p, p is under-pulled compared to its optimal allocation without taking into account the initialization phase, i.e., $T^K_L(s_i^\ell, p) - 2 < \mathbf{b}(p|s_i^\ell)(n - 2A)$, then from the constraint $\sum_{p'} \left(T^K_L(s, p') - 2\right) = n - 2SA$ and the definition of the optimal allocation, we deduce that there exists at least another state-action tuple s_i^ℓ, p' that is over-visited compared to its optimal allocation without taking into account the initialization phase, i.e., $T^K_L(s_i^\ell, p') - 2 > \mathbf{b}(s_i^\ell, p')(n - 2SA)$. Note that for this action, $T^K_L(s_i^\ell, p') - 2 > \mathbf{b}_*(p'|s_i^\ell)(n - 2SA) \geqslant 0$, so we know that this specific action is pulled at least once after the initialization phase and that it satisfies (C.35). Recall that we have defined $M(s_i^\ell) = \sum_a \pi(a|s_i^\ell)\sigma(s_i^\ell, a)$. Further define $M = \sum_{\ell=1}^L \sum_{s_i^\ell} M(s_i^\ell)$. Using the definition of the optimal allocation $T^{*,K}_L(s_i^\ell, p') = n_f \frac{\mathbf{b}_*(p'|s_i^\ell)}{M(s_i^\ell)}$, and

the fact that $T_L^K(s_i^\ell, p') \geqslant \mathbf{b}_*(p'|s_i^\ell)(n_f - 2SA) + 2$, (C.35) may be written as for any state-action tuple (s_i^ℓ, p)

$$\frac{\mathbf{b}_*(p|s_i^\ell)}{T_L^K(s_i^\ell, p)} \leqslant \frac{\mathbf{b}_*(p'|s_i^\ell)}{T_L^{*,K}(p', s_i^\ell)} \frac{n_f}{(n_f - 2SA)} \leqslant \frac{M(s_i^\ell)}{n_f} + \frac{4AM(s_i^\ell)}{n_f^2} \tag{C.36}$$

because $n_f \geqslant 4SA$. By rearranging (C.36), we obtain the lower bound on $T_L^K(s_i^\ell, p)$:

$$T_L^K(s_i^\ell, p) \geqslant \frac{\mathbf{b}_*(p|s_i^\ell)}{\frac{M(s_i^\ell)}{n_f} + \frac{4AM(s_i^\ell)}{n_f^2}} = \frac{\mathbf{b}_*(p|s_i^\ell)}{\frac{M(s_i^\ell)}{n_f}\left(1 + \frac{4A}{n_f}\right)} \overset{(a)}{\geqslant} T_L^{*,K}(s_i^\ell, p) - 4A\mathbf{b}_*(p|s_i^\ell), \tag{C.37}$$

where in (a) we use $1/(1+x) \geqslant 1 - x$ (for $x > -1$). Note that the lower bound holds on $\xi_{c,K}$ for any action p.

Step 5. (Upper bound on $T_L^K(s_i^\ell, p)$ for $\widehat{Z}^k \geqslant 0$): Now using (C.37) and the fact that n_f is given by $\sum_{\ell=1}^{L} \sum_{s_j^\ell} \sum_{a'=1}^{A} \mathbb{E}[T_L^K(s_j^\ell, a')\mathbb{I}\{\xi_{Z,K}\} \cap \mathbb{I}\{\xi_{c,K}\}] = n_f$, we obtain

$$T_L^K(s_i^\ell, p) = n_f - \sum_{p' \neq p} T_L^K(s_i^\ell, p') \leqslant \left(n_f - \sum_{p' \neq p} T_L^{*,K}(s_i^\ell, p')\right) + \sum_{p' \neq p} 4A\mathbf{b}_*(p'|s_i^\ell).$$

Now since $\sum_{p' \neq p} \mathbf{b}_*(p'|s_i^\ell) \leqslant 1$ we can show that

$$T_L^K(s_i^\ell, p) \leqslant T_L^{*,K}(s_i^\ell, p) + 4A. \tag{C.38}$$

Step 6 (Bound part A): We now bound the part A using (C.36)

$$
\begin{aligned}
&\sum_{a} \pi^2(a|s_1^1)\left[\frac{\sigma^2(s_1^1,a)}{\underline{T}_L^{(2),K}(s_1^1,a)}\right]\mathbb{E}[T_L^K(s_1^1,a)\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}] \\
&\quad+\gamma^2\sum_{a}\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\sum_{a'}\pi^2(a'|s_j^\ell)\left[\frac{\sigma^2(s_j^\ell,a')}{\underline{T}_L^{(2),K}(s_j^\ell,a')}\right]\cdot \\
&\qquad\mathbb{E}[T_L^K(s_j^\ell,a')\mathbb{I}\{\xi_{Z,K}\}\cap\mathbb{I}\{\xi_{c,K}\}] \\
&\overset{(a)}{\leqslant}\left(\frac{M(s_1^1)}{n_f}+\frac{4AM(s_1^1)}{n_f^2}\right)^2 n_f \\
&\quad+\gamma^2\sum_{a}\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\left(\frac{M(s_j^\ell)}{n_f}+\frac{4AM(s_j^\ell)}{n_f^2}\right)^2 n_f \\
&=\frac{M^2(s_1^1)}{n_f}+\frac{8AM^2(s_1^1)}{n_f^2}+\frac{16A^2M^2(s_1^1)}{n_f^3} \\
&\quad+\gamma^2\sum_{a}\pi^2(a|s_1^1)\sum_{\ell=2}^{L}\sum_{s_j^\ell}P(s_j^\ell|s_1^1,a)\left(\frac{M^2(s_j^\ell)}{n_f}+\frac{8AM^2(s_j^\ell)}{n_f^2}+\frac{16A^2M^2(s_j^\ell)}{n_f^3}\right)
\end{aligned}
$$

where, in (a) follows from the definition of $M(s)$ and n_f.

Step 7 (Upper bound to Constraint violation): In this step we bound the quantity $\mathcal{C}_n^*(\pi)=\sum_{j=1}^{k}\mathbb{I}\{\widehat{Z}^j<0,\mathbf{b}^j\in\{\mathbf{b}_*,\pi_0\}\}$. Define the number of times the policy $\mathbf{b}_*$ is played till episode k is $T^k(\mathbf{b}_*)$ and the number of times the baseline policy is played is given by $T^k(\pi_0)$. Observe that $\mathcal{C}_n^*(\pi)=\sum_{j=1}^{k}\mathbb{I}\{\widehat{Z}^j<0,\mathbf{b}^j\in\{\mathbf{b}_*,\pi_0\}\}=T^K(\pi_0)\mathbb{I}\{\xi_{Z,K}^C\}$ as when the constraint are violated and policy π_0 is played. Let

$$
\tau=\max\left\{k\leqslant K \text{ and } n_f\geqslant\frac{\log(SAn(n+1)/\delta)}{\min_{s,a}\mathbf{b}_*(a|s)\Delta^{c,\alpha,(2)}(s,a)}\mid\mathbf{b}^k=\pi_0\right\}
$$

be the last episode in which the baseline policy is played. We will define formally the gap $\Delta^{c,\alpha,(2)}(s,a)$ later. Observe that the constraint violation

can be re-stated as follows:

$$
\begin{aligned}
&\sum_{k=1}^{\tau} Y^{c}_{\mathbf{b}^k}(s_1^1) \\
&\coloneqq \sum_{k=1}^{\tau} \sum_{a} \mathbf{b}^k(a|s_1^1) \left(\widehat{\mu}_L^{c,k}(s_1,a) + \sum_{s_j^2} P(s_j^2|s_1^1,a) Y^{c}_{\mathbf{b}^k}(s_j^2) \right) < (1-\alpha)\tau V^{c}_{\pi_0}(s_1^1) \\
&\implies \sum_{k=1}^{\tau} \sum_{a} \mathbf{b}^k(a|s_1^1) \left(\underline{\widehat{\mu}}_L^{c,k}(s_1^1,a) + \sum_{s_j^2} P(s_j^2|s_1^1,a) \underline{Y}^{c}_{\mathbf{b}^k}(s_j^2) \right) < (1-\alpha)\tau V^{c}_{\pi_0}(s_1^1) \\
&\overset{(a)}{\implies} \sum_{k=1}^{\tau} \sum_{a} \mathbf{b}^k(a|s_1^1) \left(\underline{\widehat{\mu}}_L^{c,k}(s_1^1,a) + \sum_{s_j^2} P(s_j^2|s_1^1,a) \underline{Y}^{c}_{\mathbf{b}^k}(s_j^2) \right) \\
&\qquad < (1-\alpha) \sum_{k=1}^{\tau} \pi_0(0|s_1^1) \left(\mu^{c}(s_1^1,0) + \sum_{s_j^2} P(s_j^2|s_1^1,0) V^{c}_{\pi_0}(s_j^2) \right) \\
&\implies \sum_{k=1}^{\tau} \sum_{a} T_L^k(s_1^1,a) \left(\underline{\widehat{\mu}}_L^{c,k}(s_1^1,a) + \sum_{s_j^2} P(s_j^2|s_1^1,a) \underline{Y}^{c}_{\mathbf{b}^k}(s_j^2) \right) \\
&\qquad < (1-\alpha) \sum_{k=1}^{\tau} T_L^k(s_1^1,a) \left(\mu^{c}(s_1^1,0) + \sum_{s_j^2} P(s_j^2|s_1^1,0) V^{c}_{\pi_0}(s_j^2) \right) \\
&\overset{(b)}{\implies} \underbrace{\sum_{a} T_L^{\tau}(s_1^1,a) \underline{\widehat{\mu}}_L^{c,\tau}(s_1^1,a)}_{\textbf{Part A}} + \sum_{a} T_L^{\tau}(s_1^1,a) \sum_{s_j^2} P(s_j^2|s_1^1,a) \underline{Y}^{c}_{\mathbf{b}^k}(s_j^2) \\
&\qquad < \underbrace{(1-\alpha) \sum_{a} T_L^{\tau}(s_1^1,0) \mu^{c}(s_1^1,0)}_{\textbf{Part B}} + (1-\alpha) T_L^{\tau}(s_1^1,0) \sum_{s_j^2} P(s_j^2|s_1^1,0) V^{c}_{\pi_0}(s_j^2)
\end{aligned}
\tag{C.39}
$$

where (a) follows as π_0 samples baseline action 0 for each state $s \in [S]$, and in (b) the $T_L^{\tau}(s_1^1,a)$ denotes the total samples of state-action tuple till episode τ. Comparing **Part A** and **Part B** for level $\ell = 1$ we observe that

the constraint violation must satisfy

$$\sum_{a} T_L^{\tau}(s_1^1, a)\underline{\widehat{\mu}}_L^{c,\tau}(s_1^1, a) < (1-\alpha)T_L^{\tau}(s_1^1, 0)\mu^c(s_1^1, 0)$$

which can be reduced by following the same way as step 7 as Theorem 2

$$T_L^{\tau-1}(s_1^1, 0) \leqslant \frac{1}{\alpha\mu^c(s_1^1, 0)}\left(1 + \sum_{a=1}^{A} N(s_1^1, a)\right).$$

where $\Delta^{c,\alpha}(s_1^1, a) \coloneqq (1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a)$ and

$$\begin{aligned} N(s_1^1, a) &\coloneqq T_L^{\tau-1}(s_1^1, a) \cdot \Big((1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) \\ &\qquad + c_1\sqrt{\log(An(n+1)/\delta)/T_L^{\tau-1}(s_1^1, a)}\Big) \\ &= \Delta^{c,\alpha}(s_1^1, a)T_L^{\tau-1}(s_1^1, a) + c_1\sqrt{\log(An(n+1)/\delta)T_L^{\tau-1}(s_1^1, a)} \end{aligned} \tag{C.40}$$

is a bound on the decrease in $\widehat{Z}_\tau$ in the first $\tau - 1$ rounds due to choosing action a in s_1^1. We will now bound $N(s_1^1, a)$ for each a. Now observe

$$\begin{aligned} \Delta^{c,\alpha}(s_1^1, a) &= (1-\alpha)\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) \\ &= \mu^c(s_1^1, 0) - \alpha\mu^c(s_1^1, 0) - \mu^c(s_1^1, a) \\ &= -(\mu^{*,c}(s_1^1) - \mu^c(s_1^1, 0)) - \alpha\mu^c(s_1^1, 0) + (\mu^{*,c}(s_1^1) - \mu^c(s_1^1, a)) \\ &= -\Delta^c(s_1^1, 0) - \alpha\mu^c(s_1^1, 0) + \Delta^c(s_1^1, a). \end{aligned}$$

where, $\mu^{*,c}(s_1^1) = \max_a \mu^c(s_1^1, a)$. It follows then that using step 7 as Theorem 2 for the state s_1^1

$$n_u(s_1^1) \leqslant \frac{1}{\alpha\mu^c(s_1^1, 0)}\left(1 + \sum_{a=1}^{A} N(s_1^1, a)\right) \leqslant \frac{H_{*,(2)}(s_1^1)}{2}\frac{n}{M(s_1^1)}$$

where

$$\mathrm{H}_{*,(2)}(s_i^\ell) \coloneqq \sum_a \mathbf{b}_*(a|s_i^\ell) \overset{+}{\min}\{\Delta^c(s_i^\ell, a), \Delta^c(s_i^\ell, 0) - \Delta^c(s_i^\ell, a)\},$$

$$\mathrm{M}(s_i^\ell) \coloneqq \sum_a \sqrt{\pi^2(a|s_i^\ell)\left(\sigma^2(s_i^\ell, a) + \sum_{s_j^{\ell+1}} \mathrm{P}(s_j^{\ell+1}|s_i^\ell, a)\mathrm{M}^2(s_j^{\ell+1})\right)} \tag{C.41}$$

Similarly, for an arbitrary level $\ell \in [\mathrm{L}]$, we can show using (C.39) that the constraint violation must satisfy

$$\begin{aligned}
&\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_a \mathrm{T}_\mathrm{L}^\tau(s_i^{\ell'}, a)\underline{\widehat{\mu}}_\mathrm{L}^{c,\tau}(s_i^{\ell'}, a) < (1-\alpha)\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \mathrm{T}_\mathrm{L}^\tau(s_i^{\ell'}, 0)\mu^c(s_i^{\ell'}, 0) \\
\overset{(a)}{\implies} &\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_a \left(\mathrm{T}_\mathrm{L}^{*,K}(s_i^{\ell'}, a) - 4A\mathbf{b}_*(a|s_i^{\ell'})\right)\underline{\widehat{\mu}}_\mathrm{L}^{c,\tau}(s_i^{\ell'}, a) \\
&< (1-\alpha)\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \left(\mathrm{T}_\mathrm{L}^{*,K}(s_i^{\ell'}, 0) + 4A\right)\mu^c(s_i^{\ell'}, 0) \\
\implies &\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_a \left(\mathrm{T}_\mathrm{L}^{*,K}(s_i^{\ell'}, a)\right)\underline{\widehat{\mu}}_\mathrm{L}^{c,\tau}(s_i^{\ell'}, a) \\
&< (1-\alpha)\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \left(\mathrm{T}_\mathrm{L}^{*,K}(s_i^{\ell'}, 0)\right)\mu^c(s_i^{\ell'}, 0) + 8LSA^2(\mu^c(s_i^{\ell'}, 0) + \underline{\widehat{\mu}}_\mathrm{L}^{c,\tau}(s_i^{\ell'}, a))
\end{aligned}$$

$$
\begin{aligned}
\implies & \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_{a} \left(T_L^{*,K}(s_i^{\ell'}, a) \right) \underline{\widehat{\mu}}_L^{c,\tau}(s_i^{\ell'}, a) \\
& < (1-\alpha) \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \left(T_L^{*,K}(s_i^{\ell'}, 0) \right) \mu^c(s_i^{\ell'}, 0) \\
& + 8LSA^2 (\mu^c(s_i^{\ell'}, 0) + \widehat{\mu}_L^{c,\tau}(s_i^{\ell'}, a) - \sqrt{\frac{\log((SAn(n+1)/\delta)}{2T_L^{\tau}(s_i^{\ell'}, a)}}) \\
\overset{(b)}{\implies} & \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_{a} \left(T_L^{*,K}(s_i^{\ell'}, a) \right) \underline{\widehat{\mu}}_L^{c,\tau}(s_i^{\ell'}, a) \\
& < (1-\alpha) \max_{s,a} \mu_0^c(s, a) \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \left(T_L^{*,K}(s_i^{\ell'}, 0) \right) + 16LSA^2 \qquad \text{(C.42)}
\end{aligned}
$$

where, (a) follows from (C.38) and (b) follows as $\mu(s, a) \in (0, 1]$ for all s, a. It follows then that using step 7 of Theorem 2 and definition of $N(s_j^{\ell})$ from (C.40)

$$
\begin{aligned}
\sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} T_L^{*,K}(s_i^{\ell'}, 0) & \leqslant \frac{1}{\alpha \max_s \mu^c(s, 0)} \left(1 + \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_{a} N(s_j^{\ell}, a) \right) \\
& \leqslant \frac{n}{2} \sum_{\ell'=1}^{\ell} \sum_{s_i^{\ell'}} \sum_{a} \frac{H_{*,(2)}(s_i^{\ell'})}{M(s_i^{\ell'})} + 16LSA^2
\end{aligned}
$$

which gives a bound on how many times action $\{0\}$ is sampled across different states till level ℓ. Summing over all states s_j^{ℓ} till level L we can show that

$$
n_u = \sum_{\ell=1}^{L} \sum_{s_j^{\ell}} T_L^{*,K}(s_j^{\ell}, 0) \leqslant \frac{n}{2} \sum_{\ell=1}^{L} \sum_{s_j^{\ell}} \frac{H_{*,(2)}(s_j^{\ell})}{M(s_j^{\ell})} + 16LSA^2 \overset{(a)}{\leqslant} \frac{H_{*,(2)}}{2} \frac{n}{M_{\min}} + 16LSA^2 \qquad \text{(C.43)}
$$

where, in (a) we define $M_{\min} = \min_s M(s)$, and $H_{*,(2)} = \sum_{\ell=1}^{L} \sum_{s_j^\ell} H_{*,(2)}(s_j^\ell)$. Finally, observe that 16LSA^2 does not depend on the episode K.

Step 8 (Lower bound to Constraint violation): For the lower bound to the constraint we equate Equation (C.39) to 0 and show that

$$\underbrace{\sum_a T_L^\tau(s_1^1,a)\widehat{\underline{\mu}}_L^{c,\tau}(s_1^1,a)}_{\textbf{Part A}} + \sum_a T_L^\tau(s_1^1,a)\sum_{s_j^2} P(s_j^2|s_1^1,a)\underline{Y}_{\mathbf{b}^k}^c(s_j^2)$$
$$= \underbrace{(1-\alpha)\sum_a T_L^\tau(s_1^1,0)\mu^c(s_1^1,0)}_{\textbf{Part B}} + (1-\alpha)T_L^\tau(s_1^1,0)\sum_{s_j^2} P(s_j^2|s_1^1,0)V_{\pi_0}^c(s_j^2)$$

Again comparing **Part A** and **Part B** for level $\ell = 1$ we observe that the lower bound to constraint violation must satisfy

$$\sum_a T_L^\tau(s_1^1,a)\widehat{\underline{\mu}}_L^{c,\tau}(s_1^1,a) = (1-\alpha)T_L^\tau(s_1^1,0)\mu^{c,}(s_1^1,0)$$

which can be reduced by following the same way as step 8 as Theorem 2

$$\sum_a T_L^{\tau-1}(s_1^1,0) \geqslant \frac{1}{\alpha\mu^c(s_1^1,0)}\left(1+\sum_{a=1}^{A}\underline{N}(s_1^1,a)\right).$$

where $\Delta^{c,\alpha}(s_1^1,a) \coloneqq (1-\alpha)\mu^c(s_1^1,0) - \mu^c(s_1^1,a)$ and

$$\begin{aligned}\underline{N}(s_1^1,a) &\coloneqq T_L^{\tau-1}(s_1^1,a)\cdot\Big((1-\alpha)\mu^c(s_1^1,0) - \mu^c(s_1^1,a) \\ &\qquad + c_1\sqrt{\log(An(n+1)/\delta)/T_L^{\tau-1}(s_1^1,a)}\Big) \\ &= \Delta^{c,\alpha}(s_1^1,a)T_L^{\tau-1}(s_1^1,a) + c_1\sqrt{\log(An(n+1)/\delta)T_L^{\tau-1}(s_1^1,a)} \\ &\overset{(a)}{\geqslant} \Delta^{c,\alpha}(s_1^1,a)\left(T_L^{*,K}(s_1^1,a) - 4A\mathbf{b}_*(a|s_1^1)\right) \\ &\qquad + c_1\sqrt{\log(An(n+1)/\delta)\left(T_L^{*,K}(s_1^1,a) - 4A\mathbf{b}_*(a|s_1^1)\right)}\end{aligned}$$

where, (a) follows from (C.37). Then following the same way as step 8 of Theorem 2 we can show that

$$\begin{aligned}
T_L^{\tau-1}(s_1^1,0) &\geqslant \frac{1}{\alpha\mu^c(s_1^1,0)}\left(1+\sum_{a=1}^{A}\underline{N}(s_1^1,a)\right)\\
&\geqslant \frac{n_f}{M(s_1^1)}\left(\frac{H_{*,(2)}(s_1^1)}{8}-\frac{A}{2}\frac{H_{*,(2)}(s_1^1)}{M(s_1^1)}\right)-16SA
\end{aligned}$$

Similarly for any arbitrary level $\ell \in [L]$ following the same way as step 7 above it can be shown that

$$\begin{aligned}
&\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\sum_{a}\left(T_L^{*,K}(s_i^{\ell'},a)+4A\right)\underline{\widehat{\mu}}_L^{c,\tau}(s_i^{\ell'},a)\\
&\geqslant (1-\alpha)\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\left(T_L^{*,K}(s_i^{\ell'},0)-4A\mathbf{b}_*(0|s_i^{\ell'})\right)\mu^c(s_i^{\ell'},0)\\
\implies &\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}\sum_{a}T_L^{*,K}(s_j^{\ell},a)\underline{\widehat{\mu}}_L^{c,\tau}(s_j^{\ell},a)\\
&\geqslant (1-\alpha)\sum_{\ell'=1}^{\ell}\sum_{s_i^{\ell'}}T_L^{*,K}(s_i^{\ell},0)\mu^c(s_i^{\ell},0)-16LSA^2
\end{aligned}$$

Again following the same way as step 8 of Theorem 2 for the state s_j^{ℓ}, the lower bound to the total number of times the baseline actions are sampled across states till level ℓ is given by we can show that

$$\begin{aligned}
\sum_{\ell'=1}^{\ell}\sum_{s_j^{\ell'}}T_L^{*,K}(s_j^{\ell'},0) &\geqslant \frac{1}{\alpha\max_{s_j^{\ell}}\mu^c(s_j^{\ell},0)}\left(1+\sum_{\ell'=1}^{\ell}\sum_{s_j^{\ell'}}\sum_{a=1}^{A}\underline{N}(s_j^{\ell'},a)\right)\\
&\geqslant \sum_{\ell=1}^{L}\sum_{s_j^{\ell}}\frac{n_f}{M(s_j^{\ell})}\left(\frac{H_{*,(2)}(s_j^{\ell})}{8}-\frac{A}{2}\frac{H_{*,(2)}(s_j^{\ell})}{M(s_j^{\ell})}\right)-16LSA^2
\end{aligned}$$

Finally summing over all states s_j^ℓ and level L we can show that

$$\sum_{\ell=1}^{L}\sum_{s_j^\ell} T_L^{*,K}(s_j^\ell,0) \geqslant \sum_{\ell=1}^{L}\sum_{s_j^\ell} \frac{n_f}{M(s_j^\ell)}\left(\frac{H_{*,(2)}(s_j^\ell)}{8} - \frac{A}{2}\frac{H_{*,(2)}(s_j^\ell)}{M(s_j^\ell)}\right) - 16LSA^2 \tag{C.44}$$

Again, observe that $16LSA^2$ does not depend on the episode K.

Step 9 (Bound Part B): Then from (C.44) we can show that

$$\begin{aligned}
\frac{M(s_1^1)}{\sum_{\ell=1}^{L}\sum_{s_j^\ell} T_L^{*,K}(s_j^\ell,0)} &\leqslant \frac{M(s_1^1)}{\sum_{\ell=1}^{L}\sum_{s_j^\ell} \frac{n_f}{M(s_j^\ell)}\left(\frac{H_{*,(2)}(s_j^\ell)}{8} - \frac{A}{2}\frac{H_{*,(2)}(s_j^\ell)}{M(s_j^\ell)}\right) - 16LSA^2} \\
&\overset{(a)}{\leqslant} (M(s_1^1)+16LSA^2)\sum_{\ell=1}^{L}\sum_{s_j^\ell}\frac{M(s_j^\ell)}{n_f}\left(\frac{H_{*,(2)}(s_j^\ell)}{8} + \frac{A}{2}\frac{H_{*,(2)}(s_j^\ell)}{M(s_j^\ell)}\right) \\
&\leqslant (M(s_1^1)+16LSA^2)\sum_{\ell=1}^{L}\sum_{s_j^\ell}\frac{M(s_j^\ell)}{n_f}\left(2+H_{*,(2)}(s_j^\ell)\right) \\
&\overset{(b)}{\leqslant} (M(s_1^1)+16LSA^2)\frac{M}{n_f}\left(2+H_{*,(2)}\right)
\end{aligned}$$

where, (a) follows for $1/(x-c) \leqslant x + c$ for $x^2 \geqslant 1 + c^2$ and $c > 0$. The (b) follows for $M = \sum_{\ell=1}^{L}\sum_{s_j^\ell} M(s_j^\ell)$, and $H_{*,(2)} = \sum_{\ell=1}^{L}\sum_{s_j^\ell} H_{*,(2)}(s_j^\ell)$. It follows then by setting $n_f = n - n_u$ that

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{Z,K}^C\}\right] &\overset{(a)}{\leqslant} \left(\frac{(M(s_1^1)+16LSA^2)M\left(2+H_{*,(2)}\right)}{n_f}\right)^2 n_u \\
&\overset{(b)}{=} \frac{(M(s_1^1)+16LSA^2)^2 n_u}{(n-n_u)^2}\left(2+H_{*,(2)}\right)^2 \\
&\overset{(c)}{\leqslant} \frac{(M(s_1^1)+16LSA^2)^2 H_{*,(2)} n}{(n-H_{*,(2)}n)^2}\left(2+H_{*,(2)}\right)^2 \\
&\leqslant \frac{M^2(s_1^1)}{n}\left(32MLSA^2 + H_{*,(2)}\right)^2
\end{aligned}$$

where, (a) follows from Theorem C.1, (b) follows from the definition of $H_{*,(2)}$, and (c) follows from (C.43).

Step 10 (Combine everything): Combining everything from step 5, step 8 and setting $\delta = 1/n^2$ we can show that the MSE of oracle scales as

$$\begin{aligned}\mathcal{L}_n(\pi, \mathbf{b}_*^k) &\leqslant \frac{M^2(s_1^1)}{n} + \frac{8AM^2(s_1^1)}{n^2} + \frac{16A^2M^2(s_1^1)}{n^3} + \frac{M^2(s_1^1)}{n}\left(32MLSA^2 + H_{*,(2)}\right)^2 \\ &\quad + \underbrace{\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{c,K}^C\}\right]}_{\textbf{Part C, Safety event does not hold}} \\ &\overset{(a)}{\leqslant} \frac{M^2(s_1^1)}{n} + \frac{8AM^2(s_1^1)}{n^2} + \frac{16A^2M^2(s_1^1)}{n^3} + \frac{M^2(s_1^1)}{n}\left(32MLSA^2 + H_{*,(2)}\right)^2 \\ &\quad + 2\sum_{t=1}^{n} \frac{2\eta + 4\eta^2}{n^2} \end{aligned} \tag{C.45}$$

where, (a) follows as $\mathbb{E}_{\mathcal{D}}\left[\left(Y_n(s_1^1) - V_\pi(s_1^1)\right)^2 \mathbb{I}\{\xi_{c,K}^C\}\right] \leqslant 2\eta + 4\eta^2$ and using the low error probability of the constraint event from Theorem C.14. The claim of the proposition follows. □

Tree Regret Corollary

Corollary 1. *Under Assumption 6 the constraint regret in the Tree MDP is given by* $\overline{\mathcal{R}}_n^c \leqslant O\left(\frac{\log(n)}{b_{*,\min}^{3/2} n^{3/2}}\right)$ *and the regret is given by* $\overline{\mathcal{R}}_n \leqslant O\left(\frac{\log(n)}{b_{*,\min}^{3/2} n^{3/2}}\right)$.

Proof. The upper bound to the safe oracle constraint is given by (C.43) as follows

$$\mathcal{C}_n^*(\pi, \mathbf{b}_*^k) \leqslant \frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + 16LSA^2.$$

The upper bound to the constraint violation of SaVeR is given by (C.26)

$$\mathcal{C}_n(\pi, \widehat{b}^k) \leqslant \frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + 16LSA^2 + O\left(\frac{(2\eta + 4\eta^2)L^2S^2A^4H^2_{*,(2)}M^2\sqrt{\log(SAn(n+1)/\delta)}}{\min_s \mathbf{b}^{*,k,(3/2)}(s)n^{3/2}}\right).$$

Hence, from the constraint regret definition, we can show that

$$\overline{\mathcal{R}}^c_n = \mathcal{C}_n(\pi, \widehat{b}^k) - \overline{\mathcal{C}}^*_n(\pi, \mathbf{b}_*) \leqslant O\left(\frac{\log n}{\mathbf{b}^{3/2}_{*,\min} n^{3/2}}\right).$$

Observe that the loss of the agnostic algorithm SaVeR is given by (C.28) and the upper bound to the oracle loss is given by (C.45). Comparing these two losses directly leads to the regret as follows:

$$\overline{\mathcal{R}}_n = \mathcal{L}_n(\pi, \widehat{b}^k) - \overline{\mathcal{L}}^*_n(\pi, \mathbf{b}^k_*) = O\left(\frac{\log(n)}{\mathbf{b}^{3/2}_{*,\min} n^{3/2}}\right).$$

The claim of the corollary follows. □

Corollary 2. *Under Assumption 6 the constraint regret in the bandit setting is given by* $\overline{\mathcal{R}}^c_n \leqslant O\left(\frac{\log(n)}{\mathbf{b}^{3/2}_{*,\min} n^{3/2}}\right)$ *and the regret is given by* $\overline{\mathcal{R}}_n \leqslant O\left(\frac{\log(n)}{\mathbf{b}^{3/2}_{*,\min} n^{3/2}}\right)$.

Proof. The bandit setting consists of a single state, and so we can define the quantity $H_{*,(2)} = \frac{1}{\alpha\mu(0)}\sum_{a\in\mathcal{A}\setminus\{0\}} \pi(a)\sigma(a)\min^+\{\Delta^c(a), \Delta^c(0) - \Delta^c(a)\}$ The upper bound to the oracle constraint is given by (C.43) as follows

$$\mathcal{C}^*_n(\pi, \mathbf{b}^k_*) \leqslant \frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + 16A^2.$$

The upper bound to the constraint violation of SaVeR is given by (C.26)

$$\mathcal{C}_n(\pi, \widehat{b}^k) \leqslant \frac{H_{*,(2)}}{2}\frac{n}{M_{\min}} + 16A^2 + O\left(\frac{(2\eta + 4\eta^2)A^4 H_{*,(2)}^2 M^2 \sqrt{\log(An(n+1)/\delta)}}{\min_s \mathbf{b}^{*,k,(3/2)}(s) n^{3/2}}\right).$$

Hence, from the constraint regret definition, we can show that

$$\overline{\mathcal{R}}_n^c = \mathcal{C}_n(\pi, \widehat{b}^k) - \overline{\mathcal{C}}_n^*(\pi, \mathbf{b}_*^k) \leqslant O\left(\frac{\log n}{\mathbf{b}_{*,\min}^{3/2} n^{3/2}}\right).$$

Observe that the loss of the agnostic algorithm SaVeR is given by (C.28) and the upper bound to the oracle loss is given by (C.45). Comparing these two losses directly leads to the regret as follows:

$$\overline{\mathcal{R}}_n = \mathcal{L}_n(\pi, \widehat{b}^k) - \overline{\mathcal{L}}_n^*(\pi, \mathbf{b}_*^k) = O\left(\frac{\log(n)}{\mathbf{b}_{*,\min}^{3/2} n^{3/2}}\right).$$

The claim of the corollary follows. □

C.5 Support Lemmas

Lemma C.11. (***Hoeffding's Lemma***)(*Massart, 2007*) *Let Y be a real-valued random variable with expected value $\mathbb{E}[Y] = \mu$, such that $a \leqslant Y \leqslant b$ with probability one. Then, for all $\lambda \in \mathbb{R}$*

$$\mathbb{E}\left[e^{\lambda Y}\right] \leqslant \exp\left(\lambda\mu + \frac{\lambda^2(b-a)^2}{8}\right)$$

Lemma C.12. (***Concentration lemma 1***) *Let $V_t = R_t(s,a) - \mathbb{E}[R_t(s,a)]$ and be bounded such that $V_t \in [-\eta, \eta]$. Let the total number of times the state-action*

(s, a) *is sampled be* T. *Then we can show that for an* $\epsilon > 0$

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T} R_t(s,a) - \mathbb{E}[R_t(s,a)]\right| \geqslant \epsilon\right) \leqslant 2\exp\left(-\frac{2\epsilon^2 T}{\eta^2}\right).$$

Proof. Let $V_t = R_t(s,a) - \mathbb{E}[R_t(s,a)]$. Note that $\mathbb{E}[V_t] = 0$. Hence, for the bounded random variable $V_t \in [-\eta, \eta]$ we can show from Hoeffding's lemma in Theorem C.11 that

$$\mathbb{E}[\exp(\lambda V_t)] \leqslant \exp\left(\frac{\lambda^2}{8}(\eta - (-\eta))^2\right) \leqslant \exp\left(2\lambda^4\eta^2\right)$$

Let s_{t-1} denote the last time the state s is visited and action a is sampled. Observe that the reward $R_t(s,a)$ is conditionally independent and η^2-sub-

Gaussian. Next we can bound the probability of deviation as follows:

$$
\begin{aligned}
&\mathbb{P}\left(\sum_{t=1}^{T}\left(R_t(s,a)-\mathbb{E}[R_t(s,a)]\right)\geqslant\epsilon\right)\\
&=\mathbb{P}\left(\sum_{t=1}^{T}V_t\geqslant\epsilon\right)\\
&\overset{(a)}{=}\mathbb{P}\left(e^{\lambda\sum_{t=1}^{T}V_t}\geqslant e^{\lambda\epsilon}\right)\\
&\overset{(b)}{\leqslant} e^{-\lambda\epsilon}\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T}V_t}\right]\\
&=e^{-\lambda\epsilon}\mathbb{E}\left[\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T}V_t}\middle|s_{T-1}\right]\right]\\
&\overset{(c)}{=} e^{-\lambda\epsilon}\mathbb{E}\left[\mathbb{E}\left[e^{-\lambda V_T}|S_{T-1}\right]\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T-1}V_t}\middle|s_{T-1}\right]\right]\\
&\leqslant e^{-\lambda\epsilon}\mathbb{E}\left[\exp\left(2\lambda^4\eta^2\right)\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T-1}V_t}\middle|s_{T-1}\right]\right]\\
&=e^{-\lambda\epsilon}e^{2\lambda^2\eta^2}\mathbb{E}\left[e^{-\lambda\sum_{t=1}^{T-1}V_t}\right]\\
&\vdots\\
&\overset{(d)}{\leqslant} e^{-\lambda\epsilon}e^{2\lambda^2T\eta^2}\\
&\overset{(e)}{\leqslant}\exp\left(-\frac{2\epsilon^2}{T\eta^2}\right)
\end{aligned}
\tag{C.46}
$$

where (a) follows by introducing $\lambda\in\mathbb{R}$ and exponentiating both sides, (b) follows by Markov's inequality, (c) follows as V_t is conditionally independent given s_{T-1}, (d) follows by unpacking the term for T times and (e)

follows by taking $\lambda = \epsilon/4T\eta^2$. Hence, it follows that

$$\begin{aligned}
&\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=1}^{T} R_t(s,a) - \mathbb{E}[R_t(s,a)]\right| \geqslant \epsilon\right) \\
&= \mathbb{P}\left(\sum_{t=1}^{T} (R_t(s,a) - \mathbb{E}[R_t(s,a)]) \geqslant T\epsilon\right) \\
&\overset{(a)}{\leqslant} 2\exp\left(-\frac{2\epsilon^2 T}{\eta^2}\right).
\end{aligned}$$

where, (a) follows by (C.46) by replacing ϵ with ϵT, and accounting for deviations in either direction. □

Lemma C.13. (***Concentration lemma 2***) *Let* $\mu^2(s,a) = \mathbb{E}\left[R_t^2(s,a)\right]$. *Let* $R_t(s,a)$ *be* η^2 *sub-Gaussian. Let* $n = KL$ *be the total budget of state-action samples. Define the event*

$$\begin{aligned}
\xi_\delta = &\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t^2(s,a) - \mu^2(s,a)\right| \leqslant C_n(\delta)\right\}\right)\bigcap \\
&\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t(s,a) - \mu(s,a)\right| \leqslant C_n(\delta)\right\}\right)
\end{aligned} \tag{C.47}$$

where, $C_n(\delta) = C_n(\delta)$. *Then we can show that* $\mathbb{P}(\xi_\delta) \geqslant 1 - 2\delta$.

Proof. First note that the total budget $n = KL$. Observe that the random variable $R_t^k(s,a)$ and $R_t^{(2),k}(s,a)$ are conditionally independent given the previous state S_{t-1}^k. Also observe that for any $\eta > 0$ we have that $R_t^k(s,a), R_t^{(2),k}(s,a) \leqslant 2\eta + 4\eta^2$, where $R_t^{(2),k}(s,a) = (R_t^k(s,a))^2$. Hence we

can show that

$$\mathbb{P}\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t^2(s,a)-\mu^2(s,a)\right|\geqslant C_n(\delta)\right\}\right)$$
$$\leqslant \mathbb{P}\left(\bigcup_{s\in\mathcal{S}}\bigcup_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t^2(s,a)-\mu^2(s,a)\right|\geqslant C_n(\delta)\right\}\right)$$
$$\overset{(a)}{\leqslant}\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{t=1}^{n}\sum_{T_n(s,a)=1}^{t} 2\exp\left(-\frac{2T_n}{4(\eta^2+\eta)^2}\cdot\frac{4(\eta^2+\eta)^2\log(SAn(n+1)/\delta)}{2T_n(s,a)}\right)=\delta.$$

where, (a) follows from Theorem C.12. Note that in (a) we have to take a double union bound summing up over all possible pulls T_n from 1 to n as T_n is a random variable. Similarly we can show that

$$\mathbb{P}\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1}\left\{\left|\frac{1}{T_n(s,a)}\sum_{t=1}^{T_n(s,a)} R_t(s,a)-\mu(s,a)\right|\geqslant C_n(\delta)\right\}\right)$$
$$\overset{(a)}{\leqslant}\sum_{s=1}^{S}\sum_{a=1}^{A}\sum_{t=1}^{n}\sum_{T_n(s,a)=1}^{t} 2\exp\left(-\frac{2T_n}{4(\eta^2+\eta)^2}\cdot\frac{4(\eta^2+\eta)^2\log(SAn(n+1)/\delta)}{2T_n(s,a)}\right)=\delta.$$

where, (a) follows from Theorem C.12. Hence, combining the two events above we have the following bound

$$\mathbb{P}(\xi_\delta)\geqslant 1-2\delta.$$

□

Corollary 3. *Under the event ξ_δ in (C.47) we have for any state-action pair in an episode k the following relation with probability greater than $1-\delta$*

$$|\widehat{\sigma}_t^k(s,a)-\sigma(s,a)|\leqslant(2\eta+4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_L^K(s,a)}}.$$

where, $T_L^K(s,a)$ *is the total number of samples of the state-action pair* (s,a) *till episode* k*.*

Proof. Observe that the event ξ_δ bounds the sum of rewards $R_t^k(s,a)$ and squared rewards $R_t^{k,(2)}(s,a)$ for any $T_L^K(s,a) \geqslant 1$. Hence we can directly apply the Theorem C.13 to get the bound. □

Lemma C.14. *Let* $\mu^c(s,a) = \mathbb{E}\left[C_t(s,a)\right]$ *and* $C_t(s,a) \leqslant 2\eta$*. Define the event*

$$\overline{\xi}_\delta = \bigcap_{s\in\mathcal{S}} \bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1} \left\{ \left| \frac{1}{T_n(s,a)} \sum_{t=1}^{T_n(s,a)} C_t(s,a) - \mu^c(s,a) \right| \leqslant C_n(\delta) \right\}. \tag{C.48}$$

Then we can show that $\mathbb{P}(\overline{\xi}_\delta) \geqslant 1-\delta$*.*

Proof. We can show that

$$\begin{aligned}
&\mathbb{P}\left(\bigcap_{s\in\mathcal{S}} \bigcap_{1\leqslant a\leqslant A, T_n(s,a)\geqslant 1} \left\{ \left| \frac{1}{T_n(s,a)} \sum_{t=1}^{T_n(s,a)} C_t(s,a) - \mu^c(s,a) \right| \right.\right. \\
&\qquad \left.\left. \geqslant C_n(\delta) \right\} \right) \\
&\overset{(a)}{\leqslant} \sum_{s=1}^{S} \sum_{a=1}^{A} \sum_{t=1}^{n} \sum_{T_n(s,a)=1}^{t} 2\exp\left(-\frac{2T_n(s,a)}{4(\eta^2+\eta)^2} \cdot \frac{4(\eta^2+\eta)^2 \log(SAn(n+1)/\delta)}{2T_n(s,a)} \right) = \delta.
\end{aligned}$$

where, (a) follows from Theorem A.4 when applied for cost. The claim of the lemma follows. □

Corollary 4. *Let the total exploration budget be* $n_x = \frac{SA\log(SAn(n+1)/\delta)}{\min_{s,a}\Delta^{c,(2)}(s,a)}$*. Define the event* $\overline{\xi}_\delta$ *as in* (C.48)*. Then using the exploration policy* π_x *it can be shown that* $\mathbb{P}(\overline{\xi}_\delta) \geqslant 1-\delta$*.*

Proof. Let $n_x = \frac{SA\log(SAn(n+1)/\delta)}{\min_{s,a}\Delta^{c,(2)}(s,a)}$ be the total samples taken for exploration. Let π_e sample each action according to uniform random policy in

each state $s \in [S]$. Then the result follows directly from Theorem C.14 in

$$\mathbb{P}\left(\bigcap_{s\in\mathcal{S}}\bigcap_{1\leqslant a\leqslant A, T_{n_x}(s,a)\geqslant 1}\left\{\left|\frac{1}{T_{n_x}(s,a)}\sum_{t=1}^{T_{n_x}(s,a)} C_t(s,a) - \mu^c(s,a)\right| \geqslant (2\eta+4\eta^2)\sqrt{\frac{\log(SAn(n+1)/\delta)}{2T_{n_x}}}\right\}\right) \overset{(a)}{\leqslant} \delta,$$

where, (a) follows as by noting $T_{n_x} \geqslant \frac{\log(SAn(n+1)/\delta)}{\min_{s,a}\Delta^{c,(2)}(s,a)}$.

□

C.6 Additional Experimental Details

In this section we state additional experimental details.

Experiment 1 (Bandit): We implement a bandit environment for $A = 11$ and show that our proposed solution outperforms the safe on-policy and SEPEC (Wan et al., 2022) algorithm. In this experiment we have the $\mu(0) = 0.5, \sigma^2(0) = 10^{-4}$, $\mu(1) = 0.9, \sigma^2(1) = 10^{-4}$ (optimal action), and the sub-optimal actions $a \in \{2, 3, \ldots, 11\}$ have means $\mu(a) \in [0.02, 0.03]$ and high variance $\sigma^2(a) = 40$. Moreover, we set the constraint-value means $\mu^c(a)$ the same as the reward means. The target policy is initialized as $\pi(0) = \pi(1) = 0.4$ while the remaining arms have the 0.2 density evenly distributed among them. So in this environment, the safe on-policy will select the sub-optimal actions less and so reduces MSE at a slower rate. Whereas the SaVeR, complies with the safety constraint and reduces MSE maximally as the number of rounds increases. The performance is shown in Figure 4.1 (left). Again observe that in Figure 4.2 (top-left), the oracle keeps the safety budget around 0 and uses all the remaining samples to explore optimally. The SaVeR has a safety budget of almost around 0 as they sample the high cost maximizing action 1 a sufficient number of times to offset the unsafe action pulls. However, safe on-policy and SEPEC again

explores the high variance (sub-optimal and unsafe) actions less and has a very high safety budget.

Experiment 2 (Movielens): We conduct this experiment on Movielens dataset for $A = 30$ actions and show that our proposed solution outperforms safe on-policy and SEPEC algorithm. The Movielens dataset from February 2003 consist of 6k users who give 1M ratings to 4k movies. We obtain a rank-4 approximation of the dataset over 128 users and 128 movies such that all users prefer either movies 7, 13, 16, or 20 (4 user groups). The movies are the actions and we choose 30 movies that have been rated by all the users. Hence, this testbed consists of 30 actions and the mean values $\mu(a)$ are the rating of the movies given by the users. and is run over $T = 8000$. The target policy is initialized as $\pi(0) = \pi(1) = 0.4$ while the remaining arms has the 0.2 density evenly distributed among them. We set the cost means $\mu_c(a)$ such that high variance actions have high-cost means. So in this environment, the safe on-policy will select the sub-optimal cost actions less and so reduces MSE at a slower rate as the number of rounds increases. The SEPEC MSE also reduces slower than SaVeR as the number of rounds increases. This is because SEPEC uses an IPW estimator instead of tracking the optimal behavior policy like SaVeR. The SaVeR, complies with the safety constraint and reduces MSE maximally as the number of rounds increases. The performance is shown in Figure 4.1 (middle-left). Again observe that in Figure 4.2 (top-right), the oracle keeps the safety budget around 0 and uses all the remaining samples to explore optimally. The SaVeR has a safety budget of almost around 0 as they sample the high reward maximizing action 1 a sufficient number of times to offset the unsafe action pulls. However, safe on-policy and SEPEC again explores the high variance (sub-optimal and unsafe) actions less and has a very high safety budget.

Experiment 3 (Tree): We experiment with a 4-depth 2-action deterministic tree MDP **T** consisting of 15 states. In this setting, we have a

4-depth 2-action deterministic tree MDP $\mathbf{T}$ consisting of 15 states. Each state has a low variance arm with $\sigma^2(s,1) = 0.01$ and high target probability $\pi(1|s) = 0.95$ and a high variance arm with $\sigma^2(s,1) = 20.0$ and low target probability $\pi(2|s) = 0.05$. Again we set the cost means $\mu^c(a)$ such that high variance actions have high-cost means. Hence, the safe on-policy sampling which samples according to π will sample the second (high variance) arms less and suffer a high MSE. We set $\alpha = 0.25$. We assume that the learner can directly access the $V^{\pi_0}(s_1^1)$ (without any noise) when its safety budget is negative. It can observe $V^{\pi_0}(s_1^1)$ without running any episodic interaction (like Yang et al. (2021b). The oracle has access to the model and variances and performs the best. SaVeR lowers MSE comparable to safe onpolicy as the number of episodes increases and eventually matches the oracle's MSE in Figure 4.1 (middle-right). The SaVeR, oracle, and on-policy have an almost equal safety budget as shown in Figure 4.2 (bottom-left). Note that we do not run SEPEC in this experiment as it is a bandit algorithm, and the optimization problem of SEPEC do not have a closed form solution in the MDP setting.

Experiment 4 (Gridworld): In this setting we have a 4×4 stochastic gridworld consisting of 16 grid cells. Considering the current episode time-step as part of the state, this MDP is a DAG MDP in which there is multiple paths to a single state. There is a single starting location at the top-left corner and a single terminal state at the bottom-right corner. Let $\mathbf{L}, \mathbf{R}, \mathbf{D}, \mathbf{U}$ denote the left, right, down, and up actions in every state. Then in each state, the right and down actions have low variance arms with $\sigma^2(s,\mathbf{R}) = \sigma^2(s,\mathbf{D}) = 0.01$ and high target policy probability $\pi(\mathbf{R}|s) = \pi(\mathbf{D}|s) = 0.45$. The left and top actions have high variance arms with $\sigma^2(s,\mathbf{L}) = \sigma^2(s,\mathbf{U}) = 0.01$ and low target policy probability $\pi(\mathbf{L}|s) = \pi(\mathbf{U}|s) = 0.05$. We set the cost means $\mu^c(a)$ such that high variance actions have high-cost means. Hence, safe onpolicy which goes right and down with high probability (to reach the terminal state) will sample the low

variance arms more and suffer a high MSE. We set $\alpha = 0.25$. Again we assume that the learner can directly access the $V^{\pi_0}(s_1)$ (without any noise) when it's safety budget is negative. It can observe $V^{\pi_0}(s_1)$ without running any episodic interaction (like Yang et al. (2021b). SaVeR lowers MSE faster compared to safe onpolicy and actually matches MSE compared to the oracle as well as maintains the safety constraint with increasing number of episodes. We point out that the DAG structure of the Gridworld violates the tree structure under which the oracle and SaVeR bounds were derived. Nevertheless, both methods lower MSE compared to safe onpolicy. Again observe that in Figure 4.2 (bottom-right), the oracle keeps the safety budget around 0 and uses all the remaining samples to explore optimally. The SaVeR has a safety budget of almost around 0 as they sample the high reward maximizing action a sufficient number of times to offset the unsafe action pulls. However, safe on-policy again explores the high variance (sub-optimal and unsafe) actions less and has a very high safety budget.

C.7 Table of Notations

Notations	Definition
s_i^ℓ	State s in level ℓ indexed by i
$\pi(a\|s_i^\ell)$	Target policy probability for action a in s_i^ℓ
$b(a\|s_i^\ell)$	Behavior policy probability for action a in s_i^ℓ
$\sigma^2(s_i^\ell, a)$	Variance of action a in s_i^ℓ
$\widehat{\sigma}_t^{(2),k}(s_i^\ell, a)$	Empirical variance of action a in s_i^ℓ at time t in episode k
$\widehat{\sigma^u}_t^{(2),k}(s_i^\ell, a)$	UCB on variance of action a in s_i^ℓ at time t in episode k
$\mu(s_i^\ell, a)$	Mean of action a in s_i^ℓ
$\widehat{\mu}_t^k(s_i^\ell, a)$	Empirical mean of action a in s_i^ℓ at time t in episode k
$\mu^2(s_i^\ell, a)$	Square of mean of action a in s_i^ℓ
$\widehat{\mu}_t^{(2),k}(s_i^\ell, a)$	Square of empirical mean of action a in s_i^ℓ at time t in episode k
$T_n(s_i^\ell, a)$	Total Samples of action a in s_i^ℓ after n timesteps
$T_n(s_i^\ell)$	Total samples of actions in s_i^ℓ as $\sum_a T_n(s_i^\ell, a)$ after n timesteps (State count)
$T_t^k(s_i^\ell, a)$	Total samples of action a taken till episode k time t in s_i^ℓ
$T_t^k(s_i^\ell, a, s_j^{\ell+1})$	Total samples of action a taken till episode k time t in s_i^ℓ to transition to $s_j^{\ell+1}$
$P(s_j^{\ell+1}\|s_i^\ell, a)$	Transition probability of taking action a in state s_i^ℓ and transition to state $s_j^{\ell+1}$

Table C.1: Table of Notations for SaVeR

D APPENDIX: MULTI-TASK REPRESENTATION LEARNING FOR PURE EXPLORATION IN BILINEAR BANDITS

D.1 Probability Tools and Previous Results

In this section, we state useful lemmas we use in our proofs and previous results.

Lemma D.1. (*Generalized Stein's Lemma,* (*Stein et al., 2004*)) *For a random variable* X *with continuously differentiable density function* $\mathbf{p} : \mathbb{R}^d \to \mathbb{R}$*, and any continuously differentiable function* $f : \mathbb{R}^d \to \mathbb{R}$*. Let* $Q(\cdot)$ *be a scoring function defined in Theorem D.5. If the expected values of both* $\nabla f(X)$ *and* $f(X) \cdot Q(X)$ *regarding the density* p *exist, then they are identical, i.e.*

$$\mathbb{E}[f(X) \cdot Q(X)] = \mathbb{E}[\nabla f(X)].$$

Lemma D.2. (***Minsker, 2018***) *Define* $\|\mathbf{A}\|_{op}$ *as the operator norm of* $\mathbf{A}$*. Let* $\mathbf{Y}_1, \ldots, \mathbf{Y}_n \in \mathbb{R}^{d_1 \times d_2}$ *be a sequence of independent real random matrices, and assume that*

$$\sigma_n^2 \geqslant \max\left(\left\| \sum_{j=1}^{n} \mathbb{E}\left(\mathbf{Y}_j \mathbf{Y}_j^\top\right) \right\|_{op}, \left\| \sum_{j=1}^{n} \mathbb{E}\left(\mathbf{Y}_j^\top \mathbf{Y}_j\right) \right\|_{op} \right).$$

Then for any $t \in \mathbb{R}^+$ *and* $\nu \in \mathbb{R}^+$*, it holds that,*

$$P\left(\left\| \sum_{j=1}^{n} \widetilde{\psi}_\nu\left(\mathbf{Y}_j\right) - \sum_{j=1}^{n} \mathbb{E}\left(\mathbf{Y}_j\right) \right\|_{op} \geqslant t\sqrt{n} \right) \leqslant 2\left(d_1 + d_2\right) \exp\left(\nu t\sqrt{n} + \frac{\nu^2 \sigma_n^2}{2} \right)$$

Lemma D.3. (***Restatement of Theorem 4.1 in Kang et al. (2022)***) *For any low-rank linear model with samples* $\mathbf{X}_1 \ldots, \mathbf{X}_{n_1}$ *drawn from* $\mathcal{X}$ *according to* $\mathcal{D}$ *then for the optimal solution to the nuclear norm regularization problem in* (5.2)

with $\nu = \sqrt{2\log\left(2\left(d_1+d_2\right)/\delta\right)/\left(\left(4+S_0^2\right)Mn_1d_1d_2\right)}$ *and*

$$\gamma_{n_1} = 4\sqrt{\frac{2\left(4+S_0^2\right)Cd_1d_2\log\left(2\left(d_1+d_2\right)/\delta\right)}{n_1}},$$

with probability at least $1-\delta$ *it holds that:*

$$\left\|\widehat{\Theta}-\mu^*\Theta_*\right\|_F^2 \leqslant \frac{C_1d_1d_2r\log\left(\frac{2(d_1+d_2)}{\delta}\right)}{n_1},$$

for $C_1 = 36\left(4+S_0^2\right)C$, $\|\mathbf{X}\|_F, \|\Theta_*\|_F \leqslant S_0$, *some nonzero constant* μ^*, *and* $\mathbb{E}\left[\left(S^p(\mathbf{X})\right)_{ij}^2\right] \leqslant C, \forall i,j$.

D.2 G-optimal design on rotated arms

Remark D.4. (G-***optimal design on rotated arms:***) *Using the concentration inequality in Proposition 1 we can show that for any arbitrary vector* $\mathbf{y} \in \mathbb{R}^p$*:*

$$|\mathbf{y}^\top(\widehat{\theta}_\ell - \theta_*)| \leqslant \|\mathbf{y}\|_{\mathbf{V}_\ell^{-1}} 2\sqrt{14\log(2/\delta)} + \|\theta_*\|_{\Lambda_\ell} \leqslant \|\mathbf{y}\|_{\mathbf{V}_\ell^{-1}}\sqrt{8B_*^\ell\log(2/\delta)}$$

where the co-variance matrix $\mathbf{V}_\ell \coloneqq \sum_{s=1}^{\tau_\ell^E}\underline{\mathbf{w}}_s\underline{\mathbf{w}}_s^\top + \Lambda_\ell$ *and* B_*^ℓ *is defined in Proposition 1. Now we want this to hold for all* $\mathbf{y} \in \mathcal{Y}^*(\underline{\mathcal{W}}_\ell)$, *and so we need to union bound over* $\mathcal{W} \supseteq \underline{\mathcal{W}}_\ell$ *replacing* δ *with* $\delta/|\mathcal{W}|$. *Set the phase length* $\tau_\ell^G \coloneqq \left\lceil \frac{64B_*^\ell\rho^G(\mathcal{Y}(\mathcal{W}_\ell))\log(4\ell^2|\mathcal{W}|/\delta)}{\epsilon_\ell^2}\right\rceil$ *where* $\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))$ *is defined in step* 14 *of Algorithm 5.*

Then for the allocation $2\lfloor \mathbf{b}_{\underline{\mathbf{w}}}^G\tau_\ell^G\rfloor$ *for each* $\mathbf{b}_{\underline{\mathbf{w}}} \in \underline{\mathcal{W}}_\ell$, *we have for each* $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell\backslash\underline{\mathbf{w}}_*$ *that with probability at least* $1-\delta$,

$$\begin{aligned}&(\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top\widehat{\theta}_\ell \\ &\geqslant (\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top\theta_* - \|\underline{\mathbf{w}}_* - \underline{\mathbf{w}}\|_{\left(\sum_{\underline{\mathbf{w}}\in\mathcal{W}}\lceil 2\tau_\ell^G\mathbf{b}_{\underline{\mathbf{w}}}^*\rceil\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + 2\tau_\ell^G\Lambda_\ell/\tau_\ell^G\right)^{-1}}\sqrt{8B_*^\ell\log(2|\mathcal{W}|/\delta)}\end{aligned}$$

since for every $\underline{\mathbf{w}}_* - \underline{\mathbf{w}} \in \mathcal{Y}^*(\mathcal{W})$ *we have*

$$
\begin{aligned}
&(\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top \big(2 \sum_{\underline{\mathbf{w}} \in \mathcal{W}} \lceil \tau_\ell^G \mathbf{b}^*_{\underline{\mathbf{w}}} \rceil \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + 2\tfrac{\tau_\ell^G \boldsymbol{\Lambda}_\ell}{\tau_\ell^G}\big)^{-1} (\underline{\mathbf{w}}_* - \underline{\mathbf{w}}) \\
&\leqslant \tfrac{1}{\tau_\ell^G} \|\underline{\mathbf{w}}_* - \underline{\mathbf{w}}\|^2_{(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell} \mathbf{b}^*_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \frac{\boldsymbol{\Lambda}_\ell}{\tau_\ell^G})^{-1}} \\
&\leqslant \tfrac{((\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top \theta_*)^2}{\sqrt{8 B_*^\ell \log(2|\mathcal{W}|/\delta)}}.
\end{aligned}
$$

The last inequality follows by plugging in the value of τ_ℓ^G *and* $\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))$*. Hence to minimize the number of samples* τ_ℓ^G *in phase* ℓ *we can re-arrange the above equation to show that*

$$
\tau_\ell^G \geqslant \sqrt{8 B_*^\ell \log(2|\mathcal{W}|/\delta)} \max_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell \setminus \underline{\mathbf{w}}_*} \frac{\|\underline{\mathbf{w}}_* - \underline{\mathbf{w}}\|^2_{(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell} \mathbf{b}^*_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}/n)^{-1}}}{(\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top \theta_*}
$$

Hence, to minimize the sample complexity for the bilinear setting we need to sample according to

$$
\mathbf{b}_\ell^G = \arg\min_{\mathbf{b}} \max_{\underline{\mathbf{w}}} \frac{\|\underline{\mathbf{w}}_* - \underline{\mathbf{w}}\|^2_{(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}/n)^{-1}}}{(\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top \theta_*} \tag{D.1}
$$

However, note that we do know the identity of $\mathbf{w}_*$ *or the gaps* $(\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top \theta_*$*. So we replace the gaps with a lower bound of* $\epsilon = 2^{-t}$ *and compare against every pair of arms* $\underline{\mathbf{w}}$ *and* $\underline{\mathbf{w}}'$ *as follows:*

$$
\mathbf{b}_\ell^G = \arg\min_{\mathbf{b}_{\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}}, \underline{\mathbf{w}}' \in \underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell/n)^{-1}} \tag{D.2}
$$

This is shown in step 12 *of Algorithm 5.*

D.3 Application of Stein's Lemma

We also present the following two definitions from Kang et al. (2022) to facilitate analysis via Stein's method:

Definition D.5. (***Score Function***) *Let* $\mathbf{p} : \mathbb{R} \to \mathbb{R}$ *be a univariate probability density function defined on* $\mathbb{R}$*. The score function* $Q^{\mathbf{p}} : \mathbb{R} \to \mathbb{R}$ *regarding density* $\mathbf{p}(\cdot)$ *is defined as:*

$$Q^{\mathbf{p}}(x) = -\nabla_x \log(\mathbf{p}(x)) = -\nabla_x \mathbf{p}(x)/\mathbf{p}(x), \quad x \in \mathbb{R}.$$

In particular, for a random matrix with its entrywise probability density $\mathbf{p} = (p_{ij}) : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_1 \times d_2}$, we define its score function $Q^{\mathbf{p}} = \left(Q^{\mathbf{p}}_{ij}\right) : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_1 \times d_2}$ as $Q^{\mathbf{p}}_{ij}(x) = Q^{p_{ij}}(x)$ by applying the univariate score function to each entry of $\mathbf{p}$ independently.

Assumption 11. *The norm of true parameter* $\mathbf{\Theta}^*$ *and feature matrices in* $\mathcal{X}$ *is bounded: there exists* $S \in \mathbb{R}^+$*such that for all arms* $\mathbf{X} \in \mathcal{X}$*,* $\|\mathbf{X}\|_F, \|\mathbf{\Theta}^*\|_F \leqslant S_0$

Assumption 12. (***Finite second-moment score***) *There exists a sampling distribution* $\mathcal{D}$ *over* $\mathcal{X}$ *such that for the random matrix* $\mathbf{X}$ *drawn from* $\mathcal{D}$ *with its associated density* $\mathbf{p} : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_1 \times d_2}$*, we have* $\mathbb{E}\left[(Q^{\mathbf{p}}(\mathbf{X}))^2_{ij}\right] \leqslant C, \forall i, j$

Definition D.6. *Given a rectangular matrix* $\mathbf{A} \in \mathbb{R}^{d_1 \times d_2}$*, the (Hermitian) dilation* $\mathcal{H} : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$ *is defined as:*

$$\mathcal{H}(\mathbf{A}) = \begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{A}^\top & 0 \end{pmatrix}$$

Definition D.7. (***The function*** $\widetilde{\psi}_\nu$) *To explore the valid subspace of the parameter matrix* $\mathbf{\Theta}_*$*, we define a function* $\psi : \mathbb{R} \to \mathbb{R}$ *in* (D.3)*. Let* $\mathcal{H}(\cdot)$ *be as defined in Theorem D.6. Then define* $\widetilde{\psi}_\nu : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}^{d_1 \times d_2}$ *as* $\widetilde{\psi}_\nu(\mathbf{A}) =$

$\psi(\nu\mathcal{H}(\mathbf{A}))_{1:d_1,(d_1+1):(d_1+d_2)}/\nu$ *for some parameter* $\nu \in \mathbb{R}^+$

$$\psi(x) = \begin{cases} \log\left(1 + x + x^2/2\right), & x \geqslant 0 \\ -\log\left(1 - x + x^2/2\right), & x < 0 \end{cases} \tag{D.3}$$

D.4 Single-task Pure Exploration Proofs

Good Event: Define the good event $\mathcal{F}_\ell$ in phase ℓ that GOBLIN has a good estimate of $\boldsymbol{\Theta}_*$ as follows:

$$\mathcal{F}_\ell = \left\| \widehat{\boldsymbol{\Theta}}_\ell - \mu^* \boldsymbol{\Theta}_* \right\|_F^2 \leqslant \frac{C_1 d_1 d_2 r \log\left(\frac{2(d_1+d_2)}{\delta_\ell}\right)}{\tau_\ell^E}, \tag{D.4}$$

where, $C_1 = 36\left(4 + S_0^2\right) C$, $\|\mathbf{X}\|_F, \|\boldsymbol{\Theta}_*\|_F \leqslant S_0$, some nonzero constant μ^*, $\mathbb{E}\left[(S^p(\mathbf{X}))_{ij}^2\right] \leqslant C, \forall i, j$, and $\widehat{\boldsymbol{\Theta}}_\ell$ is the estimate from (5.2). Then define the good event

$$\mathcal{F} \coloneqq \bigcap_{\ell=1}^{\infty} \mathcal{F}_\ell. \tag{D.5}$$

Lemma D.8. *The event* $\mathcal{F}$ *holds with probability greater than* $(1 - \delta/2)$.

Proof. From Theorem D.3 we know the event $\mathcal{F}_\ell$ in (D.4) holds with probability $(1 - \delta_\ell)$. Taking a union bound over all phases $\ell \geqslant 1$ and recalling

$\delta_\ell := \frac{\delta}{2\ell^2}$, we obtain

$$\begin{aligned}\mathbb{P}(\mathcal{F}) &\geqslant 1 - \sum_{\ell=1}^{\infty} \mathbb{P}\left(\mathcal{F}_\ell^c\right) \\ &\geqslant 1 - \sum_{\ell=1}^{\infty} \frac{\delta_\ell}{2} \\ &= 1 - \sum_{\ell=1}^{\infty} \frac{\delta}{4\ell^2} \\ &\geqslant 1 - \frac{\delta}{2}.\end{aligned}$$

This concludes our proof. □

Now we move to the second stage for the rotated arm set $\underline{\mathbf{w}} \in \underline{\mathcal{W}}$ and prove the following concentration event.

Lemma D.9. *For any fixed $\underline{\mathbf{w}} \in \mathbb{R}^p$ and any $\delta > 0$, we have that if $\beta(\theta_*, \delta) = 2\sqrt{14\log(2/\delta)} + \|\theta_*\|_{\mathbf{\Lambda}}$, then at time $\tau_{\ell-1} + 1$ (beginning of phase ℓ):*

$$\mathbb{P}\left(\left|\underline{\mathbf{w}}^\top\left(\widehat{\theta}_\ell - \theta_*\right)\right| \leqslant \|\underline{\mathbf{w}}\|_{\mathbf{V}_\ell^{-1}} \beta(\theta_*, \delta)\right) \geqslant 1 - \delta$$

where, $\mathbf{V}_\ell := \sum_{s=\tau_{\ell-1}+1}^{\tau_\ell} \underline{\mathbf{w}}_s \underline{\mathbf{w}}_s^\top + \mathbf{\Lambda}_\ell$.

Proof. We follow the proof technique of Lemma 7 of Valko et al. (2014). Defining $\text{ع}_\ell = \sum_{s=\tau_{\ell-1}+1}^{\tau_\ell} \underline{\mathbf{w}}_s \eta_s$, we have:

$$\begin{aligned}\left|\underline{\mathbf{w}}^\top\left(\widehat{\theta}_\ell - \theta_*\right)\right| &\overset{(a)}{=} \left|\underline{\mathbf{w}}^\top\left(-\mathbf{V}_\ell^{-1}\mathbf{\Lambda}\theta_* + \mathbf{V}_\ell^{-1}\text{ع}_\ell\right)\right| \\ &\overset{(b)}{\leqslant} \left|\underline{\mathbf{w}}^\top\mathbf{V}_\ell^{-1}\mathbf{\Lambda}_\ell\theta_*\right| + \left|\underline{\mathbf{w}}^\top\mathbf{V}_\ell^{-1}\text{ع}_\ell\right| \qquad \text{(D.6)}\end{aligned}$$

where (a) follows from Woodbury matrix identity and rearranging the terms, and (b) follows from the triangle inequality.

The first term in the right-hand side of (D.6) is bounded as:

$$\begin{aligned}\left|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\Lambda}_\ell \theta_*\right| &\leqslant \left\|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\Lambda}_\ell^{1/2}\right\| \left\|\boldsymbol{\Lambda}_\ell^{1/2} \theta_*\right\| \\ &\overset{(a)}{=} \|\theta_*\|_{\boldsymbol{\Lambda}_\ell} \sqrt{\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\Lambda}_\ell \mathbf{V}_\ell^{-1} \underline{\mathbf{w}}} \\ &\leqslant \|\theta_*\|_{\boldsymbol{\Lambda}_\ell} \sqrt{\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \underline{\mathbf{w}}} = \|\theta_*\|_{\boldsymbol{\Lambda}_\ell} \|\underline{\mathbf{w}}\|_{\mathbf{V}_\ell^{-1}}\end{aligned}$$

where, (a) follows as $\|\theta_*\|_{\boldsymbol{\Lambda}_\ell} = \sqrt{\theta_*^\top \boldsymbol{\Lambda}_\ell \theta_*} = \|\boldsymbol{\Lambda}_\ell^{1/2} \theta_*\|$ and similarly for $\left\|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\Lambda}_\ell^{1/2}\right\|$. Now consider the second term in the r.h.s. of (D.6). We have:

$$\left|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\zeta}_\ell\right| = \left|\sum_{s=\tau_{\ell-1}+1}^{\tau_\ell} \left(\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \underline{\mathbf{w}}_s\right) \eta_s\right|.$$

Now note that the arms $(\underline{\mathbf{w}}_s)$ selected by the algorithm during phase ℓ only depend on the proportion $\mathbf{b}_*^G$ (the G-optimal design) and do not depend on the rewards received during the phase $\ell - 1$. Thus, given $\mathcal{F}_{j-2}$, the sequence $(\underline{\mathbf{w}}_s)_{\tau_{\ell-1}+1 \leqslant s < \tau_\ell}$ is deterministic. Consequently, one may use a variant of Azuma's inequality (Shamir (2011)) with a 1-sub Gaussian assumption:

$$\begin{aligned}\mathbb{P}\left(\left|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\zeta}_\ell\right|^2 \leqslant 28 \times 2\log(2/\delta) \times \underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \left(\sum_{s=\tau_{\ell-1}+1}^{\tau_\ell} \underline{\mathbf{w}}_s \underline{\mathbf{w}}_s^\top\right) \mathbf{V}_\ell^{-1} \underline{\mathbf{w}} \mid \mathcal{F}_{\ell-2}\right) \\ \geqslant 1 - \delta,\end{aligned}$$

from which we deduce:

$$\mathbb{P}\left(\left|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \boldsymbol{\zeta}_\ell\right|^2 \leqslant 56 \underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \underline{\mathbf{w}} \log(2/\delta) \mid \mathcal{F}_{\ell-2}\right) \geqslant 1 - \delta,$$

since $\sum_{s=\tau_{\ell-1}+1}^{\tau_\ell} \underline{\mathbf{w}}_s \underline{\mathbf{w}}_s^\top \prec \mathbf{V}_\ell$. Thus:

$$\mathbb{P}\left(\left|\underline{\mathbf{w}}^\top \mathbf{V}_\ell^{-1} \underline{\mathbf{z}}_\ell\right| \leqslant 2\|\underline{\mathbf{w}}\|_{\mathbf{V}_\ell^{-1}} \sqrt{14\log(2/\delta)}\right) \geqslant 1-\delta$$

Combining everything we get that

$$\mathbb{P}\left(\left|\underline{\mathbf{w}}^\top \left(\widehat{\theta}_\ell - \theta_*\right)\right| \leqslant 2\sqrt{14\log(2/\delta)} + \|\theta_*\|_{\mathbf{\Lambda}_\ell}\right) \leqslant 1-\delta.$$

□

We need to change Lemma 6 of Valko et al. (2014) in the following way so that the dependence on horizon n is replaced by $\tau_{\ell-1}^G$. Note that $\tau_{\ell-1}^G$ is the phase length in the $\ell-1$-th phase and is determined before the start of phase $\tau_{\ell-1}^G$. Also, note that using the standard analysis of phase-based algorithms in Fiez et al. (2019); Lattimore and Szepesvári (2020a) we do not re-use data between phases. First, we need the following support lemma from Valko et al. (2014).

Lemma D.10. (***Restatement of Lemma 4 from Valko et al. (2014)***) *Let $\mathbf{\Lambda}_\ell = \operatorname{diag}\left(\lambda_1, \ldots, \lambda_\ell^\perp\right)$ be any diagonal matrix with strictly positive entries. Define $\mathbf{V}_\ell = \mathbf{\Lambda}_\ell + \sum_{s=\tau_{\ell-1}^G+1}^{\tau_\ell^G} \underline{\mathbf{w}}_s \underline{\mathbf{w}}_s^\top$. Then for any vectors $(\underline{\mathbf{w}}_s)_{\tau_{\ell-1}^G+1 \leqslant s \leqslant \tau_\ell^G}$, such that $\|\underline{\mathbf{w}}_s\|_2 \leqslant 1$ for all rounds s such that $\tau_{\ell-1}^G + 1 \leqslant s \leqslant \tau_\ell^G$, we have that the determinant $|\mathbf{V}_\ell|$ is maximized when all $\underline{\mathbf{w}}_s$ are aligned with the axes.*

Lemma D.11. *Let k be the effective dimension. Then*

$$\log \frac{|\mathbf{V}_\ell|}{|\mathbf{\Lambda}_\ell|} \leqslant 8k \log\left(1 + \frac{\tau_{\ell-1}^G}{\lambda}\right)$$

when $\lambda_\ell^\perp = \frac{\tau_{\ell-1}^G}{k\log\left(1+\tau_{\ell-1}^G/\lambda\right)}$.

Proof. We want to bound the determinant $|\mathbf{V}_\ell|$ under the coordinate constraints $\|\underline{\mathbf{w}}_t\|_2 \leqslant 1$. Let:

$$M\left(\underline{\mathbf{w}}_1, \ldots, \underline{\mathbf{w}}_t\right) = \left|\boldsymbol{\Lambda}_\ell + \sum_{s=\tau_{\ell-1}^G+1}^{\tau_\ell^G} \underline{\mathbf{w}}_s \underline{\mathbf{w}}_s^\top\right|$$

From Theorem D.10 we deduce that the maximum of M is reached when all $\underline{\mathbf{w}}_s$ are aligned with the axes. Let the number of samples of these axes-aligned $\underline{\mathbf{w}}_s$'s during the ℓ-th phase be denoted as $t_1^\ell, t_2^\ell, \ldots, t_p^\ell$ such that $\sum_{i=1}^p t_i^\ell = \tau_\ell^G$. Then we can show that

$$\begin{aligned} M &\overset{(a)}{=} \max_{\underline{\mathbf{w}}_1,\ldots,\underline{\mathbf{w}}_t;\underline{\mathbf{w}}_s\in\{e_1,\ldots,\mathbf{e}_p\}} \left|\boldsymbol{\Lambda}_\ell + \sum_{s=\tau_{\ell-1}^G+1}^{\tau_\ell^G} \underline{\mathbf{w}}_s \underline{\mathbf{w}}_s^\top\right| \\ &\overset{(b)}{=} \max_{t_1^\ell,\ldots,t_p^\ell, \text{ positive integers, } \sum_{i=1}^p t_i^\ell = \tau_\ell^G} \left|\operatorname{diag}\left(\lambda_i + t_i^\ell\right)\right| \\ &\overset{(c)}{\leqslant} \max_{t_1^\ell,\ldots,t_p^\ell, \text{ positive integers, } \sum_{i=1}^p t_i^\ell = \tau_\ell^G} \prod_{i=1}^p \left(\lambda_i + t_i^\ell\right) \end{aligned}$$

where, (a) follows from Theorem D.10, (b) follows as the $\operatorname{diag}\left(\lambda_i + t_i^\ell\right)$ contains the number of times axis-aligned $\underline{\mathbf{w}}_s \in \{e_1, \ldots, \mathbf{e}_p\}$ are observed, and (c) follows as the determinant of a diagonal matrix is the product of

the diagonal elements. Now we can show that

$$\begin{aligned}
\log \frac{|\mathbf{V}_\ell|}{|\mathbf{\Lambda}_\ell|} &\leqslant \sum_{i=1}^{k} \log\left(1+\frac{t_i^\ell}{\lambda}\right) + \sum_{i=k+1}^{p} \log\left(1+\frac{t_i^\ell}{\lambda_i}\right) \\
&\overset{(a)}{\leqslant} k\log\left(1+\frac{t_i^\ell}{\lambda}\right) + \sum_{i=1}^{p} \frac{t_i^{\ell-1}}{\lambda_\ell^\perp} \\
&\overset{(b)}{\leqslant} k\log\left(1+\frac{t_i^\ell}{\lambda}\right) + \frac{\tau_{\ell-1}^G}{\lambda_\ell^\perp} \\
&\overset{(c)}{\leqslant} k\log\left(1+\frac{t_i^\ell}{\lambda}\right) + k\log\left(1+\frac{\tau_{\ell-1}^G}{\lambda}\right) \\
&\overset{(d)}{\leqslant} 8k\log\left(1+\frac{\tau_{\ell-1}^G}{\lambda}\right)
\end{aligned}$$

where, (a) follows as $\log(1+t_i^\ell/\lambda_i) \leqslant t_i^{\ell-1}/\lambda_\ell^\perp$, (b) follows as $\sum_{i=1}^p t_i^{\ell-1} = \tau_{\ell-1}^G$, and (c) follows for $\lambda_\ell^\perp = \frac{\tau_{\ell-1}^G}{k\log\left(1+\tau_{\ell-1}^G/\lambda\right)}$ and (d) follows from Theorem D.12. □

Lemma D.12. *Let* $\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) = \min_{\mathbf{b}_{\underline{\mathbf{w}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}}-\underline{\mathbf{w}}'\|^2_{(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \mathbf{\Lambda}_\ell/n)^{-1}}$. *Recall that* $\tau_\ell^G = \frac{8B_*^\ell \rho^G(\mathcal{Y}(\mathcal{W}_\ell))\log(4\ell^2|\mathcal{W}|/\delta)}{\epsilon_\ell^2}$. *Assume* $\log(p) \leqslant k$. *Then we can show that*

$$\log\left(1+\frac{\tau_\ell^G}{\lambda}\right) \leqslant 8k\log\left(1+\frac{\tau_{\ell-1}^G}{\lambda}\right).$$

Proof. We start by first recalling the definition of

$$\tau_\ell^G = 2\epsilon_\ell^{-2} 8k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))\log\left(4\ell^2|\underline{\mathcal{W}}|/\delta_\ell\right).$$

Then we can show the following

$$
\begin{aligned}
\frac{\tau_\ell^G}{\tau_{\ell-1}^G} &= \frac{2\epsilon_\ell^{-2} B_*^\ell \rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \log\left(4\ell^2|\mathcal{W}|/\delta_\ell\right)}{2\epsilon_{\ell-1}^{-2} B_*^{\ell-1} \rho^G(\mathcal{Y}(\underline{\mathcal{W}}_{\ell-1})) \log\left(4(\ell-1)^2|\mathcal{W}|/\delta_\ell\right)} \\
&\leqslant \frac{4\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \left(64(\lambda S^2 + \lambda_\ell^\perp S_\perp^{(2),\ell})\right)}{\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_{\ell-1})) \left(64(\lambda S^2 + \lambda_{\ell-1}^\perp S_\perp^{(2),\ell-1})\right)} \\
&\leqslant \frac{4\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))\tau_{\ell-1}^G / \log(1+\frac{\tau_{\ell-1}^G}{\lambda})}{\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_{\ell-1}))\tau_{\ell-2}^G / \log(1+\frac{\tau_{\ell-2}^G}{\lambda})} \\
&\overset{(a)}{\leqslant} \frac{4\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell))\tau_{\ell-1}^G \log(1+\frac{\tau_{\ell-1}^G}{\lambda})}{\rho^G(\mathcal{Y}(\underline{\mathcal{W}}_{\ell-1}))} \\
&\overset{(b)}{\leqslant} \frac{4p}{\gamma_{\mathcal{Y}}^2} \frac{\max_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \|\underline{\mathbf{w}}\|_2}{\max_{\mathbf{y}\in\mathcal{Y}(\underline{\mathcal{W}}_\ell)} \|\mathbf{y}\|_2^2} \tau_{\ell-1}^G \log(1+\frac{\tau_{\ell-1}^G}{\lambda}) = \frac{4p}{C\gamma_{\mathcal{Y}}^2}
\end{aligned}
$$

where, (a) follows as $\log(1+\frac{\tau_{\ell-2}^G}{\lambda}) \geqslant 1$ and $\log(1+\frac{\tau_{\ell-1}^G}{\lambda}) \geqslant \log(1+\frac{\tau_{\ell-2}^G}{\lambda})$. The (b) follows using Lemma 1 from Fiez et al. (2019) such that

$$
\max_{\mathbf{y}\in\mathcal{Y}(\underline{\mathcal{W}}_\ell)} \|\mathbf{y}\|_2^2 / \left(\max_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \|\underline{\mathbf{w}}\|_2\right) \leqslant \rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \leqslant p/\gamma_{\mathcal{Y}}^2 \overset{(a_1)}{\implies} 1 \leqslant \rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \leqslant p/\gamma_{\mathcal{Y}}^2.
$$

where, (a_1) follows as $\|\mathbf{x}\| \leqslant 1$, $\|\mathbf{z}\| \leqslant 1$. This implies that for a constant $C > 0$

$$
\begin{aligned}
\tau_\ell^G &\leqslant \frac{4p}{C\gamma_{\mathcal{Y}}^2} (\tau_{\ell-1}^G)^2 \log(1+\frac{\tau_{\ell-1}^G}{\lambda}) \\
&\implies \log\left(1+\frac{\tau_\ell^G}{\lambda}\right) \leqslant \log\left(1+\frac{4p}{C\gamma_{\mathcal{Y}}^2}(\tau_{\ell-1}^G)^2 \log(1+\frac{\tau_{\ell-1}^G}{\lambda})\right) \\
&\overset{(a)}{\implies} \log\left(1+\frac{\tau_\ell^G}{\lambda}\right) \leqslant 4k \log\left(1+\frac{\tau_{\ell-1}^G}{C\gamma_{\mathcal{Y}}^2\lambda}\right) \\
&\overset{(b)}{\implies} \log\left(1+\frac{\tau_\ell^G}{\lambda}\right) \leqslant 8k \log\left(1+\frac{\tau_{\ell-1}^G}{\lambda}\right)
\end{aligned}
$$

where, in (a) follows for $\log(p) \leqslant k$, $\log(a^2 \log(a)) \leqslant 4\log(a)$. The (b)

follows as $4k\log\left(1+\frac{\tau^G_{\ell-1}}{C\gamma_y^2\lambda}\right)\leqslant 8k\log\left(1+\frac{\tau^G_{\ell-1}}{\lambda}\right)$. □

Lemma D.13. *The* G*-optimal design in* (5.5) *is equivalent to solving the* D*-optimal design*

$$\mathbf{b}^D_\ell = \arg\max_{\mathbf{b}} \log \frac{\left|\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right|}{|\boldsymbol{\Lambda}_\ell|}.$$

Furthermore, the support of $|\mathbf{b}^D_\ell| \leqslant \frac{8k\log(1+\frac{\tau^G_{\ell-1}}{\lambda})(8k\log(1+\frac{\tau^G_{\ell-1}}{\lambda})+1)}{2}$, *where* $k = (d_1+d_2)r$.

Proof. To prove the equivalence between $\mathbf{b}^G_*$ and $\mathbf{b}^D_*$ we need to first show that the regularization matrix $\boldsymbol{\Lambda}_\ell$ does not depend on $\underline{\mathbf{w}}$ or $\mathbf{y} = \underline{\mathbf{w}} - \underline{\mathbf{w}}'$, where $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell$. Define $\mathrm{conv}(\mathcal{X}\cup-\mathcal{X})$ as the convex hull of $\mathcal{X}\cup-\mathcal{X}$. Now recall we have from Lemma 1 of Fiez et al. (2019) that

$$1 \leqslant \rho^G(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \leqslant p/\gamma_y^2 \tag{D.7}$$

where, $\gamma_{\mathcal{Y}} = \max\{c>0 : c\mathcal{Y}\subset \mathrm{conv}(\underline{\mathcal{W}}_\ell\cup-\underline{\mathcal{W}}_\ell)\}$ as the gauge norm of $\mathcal{Y}$ (Rockafellar, 2015). We can consider the gauge norm $\gamma_{\mathcal{Y}}$ as a problem-dependent constant. Now recall that

$$\begin{aligned}\lambda^\ell_\perp = \frac{\tau^G_{\ell-1}}{8k\log(1+\frac{\tau^G_{\ell-1}}{\lambda})} \leqslant 2\tau^G_{\ell-1} &\leqslant \frac{64B_*^{\ell-1}\rho^G(\mathcal{Y}(\mathcal{W}_{\ell-1}))\log(4(\ell-1)^2|\mathcal{W}|/\delta_\ell)}{\epsilon^2_{\ell-1}}\\ &\overset{(a)}{\leqslant} \frac{64(B_*^{\ell-1})^2 p\log(4(\ell-1)^2|\mathcal{W}|/\delta_\ell)}{\gamma_y^2\epsilon^2_{\ell-1}}\\ &\overset{(b)}{\leqslant} \frac{\left(256(\lambda S^2+8p^2r)\right)\log(4(\ell-1)^2 p|\mathcal{W}|/\delta_\ell)}{S_r\gamma_y^2\epsilon^2_{\ell-1}}\end{aligned}$$

where, (a) follows from (D.7) and noting that $B_*^{\ell-1}\leqslant (B_*^{\ell-1})^2$. The (b) follows as $S^\perp_\ell \coloneqq \frac{8pr}{\tau^E_\ell S^2_\tau}\log\left(\frac{d_1+d_2}{\delta_\ell}\right)$, $p = d_1d_2$ and substituting this value

and $\lambda_{\perp}^{\ell}$ in B_*^{ℓ} we get

$$
\begin{aligned}
B_*^{\ell} &\leqslant (B_*^{\ell})^2 \leqslant \left(256(\lambda S^2 + \lambda_{\perp}^{\ell} S_{\perp}^{(2),\ell})\right) \\
&\leqslant \left(256\left(\lambda S^2 + \frac{8pr}{S_r^2}\frac{\tau_{\ell-1}^G}{\tau_{\ell}^E} \cdot \log(4(\ell-1)^2|\mathcal{W}|/\delta_\ell)\log(d_1 + d_2/\delta_\ell)\right)\right) \\
&\leqslant \left(256\left(\lambda S^2 + \frac{8pr}{S_r^2}\rho_{\ell-1}^G(\underline{\mathcal{W}})\log(p|\mathcal{W}|/\delta_\ell)\right)\right) \overset{(a)}{\leqslant} \left(512\left(\lambda S^2 + 8p^2 r\right)/S_r\right).
\end{aligned}
$$

Here S_r is the r-th larget eigenvalue of matrix $\boldsymbol{\Theta}_*$. Substituting this value of $\lambda_{\ell}^{\perp}$ we can show that $\boldsymbol{\Lambda}_\ell$ does not depend on $\underline{\mathbf{w}}$ or $\mathbf{y} = \underline{\mathbf{w}} - \underline{\mathbf{w}}'$. The rest of the proof to show equivalence follows the same way as in Theorem 21.1 in Lattimore and Szepesvári (2020a).

To bound the support of $\mathbf{b}_*^D$ we proceed as follows: Define the set $\mathcal{Y}(\underline{\mathcal{W}}_\ell)$ as the set of all arms containing $\mathbf{y} = \underline{\mathbf{w}} - \underline{\mathbf{w}}' \in \mathbb{R}^p$. Then we can use Lemma 7 of Soare et al. (2014) to show that the solution to

$$
\max_{\mathbf{y}\in\mathcal{Y}(\underline{\mathcal{W}}_\ell)} \|\mathbf{y}\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right)^{-1}} = \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right)^{-1}}
$$

has a support of atmost $(k_1+1)k_1/2$ where $k_1 = 8k\log(1 + \tau_{\ell-1}^G/\lambda)$. The proof follows from the fact that for any pair $(\underline{\mathbf{w}}, \underline{\mathbf{w}}')$ we can show that

$$
\|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \leqslant 2\max_{\underline{\mathbf{w}}''\in\underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}}''\|_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right)^{-1}}.
$$

Then following the work of Jamieson and Jain (2022) Frank-Wolfe algorithm (in section 2.3.1) with the

$$
g(\mathbf{b}) = \log\frac{\left|\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right|}{|\boldsymbol{\Lambda}_\ell|} = \log\left|\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right| - \log|\boldsymbol{\Lambda}_\ell|,
$$

and setting for the j-th iteration of the Frank-Wolfe the

$$\mathrm{I}_{\mathrm{j}} = \underset{\mathbf{y}\in\mathcal{Y}(\underline{\mathcal{W}}_\ell)}{\arg\max} \|\mathbf{y}\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}^{\mathrm{j}}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right)^{-1}},$$

and stopping condition

$$\max_{\mathbf{y}\in\mathcal{Y}(\underline{\mathcal{W}}_\ell)} \|\mathbf{y}\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}^{\mathrm{j}}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right)^{-1}} \leqslant 8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda}) \tag{D.8}$$

This can be done because note that for any $\mathbf{b} \in \triangle_{\underline{\mathcal{W}}_\ell}$ we have by Kiefer-Wolfowitz Theorem (Kiefer and Wolfowitz, 1960) that $[\nabla g(\mathbf{b})]_{\mathbf{y}} = \|\mathbf{y}\|^2_{\left(\sum_{\mathbf{x}\in\mathcal{X}} \mathbf{b}_{\mathbf{x}}\mathbf{x}\mathbf{x}^\top + \boldsymbol{\Lambda}\right)^{-1}} \geqslant 8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})$. This is because $\boldsymbol{\Lambda}_\ell$ does not depend on $\underline{\mathbf{w}}$ or $\mathbf{y}$ by the same logic as discussed before. The rest of the proof follows by the same way as in section 2.3.1 in Jamieson and Jain (2022). This will result in a support size of $\mathbf{b}$ at most $8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})(8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})+1)/2$ following Lemma 7 of Soare et al. (2014). Hence, it follows that solving the eq. (5.5) will result in a support of $|\mathbf{b}^{\mathrm{D}}_\ell| \leqslant \frac{8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})(8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})+1)}{2}$. □

Proposition 1. *If $\mathbf{b}^{\mathrm{G}}_\ell$ is the G-optimal design for $\underline{\mathcal{W}}_\ell$ then if we pull arm $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell$ exactly $\lceil \tau^{\mathrm{G}}\mathbf{b}^{\mathrm{G}}_\ell \rceil$ times for some $\tau^{\mathrm{G}}_\ell > 0$ and compute the least squares estimator $\widehat{\theta}_\ell$. Then for each $\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell$ we have with probability at least $1-\delta$*

$$\mathbb{P}\Bigg(\bigcup_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\left\{\left|\left\langle \underline{\mathbf{w}}, \widehat{\theta}_\ell - \theta^*\right\rangle\right| \leqslant \sqrt{\frac{64\mathrm{B}^\ell_*{}^2 k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})\log\left(2|\mathcal{W}|/\delta_\ell\right)}{\tau^{\mathrm{G}}_\ell}}\right\}\Bigg) \geqslant 1-\delta_\ell.$$

where, $\|\theta^*\|_\Lambda \leqslant \sqrt{\lambda\|\theta_{1:k}\|_2^2 + \lambda^\perp_\ell\|\theta_{k+1:p}\|_2^2} \leqslant \sqrt{\lambda}S + \sqrt{\lambda^\perp_\ell}S^\ell_\perp$, $\mathrm{B}^\ell_* = \sqrt{\lambda}S + \sqrt{\lambda^\perp_\ell}S^\perp_\ell$.

Proof. From Woodbury Matrix Identity we know that for any arbitrary matrix $\mathbf{A}$ and $\mathbf{B}$, we have the following identity $(\mathbf{A}+\mathbf{B})^{-1} = \mathbf{A}^{-1} -$

$\left(\mathbf{A}+\mathbf{A}\mathbf{B}^{-1}\mathbf{A}\right)^{-1}$. It follows then that

$$\underline{\mathbf{w}}^\top(\mathbf{A}+\mathbf{B})^{-1}\underline{\mathbf{w}} = \underline{\mathbf{w}}^\top\left(\mathbf{A}^{-1}-\left(\mathbf{A}+\mathbf{A}\mathbf{B}^{-1}\mathbf{A}\right)^{-1}\right)\underline{\mathbf{w}} \leqslant \underline{\mathbf{w}}^\top\mathbf{A}^{-1}\underline{\mathbf{w}} = \|\underline{\mathbf{w}}\|_{\mathbf{A}^{-1}}.$$

Hence we can show that,

$$\|\underline{\mathbf{w}}\|_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\lceil\tau_\ell^G\mathbf{b}_\ell^G\rceil\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}_\ell\right)^{-1}} \leqslant \underline{\mathbf{w}}^\top\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\lceil\tau_\ell^G\mathbf{b}_\ell^G\rceil\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top\right)^{-1}\underline{\mathbf{w}}. \tag{D.9}$$

From Theorem D.13 we know the support of $\mathbf{b}_\ell^G$ is less than

$$\frac{8k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})(8k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})+1)}{2} \leqslant (8k\log(1+\frac{\tau_{\ell-1}^G}{\lambda}))^2.$$

Also note that $\|\theta^*\|_{\boldsymbol{\Lambda}_\ell} \leqslant \sqrt{\lambda\|\theta_{1:k}\|_2^2+\lambda_\perp^\ell\|\theta_{k+1:p}\|_2^2} \leqslant \sqrt{\lambda}S+\sqrt{\lambda_\perp^\ell}S_\perp^\ell$. Then we can show that

$$\begin{aligned}
&\left\langle\underline{\mathbf{w}},\widehat{\theta}_\ell-\theta^*\right\rangle\\
&\leqslant \|\underline{\mathbf{w}}\|_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\lceil\tau_\ell^G\mathbf{b}_\ell^G\rceil\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}}\left(2\sqrt{14\log(2/\delta_\ell)}+\sqrt{\lambda}S+\sqrt{\lambda_\perp^\ell}S_\perp^\ell\right)\\
&\leqslant \frac{1}{\sqrt{\tau_\ell^G}}\|\underline{\mathbf{w}}\|_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\mathbf{b}_{\ell^*}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}}\left(2\sqrt{14\log(2/\delta_\ell)}+\sqrt{\lambda}S+\sqrt{\lambda_\perp^\ell}S_\perp^\ell\right)\\
&\overset{(a)}{\leqslant} \sqrt{\frac{56\times 8k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})\log(2/\delta_\ell)}{\tau_\ell^G}}+\sqrt{\frac{28k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})\log(2/\delta_\ell)}{\tau_\ell^G}}(\sqrt{\lambda}S+\sqrt{\lambda_\perp^\ell}S_\perp^\ell)\\
&= \sqrt{\frac{8k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})\log(2/\delta_\ell)}{\tau_\ell^G}}\left(\sqrt{56}+\underbrace{\sqrt{\lambda}S+\sqrt{\lambda_\perp^\ell}S_\perp^\ell}_{B_*^\ell}\right)\\
&\leqslant \sqrt{\frac{64B_*^\ell k\log(1+\frac{\tau_{\ell-1}^G}{\lambda})\log(2/\delta_\ell)}{\tau_\ell^G}}
\end{aligned}$$

where, (a) follows as $\|\underline{\mathbf{w}}\|_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell} \mathbf{b}^{\mathrm{G}}_{\ell,\underline{\mathbf{w}}} \underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}_\ell\right)^{-1}} \leqslant (8k\log(1+\tau^{\mathrm{G}}_{\ell-1}/\lambda))^2$. Thus we have taken at most $\tau^{\mathrm{G}}_\ell + \frac{8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})(8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})+1)}{2}$ pulls. Thus, for any $\delta \in (0,1)$ we have

$$\mathbb{P}\Bigg(\bigcup_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\left\{\left|\left\langle \underline{\mathbf{w}}, \widehat{\theta}_\ell - \theta^*\right\rangle\right| \geqslant \sqrt{\frac{64kB_*^\ell \log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda})\log\left(2|\mathcal{W}|/\delta_\ell\right)}{\tau^{\mathrm{G}}_\ell}}\right\}\Bigg) \leqslant \delta_\ell.$$

The claim of the lemma follows. □

Discussion 3. (Phase Length) It follows from Proposition 1 that if the gaps are known, one can set the phase length as

$$\tau^{\mathrm{G}}_\ell = \frac{64B_*^\ell \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell))\log\left(2|\mathcal{W}|/\delta\right)}{(\underline{\mathbf{w}}^\top(\widehat{\theta}_\ell - \theta^*))^2}$$

since $8k\log(1+\frac{\tau^{\mathrm{G}}_{\ell-1}}{\lambda}) \leqslant \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \coloneqq 8k\log(1+\frac{\tau^{\mathrm{G}}_\ell}{\lambda})$ to guarantee that the event $\bigcup_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\left\{\left|\left\langle \underline{\mathbf{w}}, \widehat{\theta}_\ell - \theta^*\right\rangle\right|$ holds with probability greater than $1-\delta$.

However, since in practice, the gaps are not known, for an agnostic algorithm that does not know the gaps, one can set a proxy for the gap as ϵ_ℓ (for some $\epsilon_\ell > 0$) and get the phase length as follows:

$$\tau^{\mathrm{G}}_\ell = \frac{64B_*^\ell \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell))\log\left(2|\mathcal{W}|/\delta\right)}{\epsilon_\ell^2}.$$

This gives us the desired phase length so that the event $\bigcup_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\left\{\left|\left\langle \underline{\mathbf{w}}, \widehat{\theta}_\ell - \theta^*\right\rangle\right|$ holds with probability greater than $1-\delta$.

Lemma D.14. *Assume that* $\max_{\underline{\mathbf{w}}\in\mathcal{W}}\langle \underline{\mathbf{w}}_* - \underline{\mathbf{w}}, \theta^*\rangle \leqslant 2$. *With probability at least* $1-\delta$, *we have* $\underline{\mathbf{w}}_* \in \underline{\mathcal{W}}_\ell$ *and* $\max_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\langle \underline{\mathbf{w}}_* - \underline{\mathbf{w}}, \theta^*\rangle \leqslant 4\epsilon_\ell$ *for all* $\ell \in \mathbb{N}$.

Proof. For any $\mathbb{V} \subseteq \underline{\mathcal{W}}_\ell$ be the active set and $\underline{\mathbf{w}} \in \mathbb{V}$ define

$$\mathcal{E}_{\underline{\mathbf{w}},\ell}(\mathbb{V}) = \left\{ \left| \left\langle \underline{\mathbf{w}} - \underline{\mathbf{w}}_*, \widehat{\theta}_\ell(\mathbb{V}) - \theta^* \right\rangle \right| \leqslant \epsilon_\ell \right\} \tag{D.10}$$

where it is implicit that $\widehat{\theta}_\ell := \widehat{\theta}_\ell(\mathbb{V})$ is the design constructed in the algorithm at stage ℓ with respect to $\underline{\mathcal{W}}_\ell = \mathbb{V}$. Also note that $\delta_\ell = \frac{\delta}{4\ell^2}$. Given $\underline{\mathcal{W}}_\ell$, with probability at least $1 - 2 \cdot \frac{\delta}{4\ell^2|\underline{\mathcal{W}}|}$

$$\begin{aligned}
&\left| \left\langle \underline{\mathbf{w}} - \underline{\mathbf{w}}_*, \widehat{\theta}_\ell - \theta^* \right\rangle \right| \\
&\overset{(a)}{\leqslant} \sqrt{64 B_*^\ell k \log(1 + \frac{\tau_{\ell-1}^G}{\lambda}) \log\left(4\ell^2 |\underline{\mathcal{W}}|/\delta\right)} \\
&\leqslant \frac{(k \log(1 + \frac{\tau_{\ell-1}^G}{\lambda}))^2}{\sqrt{\tau_\ell^G}} \sqrt{64 B_*^\ell \log\left(4\ell^2 |\underline{\mathcal{W}}|/\delta\right)} \\
&\overset{(b)}{\leqslant} \sqrt{\frac{\|\underline{\mathbf{w}} - \underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \mathbb{V}} \mathbf{b}^G_{\ell,\underline{\mathbf{w}}}(\mathbb{V}) \underline{\mathbf{w}}\ \underline{\mathbf{w}}^\top + \mathbf{\Lambda}_\ell\right)^{-1}}}{64 B_*^\ell \epsilon_\ell^{-2} \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \log\left(4\ell^2 |\underline{\mathcal{W}}|/\delta\right)}} \sqrt{64 B_*^\ell \log\left(4\ell^2 |\underline{\mathcal{W}}|/\delta\right)} \\
&\leqslant \sqrt{\frac{\|\underline{\mathbf{w}} - \underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \mathbb{V}} \mathbf{b}^G_{\ell,\underline{\mathbf{w}}}(\mathbb{V}) \underline{\mathbf{w}}\ \underline{\mathbf{w}}^\top + \mathbf{\Lambda}_\ell\right)^{-1}}}{\epsilon_\ell^{-2} \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) \log\left(4\ell^2 |\underline{\mathcal{W}}|/\delta\right)}} \sqrt{\log\left(4\ell^2 |\underline{\mathcal{W}}|/\delta\right)} \\
&\overset{(c)}{=} \epsilon_\ell
\end{aligned}$$

where, (a) follows Proposition 1. The (b) follows as

$$(k \log(1 + \frac{\tau_{\ell-1}^G}{\lambda}))^2 \leqslant (k \log(1 + \frac{\tau_\ell^G}{\lambda}))^2 := \rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) = \|\underline{\mathbf{w}} - \underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \mathbb{V}} \mathbf{b}^G_{\ell,\underline{\mathbf{w}}}(\mathbb{V}) \underline{\mathbf{w}}\ \underline{\mathbf{w}}^\top + \mathbf{\Lambda}_\ell\right)^{-1}}.$$

The (c) follows as $\rho(\mathcal{Y}(\underline{\mathcal{W}}_\ell)) = \|\underline{\mathbf{w}} - \underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \mathbb{V}} \mathbf{b}^G_{\ell,\underline{\mathbf{w}}}(\mathbb{V}) \underline{\mathbf{w}}\ \underline{\mathbf{w}}^\top + \mathbf{\Lambda}_\ell\right)^{-1}}$. By exactly

the same sequence of steps as above, we have

$$\mathbb{P}\left(\bigcap_{\ell=1}^{\infty}\bigcap_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\left\{\left|\left\langle \underline{\mathbf{w}}-\underline{\mathbf{w}}_*,\widehat{\theta}_t-\theta^*\right\rangle\right|>\epsilon_t\right\}\right)=\mathbb{P}\left(\bigcap_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell}\bigcap_{\ell=1}^{\infty}\mathcal{E}_{\underline{\mathbf{w}},\ell}\left(\underline{\mathcal{W}}_\ell\right)\right)\geqslant 1-\delta,$$

so assume these events hold. Consequently, for any $\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell$

$$\begin{aligned}\left\langle \underline{\mathbf{w}}'-\underline{\mathbf{w}}_*,\widehat{\theta}_\ell\right\rangle&=\left\langle \underline{\mathbf{w}}'-\underline{\mathbf{w}}_*,\widehat{\theta}_\ell-\theta^*\right\rangle+\left\langle \underline{\mathbf{w}}'-\underline{\mathbf{w}}_*,\theta^*\right\rangle\\&\leqslant\left\langle \underline{\mathbf{w}}'-\underline{\mathbf{w}}_*,\widehat{\theta}_\ell-\theta^*\right\rangle\\&\leqslant\epsilon_\ell\end{aligned}$$

so that $\underline{\mathbf{w}}_*$ would survive to round $\ell+1$. And for any $\underline{\mathbf{w}}\in\underline{\mathcal{W}}_\ell$ such that $\left\langle \underline{\mathbf{w}}_*-\underline{\mathbf{w}},\theta^*\right\rangle>2\epsilon_\ell$ we have

$$\begin{aligned}\max_{\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell}\left\langle \underline{\mathbf{w}}'-\underline{\mathbf{w}},\widehat{\theta}_\ell\right\rangle&\geqslant\left\langle \underline{\mathbf{w}}_*-\underline{\mathbf{w}},\widehat{\theta}_\ell\right\rangle\\&=\left\langle \underline{\mathbf{w}}_*-\underline{\mathbf{w}},\widehat{\theta}_\ell-\theta^*\right\rangle+\left\langle \underline{\mathbf{w}}_*-\underline{\mathbf{w}},\theta^*\right\rangle\\&>-\epsilon_\ell+2\epsilon_\ell\\&=\epsilon_\ell\end{aligned}$$

which implies this $\underline{\mathbf{w}}$ would be eliminated. Note that this implies that $\max_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_{\ell+1}}\left\langle \underline{\mathbf{w}}_*-\underline{\mathbf{w}},\theta^*\right\rangle\leqslant 2\epsilon_\ell=4\epsilon_{\ell+1}$. Hence, the claim of the lemma follows. □

Final Sample Complexity Bound for Single Task Setting

Theorem 1. (***Restatement***) *With probability at least* $1-\delta$, *GOBLIN returns the best arms* $\mathbf{x}_*$, $\mathbf{z}_*$, *and the number of samples used is bounded by*

$$\widetilde{O}\left(\frac{(d_1+d_2)r}{\Delta^2}+\frac{\sqrt{d_1d_2r}}{S_r}\right)$$

where, $\Delta = \min_{\mathbf{x}\in\mathcal{X}\setminus\{\mathbf{x}_*\},\mathbf{z}\in\mathcal{Z}\setminus\{\mathbf{z}_*\}}(\mathbf{x}_*^\top\boldsymbol{\Theta}_*\mathbf{z}_* - \mathbf{x}^\top\boldsymbol{\Theta}_*\mathbf{z})$*, and* S_r *is the* r*-th largest singular value of* $\boldsymbol{\Theta}_*$*.*

Proof. For the rest of the proof we have that the good events $\mathcal{F}_\ell \bigcap \mathcal{E}_{\underline{\mathbf{w}},\ell}(\underline{\mathcal{W}}_\ell)$ holds true for each phase ℓ with probability greater than $(1-\delta)$. The two events are defined in (D.4) and (D.10).

Second Stage: Define $\underline{\mathcal{A}}_\ell = \{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell : \langle \underline{\mathbf{w}}_* - \underline{\mathbf{w}}, \theta^* \rangle \leqslant 4\epsilon_\ell\}$. Note that by assumption $\underline{\mathcal{W}} = \underline{\mathcal{W}}_1 = \underline{S}_1$. The above lemma implies that with probability at least $1-\delta$ we have $\bigcap_{\ell=1}^{\infty}\{\underline{\mathcal{W}}_\ell \subseteq \underline{\mathcal{A}}_\ell\}$. This implies that

$$\begin{aligned}
\rho^G(\underline{\mathcal{W}}_\ell) &= \min_{\mathbf{b}\in\Delta_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_\ell} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \\
&\leqslant \min_{\mathbf{b}\in\Delta_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{A}}_\ell} \|\underline{\mathbf{w}} - \underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \\
&= \rho^G(\underline{\mathcal{A}}_\ell).
\end{aligned}$$

Define $k_1^\ell = 8k\log(1+\tau_{\ell-1}^G/\lambda)$. For $\ell \geqslant \lceil \log_2(4\Delta^{-1}) \rceil$ we have that $\underline{\mathcal{A}}_\ell =$

$\{\underline{\mathbf{w}}_*\}$, thus, the sample complexity to identify $\underline{\mathbf{w}}_*$ is equal to

$$\sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \left\lceil \tau_\ell^G \widehat{b}_{\ell,\underline{\mathbf{w}}}^G \right\rceil = \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(\frac{(k_1^\ell+1)k_1^\ell}{2} + \tau_\ell^G \right)$$

$$= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(\frac{(k_1^\ell+1)k_1^\ell}{2} + 2\epsilon_\ell^{-2}\rho^G(\underline{\mathcal{W}}_\ell)B_*^\ell \log\left(4k_1^\ell \ell^2 |\underline{\mathcal{W}}|/\delta\right) \right)$$

$$\overset{(a)}{\leqslant} 2 \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(\frac{(k+1)k}{2} \log^2(1+\tau_{\ell-1}^G) + 2\epsilon_\ell^{-2}\rho^G(\underline{\mathcal{W}}_\ell)B_*^\ell \log\left(4k_1^\ell \ell^2 |\underline{\mathcal{W}}|/\delta\right) \right)$$

$$\overset{(b)}{\leqslant} 2\frac{(k+1)k}{2} \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(\log^2(1+\tau_{\ell-1}^G) + 8\epsilon_\ell^{-2}\rho^G(\underline{\mathcal{W}}_\ell)B_*^\ell \log\left(4k_1^\ell \ell^2 |\underline{\mathcal{W}}|/\delta\right) \right)$$

$$\overset{(c)}{\leqslant} (k+1)k \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(1 + 16\epsilon_\ell^{-2}\rho^G(\underline{\mathcal{W}}_\ell)B_*^\ell \log^2\left(4k_1^\ell \ell^2 |\underline{\mathcal{W}}|/\delta\right) \right)$$

$$\overset{(d)}{\leqslant} (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_\ell^{-2} f\left(\underline{\mathcal{A}}_\ell\right) B_*^\ell \log\left(4k\ell^2 |\underline{\mathcal{W}}|/\delta\right)$$

$$\overset{(e)}{\leqslant} (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_\ell^{-2} f\left(\underline{\mathcal{A}}_\ell\right) \left(64\lambda S^2 + 64\tau_{\ell-1}^G\right) \log\left(4k\ell^2 |\underline{\mathcal{W}}|/\delta\right)$$

$$= (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_\ell^{-2} f\left(\underline{\mathcal{A}}_\ell\right) \left(64\lambda S^2\right) \log\left(4k\ell^2 |\underline{\mathcal{W}}|/\delta\right)$$

$$+ (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_\ell^{-2} f\left(\underline{\mathcal{A}}_\ell\right) \left(64\tau_{\ell-1}^G\right) \log\left(4k\ell^2 |\underline{\mathcal{W}}|/\delta\right)$$

$$\overset{(f)}{\leqslant} (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 64\epsilon_\ell^{-2} f\left(\underline{\mathcal{A}}_\ell\right) \left(64\lambda S^2\right) \log\left(4k\ell^2 |\underline{\mathcal{W}}|/\delta\right)$$

$$\leqslant (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + 2048\lambda S^2 \log\left(\frac{4k\log_2^2\left(8\Delta^{-1}\right)|\underline{\mathcal{W}}|}{\delta}\right) \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 2^{2\ell} f\left(\underline{\mathcal{A}}_\ell\right)$$

where, (a) follows as $\log^2(1+\tau_{\ell-1}^G/\lambda) \leqslant \log^2(1+\tau_{\ell-1}^G)$, (b) follows by noting that $\log(x\log(1+x)) \leqslant 2\log(x)$ for any $x > 1$. The (c) follows by subsuming the $\log^2(1+\tau_{\ell-1}^G)$ into $2\tau_\ell^G$. The (d) follows as $\log(1+\tau_{\ell-1}^G) < \tau_\ell^G$ which enables us to replace the k_1^ℓ inside the log with an additional factor of 2. The (e) follows by noting that

$$\begin{aligned} B_*^\ell &\leqslant 64(\sqrt{\lambda}S + \sqrt{\lambda_\ell^\perp}S_\ell^\perp) \\ &\leqslant 64\lambda S^2 + \left(\frac{64\tau_{\ell-1}^G}{8(d_1+d_2)r\log(1+\frac{\tau_{\ell-1}^G}{\lambda})}\right)\cdot\left(\frac{8d_1d_2r}{\tau_\ell^E S_r^2}\log\left(\frac{d_1+d_2}{\delta_\ell}\right)\right) \\ &\overset{(a_1)}{\leqslant} 64\lambda S^2 + 64\tau_{\ell-1}^G. \end{aligned} \tag{D.11}$$

where, (a_1) follows by first substituting the value of $\tau_\ell^E \coloneqq \frac{\sqrt{8d_1d_2r\log(4\ell^2|\mathcal{W}|/\delta_\ell)}}{S_r}$ and noting that $\sqrt{(d_1d_2r)} \leqslant (d_1+d_2)r$ and cancelling out the other terms. Finally the (f) follows by subsuming the $\tau_{\ell-1}^G$ with a factor of 2 into the quantity of τ_ℓ^G. Then it follows that

$$
\begin{aligned}
\rho_*^{\mathrm{G}} &= \inf_{\mathbf{b}\in\triangle_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \frac{\|\underline{\mathbf{w}}-\underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}}}{\left(\langle \underline{\mathbf{w}}-\underline{\mathbf{w}}_*, \theta^*\rangle\right)^2} \\
&= \inf_{\mathbf{b}\in\triangle_{\underline{\mathcal{W}}}} \max_{\ell\leqslant\left\lceil\log_2(4\Delta^{-1})\right\rceil} \max_{\underline{\mathbf{w}}\in\underline{\mathcal{A}}_\ell} \frac{\|\underline{\mathbf{w}}-\underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}}}{\left(\langle \underline{\mathbf{w}}-\underline{\mathbf{w}}_*, \theta^*\rangle\right)^2} \\
&\geqslant \frac{1}{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} \inf_{\mathbf{b}\in\triangle_{\underline{\mathcal{W}}}} \sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} \max_{\underline{\mathbf{w}}\in\underline{\mathcal{A}}_\ell} \frac{\|\underline{\mathbf{w}}-\underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}}}{\left(\langle \underline{\mathbf{w}}-\underline{\mathbf{w}}_*, \theta^*\rangle\right)^2} \\
&\geqslant \frac{1}{16\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} \sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} 2^{2\ell} \inf_{\mathbf{b}\in\triangle_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}}\in\underline{\mathcal{A}}_\ell} \|\underline{\mathbf{w}}-\underline{\mathbf{w}}_*\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}} \\
&\geqslant \frac{1}{64\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} \sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} 2^{2\ell} \inf_{\mathbf{b}\in\triangle_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{A}}_\ell} \|\underline{\mathbf{w}}-\underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}} \\
&\geqslant \frac{1}{64\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} \sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} 2^{2\ell} \mathrm{f}\left(\underline{\mathcal{A}}_\ell\right).
\end{aligned}
$$

This implies that

$$
\sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} 2^{2\ell} \mathrm{f}\left(\underline{\mathcal{A}}_\ell\right) \leqslant \rho_*^{\mathrm{G}} 64 \left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil
$$

Plugging this back we get

$$
\begin{aligned}
&\sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil} \sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \left\lceil \tau_\ell^{\mathrm{G}} \widehat{\mathbf{b}}_{\ell,\underline{\mathbf{w}}} \right\rceil \\
&\leqslant (\mathrm{k}+1)\mathrm{k}\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil + 2048\lambda \mathrm{S}^2 \log\left(\frac{8\mathrm{k}\log_2^2\left(8\Delta^{-1}\right)|\underline{\mathcal{W}}|}{\delta}\right) \rho_*^{\mathrm{G}} 64 \left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil \\
&\leqslant (\mathrm{k}+1)\mathrm{k}\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil + \mathrm{C}_2\lambda \mathrm{S}^2 \log\left(\frac{8\mathrm{k}\log_2^2\left(8\Delta^{-1}\right)|\underline{\mathcal{W}}|}{\delta}\right) \rho_*^{\mathrm{G}} \left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil
\end{aligned}
$$

for some constant $C_2 > 0$. Now to understand the bound we need the following: Let $\text{conv}(\underline{\mathcal{W}} \cup -\underline{\mathcal{W}})$ denote the convex hull of $\underline{\mathcal{W}} \cup -\underline{\mathcal{W}}$, and for any set $\mathcal{Y} \subset \mathbb{R}^p$ define the gauge of $\mathcal{Y}$

$$\gamma_{\mathcal{Y}} = \max\{c > 0 : c\mathcal{Y} \subset \text{conv}(\underline{\mathcal{W}} \cup -\underline{\mathcal{W}})\} \tag{D.12}$$

In the case where $\mathcal{Y}$ is a singleton $\mathcal{Y} = \{y\}, \gamma(y) := \gamma_{\mathcal{Y}}$ is the gauge norm of $\mathbf{y}$ with respect to $\text{conv}(\underline{\mathcal{W}} \cup -\underline{\mathcal{W}})$. We can provide a natural upper bound for $\rho(\mathcal{Y})$ in terms of the gauge. Observe that

$$\begin{aligned}
\rho_*^G &= \inf_{\mathbf{b} \in \triangle_{\underline{\mathcal{W}}}} \max_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \\
&= \frac{1}{\gamma_{\mathcal{Y}}^2} \inf_{\mathbf{b} \in \triangle_{\underline{\mathcal{W}}}} \max_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{y}\gamma_{\mathcal{Y}}\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \\
&\leqslant \frac{1}{\gamma_{\mathcal{Y}}^2} \inf_{\mathbf{b} \in \triangle_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}} \in \text{conv}(\underline{\mathcal{W}} \bigcup -\underline{\mathcal{W}})} \|\underline{\mathbf{w}}\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \\
&\overset{(a)}{=} \frac{1}{\gamma_{\mathcal{Y}}^2} \inf_{\mathbf{b} \in \triangle_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \|\underline{\mathbf{w}}\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \overset{(b)}{\leqslant} k^{3/2} \\
&\leqslant \frac{k}{\gamma_{\mathcal{Y}}^2} O\left(B_* \log\left(\frac{k \log_2\left(\Delta^{-1}\right) |\underline{\mathcal{W}}|}{\delta}\right) \left\lceil \log_2\left(\Delta^{-1}\right) \right\rceil\right)
\end{aligned}$$

The (a) follows from the fact that the maximum value of a convex function on a convex set must occur at a vertex. The (b) follows from Kiefer-wolfowitz theorem for $\underline{\mathbf{w}} \in \mathbb{R}^p$ such that $\inf_{\mathbf{b} \in \triangle_{\underline{\mathcal{W}}}} \max_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \|\underline{\mathbf{w}}\|^2_{\left(\sum_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}} \underline{\mathbf{w}}\, \underline{\mathbf{w}}^\top + \boldsymbol{\Lambda}\right)^{-1}} \leqslant k \log(1 + \frac{\tau_{\ell-1}^G}{\lambda})$. The simplified sample complexity for the second stage is given by

$$N_2 \leqslant O\left(\frac{k}{\Delta^2} \log\left(\frac{k \log_2\left(\Delta^{-1}\right) |\underline{\mathcal{W}}|}{\delta}\right)\right) = \widetilde{O}\left(\frac{(d_1 + d_2)r}{\Delta^2}\right)$$

where $\Delta = \min_{\underline{\mathbf{w}} \in \underline{\mathcal{W}}} (\underline{\mathbf{w}}_* - \underline{\mathbf{w}})^\top \theta_* \overset{(a_1)}{=} \min_{\mathbf{x} \in \mathcal{X} \setminus \{\mathbf{x}_*\}, \mathbf{z} \in \mathcal{Z} \setminus \{\mathbf{z}_*\}} (\mathbf{x}_*^\top \boldsymbol{\Theta}_* \mathbf{z}_* - \mathbf{x}^\top \boldsymbol{\Theta}_* \mathbf{z})$. The (a_1) follows by reshaping the arms in $\underline{\mathcal{W}}$ to recover the arms in $\mathcal{X}$ and

$\mathcal{Z}$.

1st Stage: First recall that the E-optimal design in step 3 of Algorithm 5 satisfies the Assumption 12 as the sample distribution $\mathcal{D}$ has finite second order moments. For the first stage first observe that by plugging in the definition of $\tau_\ell^{\mathrm{E}} = \frac{\sqrt{8d_1d_2r\log(4\ell^2|\mathcal{W}|/\delta)}}{S_r}$ we get

$$\begin{aligned}\left\|\theta^*_{k+1:p}\right\|_2^2 &= \sum_{i>r\wedge j>r} H_{ij}^2 = \left\|(\widehat{\mathbf{U}}_\ell^\perp)^\top \left(\mathbf{U}^*\mathbf{S}^*\mathbf{V}^{*\top}\right)\widehat{\mathbf{V}}_\ell^\perp\right\|_{\mathrm{F}}^2 \\ &\leqslant \left\|(\widehat{\mathbf{U}}_\ell^\perp)^\top\mathbf{U}^*\right\|_{\mathrm{F}}^2 \|\mathbf{S}^*\|_2^2 \left\|(\widehat{\mathbf{V}}_\ell^\perp)^\top\mathbf{V}^*\right\|_{\mathrm{F}}^2 \leqslant O\left(\frac{d_1d_2r}{\tau_\ell^{\mathrm{E}}S_r^2}\log\left(\frac{d_1+d_2}{\delta}\right)\right) \\ &= O\left(\frac{\sqrt{d_1d_2r}}{S_r}\log\left(\frac{d_1+d_2}{\delta}\right)\right)\end{aligned}$$

which implies $\left\|\theta^*_{k+1:p}\right\|_2 = \widetilde{O}\left(\sqrt{d_1d_2r}/S_r\right)$. Now we bound the sample complexity from the first stage. From the first stage we can show that we have for the arm set $\overline{\mathcal{W}}$

$$\begin{aligned}N_1 &= \sum_{\ell=1}^{\lceil\log_2(4\Delta^{-1})\rceil}\sum_{\overline{\mathbf{w}}\in\overline{\mathcal{W}}}\left\lceil\tau_\ell^{\mathrm{E}}\widehat{b}_{\ell,\overline{\mathbf{w}}}^{\mathrm{E}}\right\rceil \\ &= \sum_{\ell=1}^{\lceil\log_2(4\Delta^{-1})\rceil}\left(\frac{(p+1)p}{2}+\tau_\ell^{\mathrm{E}}\right) \\ &= \sum_{\ell=1}^{\lceil\log_2(4\Delta^{-1})\rceil}\left(\frac{(p+1)p}{2}+\frac{\sqrt{8d_1d_2r\log\left(4\ell^2|\mathcal{W}|/\delta\right)}}{S_r}\right) \\ &\leqslant (p+1)p\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil + 32\frac{\sqrt{d_1d_2r}}{S_r}\log\left(\frac{4\log_2^2\left(8\Delta^{-1}\right)|\mathcal{W}|}{\delta}\right)\left\lceil\log_2\left(4\Delta^{-1}\right)\right\rceil \\ &\overset{(a)}{=} O\left(\frac{\sqrt{d_1d_2r}}{S_r}\log\left(\frac{4\log_2^2\left(8\Delta^{-1}\right)|\mathcal{W}|}{\delta}\right)\right) = \widetilde{O}\left(\frac{\sqrt{d_1d_2r}}{S_r}\right)\end{aligned}$$

where, (a) follows as $p = d_1d_2$. Combining N_1 and N_2 gives the claim of the theorem. □

D.5 Multi-Task Pure Exploration Proofs

Remark D.15. (***Comparison with Du et al. (2023)***) *In this remark, we discuss a key comparison of DouExpDes (Du et al., 2023) with GOBLIN. Note that DouExpDes does not implement the second stage of finding the* $\mathbf{S}_{m,*} \in \mathbb{R}^{k_1 \times k_2}$ *for each of the* m *bilinear bandits. Hence, DouExpDes does not rotate the arms so that the last* $(k_1 - r) \cdot (k_2 - r)$ *components are from the complementary subspaces of the left and right eigenvectors of* $\mathbf{S}_{m,*}$*. This results in DouExpDes suffering a sample complexity of* $\widetilde{O}(k_1 k_2/\Delta^2)$ *even though it learns the common feature extractors shared across the tasks. In contrast GOBLIN uses the second stage to learn* $\mathbf{S}_{m,*} \in \mathbb{R}^{k_1 \times k_2}$ *and reduces the latent bilinear bandits problem of* $k_1 k_2$ *dimension to* $(k_1 + k2)r$ *dimension by rotating the arms so that the last* $(k_1 - r) \cdot (k_2 - r)$ *components are from the complementary subspaces of the left and right eigenvectors of* $\mathbf{S}_{m,*}$*. Hence, GOBLIN suffers a sample complexity of* $\widetilde{O}((k_1 + k_2)r/\Delta^2)$.

Remark D.16. (***Arm set***) *The observable left and right arm sets* $\mathcal{X}$ *and* $\mathcal{Z}$ *are common across the* M *tasks. This leads to each task estimating the same E-optimal design in line* 3 *of Algorithm 6 of stage* 1*. Note that Du et al. (2023) also uses a similar idea of the same arm set* $\mathcal{X}$ *shared across tasks in the linear bandit setting. Observe that if each task has access to its own separate arm sets* $\mathcal{X}_m$ *and* $\mathcal{Z}_m$*, then each of the* m*-tasks has to estimate a separate E-optimal design for the stage* 1*. This will lead to the sample complexity of the first stage scaling as* $\widetilde{O}(M\sqrt{d_1 d_2}/S_r)$ *instead of* $\widetilde{O}(\sqrt{d_1 d_2 r}/S_r)$.

Good Event: We first recall the total stage 1 length as

$$\tau_\ell^E := \frac{\sqrt{8 d_1 d_2 r \log(4\ell^2|\mathcal{W}|/\delta_\ell)}}{S_r}.$$

Then define the good event $\mathcal{F}_\ell$ in phase ℓ that GOBLIN has a good estimate

of $\mathbf{Z}_* = \frac{1}{M}\sum_{m=1}^{M} \mathbf{\Theta}_m$ as follows: For any phase $\ell > 0$

$$\mathcal{F}_\ell := \left\{ \left\| \widehat{Z}_\ell - \mu^* \mathbf{Z}_* \right\|_F^2 \leqslant \frac{C_1 d_1 d_2 r \log\left(\frac{2(d_1+d_2)}{\delta_\ell}\right)}{\tau_\ell^E} \right\}, \tag{D.13}$$

where, $C_1 = 36\left(4 + S_0^2\right) C$, $\|\mathbf{X}\|_F, \|\mathbf{\Theta}_*\|_F \leqslant S_0$, some nonzero constant μ^*, $\mathbb{E}\left[(S^p(\mathbf{X}))_{ij}^2\right] \leqslant C, \forall i, j$, and $\widehat{\mathbf{\Theta}}_\ell$ is the estimate from (5.8). Then define the event

$$\mathcal{F} := \bigcap_{\ell=1}^{\infty} \mathcal{F}_\ell \tag{D.14}$$

Then we start by modifying Theorem D.3 for the multi-task setting. We first prove this support lemma for the loss function defined in (5.8).

Lemma D.17. *Let $L : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}$ is the loss function defined in (5.8). Then by setting*

$$t = \sqrt{2 d_1 d_2 C \left(4 + S_0^2\right) \log\left(\frac{2\left(d_1 + d_2\right)}{\delta_\ell}\right)},$$

$$\nu = \frac{t}{(4 + S_0) C d_1 d_2 \sqrt{M \tau_\ell^E}} = \sqrt{\frac{2 \log\left(\frac{2(d_1+d_2)}{\delta_\ell}\right)}{M \tau_\ell^E d_1 d_2 C \left(4 + S_0^2\right)}},$$

we have with probability at least $1 - \delta_\ell$, it holds that

$$\mathbb{P}\left(\left\| \nabla L\left(\mu^* \mathbf{Z}_*\right) \right\|_{op} \geqslant \frac{2t}{\sqrt{M \tau_\ell^E}} \right) \leqslant \delta_\ell,$$

where $\mu^* = \frac{2}{M}\mathbb{E}\left[\langle \mathbf{X}_m, \mathbf{Z}_* \rangle\right] > 0$, and $\mathbf{X}_m = \mathbf{x}_m \mathbf{z}_m^\top$.

Proof. Let $\mathbf{Z}_* = \frac{1}{M}\sum_{m=1}^{M} \mathbf{\Theta}_{m,*}$. Let $\mathbf{X}_{m,i} = \mathbf{x}_{m,i} \mathbf{z}_{m,i}^\top$ for $i \in [\tau_\ell^E]$. Based on

the definition of our loss function $L(\cdot)$ in (5.8), we have that

$$\begin{aligned}
\nabla_{x_m} L\left(\mathbf{Z}_*\right) &= \mu^* \mathbf{Z}_* - \frac{2}{M\tau_\ell^E} \sum_{m=1}^{M} \sum_{i=1}^{\tau_\ell^E} \widetilde{\psi}_\nu(r_{m,i} \cdot Q(x_m)) \\
&= \frac{2}{M} \mathbb{E}\left[\langle \mathbf{X}_{m,1}, \mathbf{Z}_* \rangle\right] \mathbf{Z}_* - \frac{2}{M\tau_\ell^E} \sum_{m=1}^{M} \sum_{i=1}^{\tau_\ell^E} \widetilde{\psi}_\nu\left(r_{m,i} \cdot Q\left(\mathbf{X}_{m,i}\right)\right) \\
&\overset{(a)}{=} \frac{2}{M} \mathbb{E}\left[\langle \mathbf{X}_{m,1}, \mathbf{Z}_* \rangle\, Q\left(\mathbf{X}_{m,1}\right)\right] - \frac{2}{M\tau_\ell^E} \sum_{m=1}^{M} \sum_{i=1}^{\tau_\ell^E} \widetilde{\psi}_\nu\left(r_{m,i} \cdot Q\left(\mathbf{X}_{m,i}\right)\right) \\
&\overset{(b)}{=} \frac{2}{M} \left[\mathbb{E}\left(r_{m,1} \cdot Q\left(\mathbf{X}_{m,1}\right)\right) - \frac{1}{M\tau_\ell^E} \sum_{m=1}^{M} \sum_{i=1}^{\tau_\ell^E} \widetilde{\psi}_\nu\left(r_{m,i} \cdot Q\left(\mathbf{X}_{m,i}\right)\right)\right]
\end{aligned}$$

where we have (a) due to the generalized Stein's Lemma stated in Theorem D.1, and (b) comes from the fact that the random noise $\eta_1 = y_1 - \langle \mathbf{X}_1, \mathbf{Z}_* \rangle$ is zero-mean and independent from $\mathbf{X}_1$. Therefore, in order to implement the Theorem D.2, we can see that it suffices to get σ^2 defined as:

$$\begin{aligned}
\sigma^2 = \max \Bigg(& \left\| \frac{2}{M} \sum_{m=1}^{M} \sum_{j=1}^{\tau_\ell^E} \mathbb{E}\left[r_{m,j}^2 Q\left(\mathbf{X}_{m,j}\right) Q\left(\mathbf{X}_{m,j}\right)^\top\right] \right\|_{op}, \\
& \left\| \frac{2}{M} \sum_{m=1}^{M} \sum_{j=1}^{\tau_\ell^E} \mathbb{E}\left[r_{m,j}^2 Q\left(\mathbf{X}_{m,j}\right)^\top Q\left(\mathbf{X}_{m,j}\right)\right] \right\|_{op} \Bigg).
\end{aligned}$$

$$
\begin{aligned}
&\left\| \frac{2}{M} \sum_{m=1}^{M} \sum_{j=1}^{\tau_\ell^E} \mathbb{E}\left[r_{m,j}^2 Q\left(\mathbf{X}_{m,j}\right) Q\left(\mathbf{X}_{m,j}\right)^\top \right] \right\|_{op} \leqslant \tau_\ell^E \times \left\| \mathbb{E}\left[r_{m,1}^2 Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right\|_{op} \\
&\overset{(a)}{=} \tau_\ell^E \times \left\| \mathbb{E}\left[\left(\eta_{m,1} + \langle \mathbf{X}_{m,1}, \mathbf{Z}_* \rangle\right)^2 Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right\|_{op} \\
&\overset{(b)}{=} \tau_\ell^E \times \| \mathbb{E}\left[\eta_{m,1}^2 Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] + \mathbb{E}\left[\langle \mathbf{X}_{m,1}, \mathbf{Z}_* \rangle\right)^2 Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \|_{op} \\
&\overset{(c)}{=} \tau_\ell^E \times \| \mathbb{E}\left(\eta_{m,1}^2\right) \mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] + \mathbb{E}\left[\langle \mathbf{X}_{m,1}, \mathbf{Z}_* \rangle\right)^2 Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \|_{op} \\
&\overset{(d)}{\leqslant} \tau_\ell^E \times \left\| 4\mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] + S_0^2 \mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right\|_{op} \\
&= \left(4 + S_0^2\right) \tau_\ell^E \times \left\| \mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right\|_{op}
\end{aligned}
$$

where the (a) follows by plugging in the definition for reward, (b) follows by the linearity of expectation, (c) follows as noises are independent, and the inequality (d) comes from the fact that $|\langle \mathbf{X}_{m,1}, \mathbf{Z}_* \rangle| \leqslant S_0$, and $Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top$ is always positive semidefinite. Next, since we know that $\mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right]$ is always symmetric and positive semidefinite, and hence we have

$$
\begin{aligned}
&\left\| \mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right\|_{op} \\
&\overset{(a)}{\leqslant} \left\| \mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right\|_{nuc} = \text{trace}\left(\mathbb{E}\left[Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right] \right) \\
&= \mathbb{E}\left[\text{trace}\left(Q\left(\mathbf{X}_{m,1}\right) Q\left(\mathbf{X}_{m,1}\right)^\top \right) \right] = \mathbb{E}\left(\frac{2}{M} \sum_{m=1}^{M} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} Q_{ij}\left(\mathbf{X}_{m,1}\right)^2 \right) \\
&\leqslant d_1 d_2 C.
\end{aligned}
$$

where, in (a) $\|\cdot\|_{nuc}$ denotes the nuclear norm. Therefore, we have that under 1-subGaussian assumption

$$
\left\| \frac{2}{M} \sum_{m=1}^{M} \sum_{j=1}^{\tau_\ell^E} \mathbb{E}\left[r_{m,j}^2 Q\left(\mathbf{X}_{m,j}\right) Q\left(\mathbf{X}_{m,j}\right)^\top \right] \right\|_{op} \leqslant \left(4 + S_0^2\right) d_1 d_2 \tau_\ell^E C.
$$

And similarly, we can prove that

$$\left\| \frac{2}{M} \sum_{m=1}^{M} \sum_{j=1}^{\tau_\ell^E} \mathbb{E}\left[r_{m,j}^2 Q\left(\mathbf{X}_{m,j}\right)^\top Q\left(\mathbf{X}_{m,j}\right) \right] \right\|_{op} \leqslant \left(4 + S_0^2\right) d_1 d_2 \tau_\ell^E C.$$

Therefore, we can take $\sigma^2 = \left(4 + S_0^2\right) d_1 d_2 \tau_\ell^E C$ consequently. By using Theorem D.2, we have

$$\begin{aligned} &\mathbb{P}\left(\left\| \nabla L\left(\mu^* \mathbf{Z}_*\right) \right\|_{op} \geqslant \frac{2t}{\sqrt{M\tau_\ell^E}} \right) \\ &\leqslant 2\left(d_1 + d_2\right) \exp\left(-\nu t \sqrt{M\tau_\ell^E} + \frac{\nu^2 \left(4 + S_0^2\right) C d_1 d_2 \tau_\ell^E}{2} \right) \end{aligned}$$

By plugging the values of t and ν in Theorem D.17, we finish the proof. □

Lemma D.18. *For any low-rank linear model with samples $\mathbf{X}_1 \ldots, \mathbf{X}_{\tau_\ell^E}$ drawn from $\mathcal{X}$ according to $\mathcal{D}$ then for the optimal solution to the nuclear norm regularization problem in (5.2) with $\nu = \sqrt{2 \log\left(2\left(d_1 + d_2\right)/\delta_\ell\right) / \left(\left(4 + S_0^2\right) M \tau_\ell^E d_1 d_2\right)}$ and*

$$\gamma_\ell = 4\sqrt{\frac{2\left(4 + S_0^2\right) C d_1 d_2 \log\left(2\left(d_1 + d_2\right)/\delta_\ell\right)}{M\tau_\ell^E}},$$

with probability at least $1 - \delta_\ell$ it holds that:

$$\left\| \widehat{Z}_\ell - \mu^* \mathbf{Z}_* \right\|_F^2 \leqslant \frac{C_1 d_1 d_2 r \log\left(\frac{2(d_1+d_2)}{\delta_\ell}\right)}{M\tau_\ell^E},$$

for $C_1 = 36\left(4 + S_0^2\right) C$, $\|\mathbf{X}\|_F, \|\mathbf{Z}_*\|_F \leqslant S_0$, some nonzero constant μ^*, and $\mathbb{E}\left[\left(S^P(\mathbf{X})\right)_{ij}^2\right] \leqslant C, \forall i, j$. Summing over all phases $\ell \geqslant 1$ it follows that $\mathbb{P}(\mathcal{F}) \geqslant 1 - \delta/2$.

Proof. Since the estimator $\widehat{Z}_\ell$ minimizes the regularized loss function de-

fined in (5.8), we have

$$L(\widehat{Z}_\ell) + \gamma_\ell \|\widehat{Z}_\ell\|_{nuc} \leqslant L(\mu^* \mathbf{Z}_*) + \gamma_\ell \|\mu^* \mathbf{Z}_*\|_{nuc}$$

And due to the fact that $L(\cdot)$ is a quadratic function, we have the following expression based on multivariate Taylor's expansion:

$$L(\widehat{Z}_\ell) - L(\mu^* \mathbf{Z}_*) = \langle \nabla L(\mu^* \mathbf{Z}_*), \boldsymbol{\Theta} \rangle + 2\|\boldsymbol{\Theta}\|_F^2, \quad \text{where } \boldsymbol{\Theta} = \widehat{Z}_\ell - \mu^* \mathbf{Z}_*$$

By rearranging the above two results, we can deduce that

$$\begin{aligned} 2\|\boldsymbol{\Theta}\|_F^2 &\leqslant -\langle \nabla L(\mu^* \mathbf{Z}_*), \boldsymbol{\Theta} \rangle + \gamma_\ell \|\mu^* \mathbf{Z}_*\|_{nuc} - \gamma_\ell \|\widehat{Z}_\ell\|_{nuc} \\ &\overset{(a)}{\leqslant} \|\nabla L(\mu^* \mathbf{Z}_*)\|_{op} \|\boldsymbol{\Theta}\|_{nuc} + \gamma_\ell \|\mu^* \mathbf{Z}_*\|_{nuc} - \gamma_\ell \|\widehat{Z}_\ell\|_{nuc}, \end{aligned} \tag{D.15}$$

where (a) comes from the duality between matrix operator norm and nuclear norm. Next, we represent the saturated SVD of $\mathbf{Z}_*$ in the main paper as $\mathbf{Z}_* = \overline{U}\mathbf{D}\mathbf{V}^\top$ where $\overline{U} \in \mathbb{R}^{d_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{d_2 \times r}$, and here we would work on its full version, i.e.

$$\mathbf{Z}_* = (\overline{U}, \overline{U}_\perp) \begin{pmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{pmatrix} (\mathbf{V}, \mathbf{V}_\perp)^\top = (\overline{U}, \overline{U}_\perp) \mathbf{D}^* (\mathbf{V}, \mathbf{V}_\perp)^\top$$

where we have $\overline{U}_\perp \in \mathbb{R}^{d_1 \times (d_1 - r)}$, $\mathbf{D}^* \in \mathbb{R}^{d_1 \times d_2}$ and $\mathbf{V}_\perp \in \mathbb{R}^{d_2 \times (d_2 - r)}$. Furthermore, we define

$$\boldsymbol{\Lambda} = (\overline{U}, \overline{U}_\perp)^\top \boldsymbol{\Theta} (\mathbf{V}, \mathbf{V}_\perp) = \begin{pmatrix} \overline{U}^\top \boldsymbol{\Theta} \mathbf{V} & \overline{U}^\top \boldsymbol{\Theta} \mathbf{V}_\perp \\ \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V} & \overline{U}_\perp^\top \boldsymbol{\Theta} V_\perp \end{pmatrix} = \boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2$$

where we write

$$\boldsymbol{\Lambda}_1 = \begin{pmatrix} 0 & 0 \\ 0 & \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V}_\perp \end{pmatrix}, \quad \boldsymbol{\Lambda}_2 = \begin{pmatrix} \overline{U}^\top \boldsymbol{\Theta} \mathbf{V} & \overline{U}^\top \boldsymbol{\Theta} \mathbf{V}_\perp \\ \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V} & 0 \end{pmatrix}.$$

Afterward, it holds that

$$\begin{aligned}\|\widehat{Z}_\ell\|_{\text{nuc}} = \|\mu^* \mathbf{Z}_* + \boldsymbol{\Theta}\|_{\text{nuc}} &\overset{(a)}{=} \left\| \left(\overline{U}, \overline{U}_\perp\right) \left(\mu^* \mathbf{D}^* + \boldsymbol{\Lambda}\right) \left(\mathbf{V}, \mathbf{V}_\perp\right)^\top \right\|_{\text{nuc}} \\ &\overset{(b)}{=} \|\mu^* \mathbf{D}^* + \boldsymbol{\Lambda}\|_{\text{nuc}} + \|\mu^* \mathbf{D}^* + \boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\|_{\text{nuc}} \\ &\geqslant \|\mu^* \mathbf{D}^* + \boldsymbol{\Lambda}_1\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} \\ &= \|\mu^* \mathbf{D}\|_{\text{nuc}} + \|\boldsymbol{\Lambda}_1\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} \\ &= \|\mu^* \mathbf{Z}_*\|_{\text{nuc}} + \|\boldsymbol{\Lambda}_1\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_2\|_{\text{nuc}}, \end{aligned} \tag{D.16}$$

where, (a) follows from the definition of $\mathbf{Z}_*$, and (b) follows from the definition of $\boldsymbol{\Lambda}$. This implies that

$$\|\mu^* \mathbf{Z}_*\|_{\text{nuc}} - \|\widehat{Z}_\ell\|_{\text{nuc}} \leqslant \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_1\|_{\text{nuc}}.$$

Combining (D.15) and (D.16), we have that

$$2\|\boldsymbol{\Theta}\|_F^2 \leqslant \left(\|\nabla L\left(\mu^* \mathbf{Z}_*\right)\|_{\text{op}} + \gamma_\ell\right) \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} + \left(\|\nabla L\left(\mu^* \mathbf{Z}_*\right)\|_{\text{op}} - \gamma_\ell\right) \|\boldsymbol{\Lambda}_1\|_{\text{nuc}}.$$

Then, we refer to the setting in our Theorem D.17, and we choose $\gamma_\ell = 4t/\sqrt{M\tau_\ell^E}$ where the value of t is determined in Theorem D.17, i.e.

$$\gamma_\ell = 4\sqrt{\frac{2\left(4 + S_0^2\right) C d_1 d_2 \log\left(2\left(d_1 + d_2\right)/\delta_\ell\right)}{M\tau_\ell^E}},$$

we know that $\lambda_{T-1} \geqslant 2\left\|\nabla L\left(\mu^* \mathbf{Z}_*\right)\right\|_{\text{op}}$ with probability at least $1 - \delta_\ell$ for any $\delta_\ell \in (0, 1)$. Therefore, with a probability at least $1 - \delta_\ell$, we have

$$2\|\boldsymbol{\Theta}\|_F^2 \leqslant \frac{3}{2}\gamma_\ell \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} - \frac{1}{2}\gamma_\ell \|\boldsymbol{\Lambda}_1\|_{\text{nuc}} \leqslant \frac{3}{2}\gamma_\ell \|\boldsymbol{\Lambda}_2\|_{\text{nuc}}.$$

Since we can easily verify that the rank of $\boldsymbol{\Lambda}_2$ is at most $2r$, and by using

Cauchy-Schwarz Inequality we have that

$$2\|\boldsymbol{\Theta}\|_F^2 \leqslant \frac{3}{2}\gamma_\ell\sqrt{2r}\,\|\boldsymbol{\Lambda}_2\|_F \leqslant \frac{3}{2}\gamma_\ell\sqrt{2r}\|\boldsymbol{\Lambda}\|_F = \frac{3}{2}\gamma_\ell\sqrt{2r}\|\boldsymbol{\Theta}\|_F$$

which implies that

$$\|\boldsymbol{\Theta}\|_F \leqslant \frac{3}{4}\sqrt{2r}\gamma_\ell = 6\sqrt{\frac{(4+S_0^2)\,C d_1 d_2 r \log\left(\frac{2(d_1+d_2)}{\delta_\ell}\right)}{M\tau_\ell^E}}.$$

This implies that $\mathbb{P}(\mathcal{F}_\ell) \geqslant 1-\delta_\ell$. Taking a union bound over all phases $\ell \geqslant 1$ and recalling $\delta_\ell := \frac{\delta}{2\ell^2}$, we obtain

$$\begin{aligned}
\mathbb{P}(\mathcal{F}) &\geqslant 1 - \sum_{\ell=1}^{\infty} \mathbb{P}\left(\mathcal{F}_\ell^c\right) \\
&\geqslant 1 - \sum_{\ell=1}^{\infty} \frac{\delta_\ell}{2} \\
&= 1 - \sum_{\ell=1}^{\infty} \frac{\delta}{4\ell^2} \\
&\geqslant 1 - \frac{\delta}{2}.
\end{aligned}$$

This concludes our proof. □

Define $\mathbf{X}_{\text{batch}}^+ := \left(\mathbf{X}_{\text{batch}}^\top \mathbf{X}_{\text{batch}}\right)^{-1}\mathbf{X}_{\text{batch}}^\top$ where $\mathbf{X}_{\text{batch}}^+$ is constructed through the E-optimal design. Using Lemma C.1 from Du et al. (2023) it holds that

$$\|\mathbf{X}_{\text{batch}}^+\| \leqslant \sqrt{\frac{(1+\beta)\rho_\ell^E}{\overline{p}}}.$$

where $\overline{p} = 180 d_1 d_2/\beta^2$ is the batch size to control the rounding procedure and $\rho_\ell^E = \min_{\mathbf{b}\in\triangle_{\overline{W}}} \left\|\left(\sum_{\overline{\mathbf{w}}\in\overline{W}} \mathbf{b}_{\overline{\mathbf{w}}}\overline{\mathbf{w}}\;\overline{\mathbf{w}}^\top\right)^{-1}\right\|$. It follows then that $\|\mathbf{X}_{\text{batch}}^+\|^2 \leqslant 4\rho_\ell^E$.

Lemma D.19. (*Expectation of* $\widehat{Z}_\ell$). *It holds that* $\mathbb{E}\left[\widehat{Z}_\ell\right] = \mathbf{Z} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{\Theta}_m$.

Proof. Note that we can re-write

$$\begin{aligned}\widehat{Z}_\ell &= \underset{\mathbf{\Theta}\in\mathbb{R}^{d_1\times d_2}}{\arg\min}\, L_\ell(\mathbf{\Theta}) + \gamma_\ell\|\mathbf{\Theta}\|_{\text{nuc}}, L_\ell(\mathbf{\Theta}) \\ &= \langle\mathbf{\Theta},\mathbf{\Theta}\rangle - \tfrac{2}{M\tau_\ell^E}\sum_{m=1}^{M}\sum_{s=1}^{\tau_\ell^E}\langle\widetilde{\psi}_\nu(r_{m,s}\cdot Q(\mathbf{x}_{m,s}\mathbf{z}_{m,s}^\top)),\mathbf{\Theta}\rangle\end{aligned}$$

such that

$$\widehat{Z}_\ell = \frac{2}{M\tau_\ell^E}\sum_{m=1}^{M}\sum_{s=1}^{\tau_\ell^E}\widehat{\mathbf{\Theta}}_{m,s} - \mathbf{X}_{\text{batch}}^{+}\left(\mathbf{X}_{\text{batch}}^{+}\right)^\top$$

where $\widehat{\mathbf{\Theta}}_{m,s} = \langle\mathbf{\Theta},\mathbf{\Theta}\rangle - \frac{2}{Ms}\sum_{m=1}^{M}\langle\widetilde{\psi}_\nu(r_{m,s}\cdot Q(\mathbf{x}_{m,s}\mathbf{z}_{m,s}^\top)),\mathbf{\Theta}\rangle$. Now using Lemma C.2 from Du et al. (2023) we can prove the result of the lemma. □

Lemma D.20. (*Concentration of* $\widehat{B}_{1,\ell}$). *Suppose that event* $\mathcal{F}_\ell$ *holds. Then, for any phase* $\ell > 0$,

$$\left\|(\widehat{B}_{1,\ell}^{\perp})^\top\mathbf{B}_1\right\| \leqslant \frac{c'\rho_\ell^E\sqrt{(d_1+d_2)r}}{S_r\sqrt{M\tau_\ell^E}}\log\left(\frac{16(d_1+d_2)rM\tau_\ell^E}{\delta_\ell}\right),$$

for some constant $c' > 0$ *and* $\rho_\ell^E = \min_{\mathbf{b}\in\triangle_{\overline{W}}}\left\|\left(\sum_{\overline{\mathbf{w}}\in\overline{W}}\mathbf{b}_{\overline{\mathbf{w}}}\overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top\right)^{-1}\right\|$.

Proof. Using the Davis-Kahan $\sin\theta$ Theorem (Bhatia, 2013) and letting τ_ℓ^E

be large enough to satisfy $\left\|\widehat{Z}_\ell - \mu^* \mathbf{Z}_*\right\|_F^2 \leqslant \frac{C_1 d_1 d_2 r \log\left(\frac{2(d_1+d_2)}{\delta_\ell}\right)}{M\tau_\ell^E}$, we have

$$\begin{aligned}
\left\|(\widehat{B}_{1,\ell}^{\perp})^\top \mathbf{B}_1\right\| &\leqslant \frac{\left\|\widehat{Z}_\ell - \mathbb{E}\left[\widehat{Z}_\ell\right]\right\|}{\sigma_r\left(\mathbb{E}\left[\widehat{Z}_\ell\right]\right) - \sigma_{r+1}\left(\mathbb{E}\left[\widehat{Z}_\ell\right]\right) - \left\|\widehat{Z}_\ell - \mathbb{E}\left[\widehat{Z}_\ell\right]\right\|} \\
&\overset{(a)}{\leqslant} \frac{c_0}{S_r}\left\|\widehat{Z}_\ell - \mathbb{E}\left[\widehat{Z}_\ell\right]\right\| \\
&\overset{(b)}{\leqslant} \frac{c c_0 \|\mathbf{X}_{\text{batch}}^+\|^2 \sqrt{(d_1+d_2)r}}{S_r\sqrt{M\tau_\ell^E}} \log\left(\frac{16(d_1+d_2)rM\tau_\ell^E}{\delta_\ell}\right) \\
&\overset{(c)}{\leqslant} \frac{c'\rho_\ell^E\sqrt{(d_1+d_2)r}}{S_r\sqrt{M\tau_\ell^E}} \log\left(\frac{16(d_1+d_2)rM\tau_\ell^E}{\delta_\ell}\right).
\end{aligned}$$

where, (a) follows from Assumption 8, the (b) follows from event $\mathcal{F}_\ell$ and (b) follows as $\|\mathbf{X}_{\text{batch}}^+\|^2 \leqslant 4\rho_\ell^E$. The claim of the lemma follows. □

Lemma D.21. (*Concentration of* $\widehat{B}_{2,\ell}$). *Suppose that event* $\mathcal{F}_\ell$ *holds. Then, for any phase* $\ell > 0$,

$$\left\|(\widehat{B}_{2,\ell}^{\perp})^\top \mathbf{B}_2\right\| \leqslant \frac{c\rho_\ell^E\sqrt{(d_1+d_2)r}}{S_r\sqrt{M\tau_\ell^E}} \log\left(\frac{16(d_1+d_2)rM\tau_\ell^E}{\delta_\ell}\right),$$

for some constant $c' > 0$ *and* $\rho_\ell^E = \min_{\mathbf{b}\in\triangle_{\overline{W}}} \left\|\left(\sum_{\overline{\mathbf{w}}\in\overline{W}} \mathbf{b}_{\overline{\mathbf{w}}}\overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top\right)^{-1}\right\|$.

Proof. The proof follows the same way as Theorem D.20 and using the Davis-Kahan $\sin\theta$ Theorem (Bhatia, 2013)

$$\left\|(\widehat{B}_{2,\ell}^{\perp})^\top \mathbf{B}_2\right\| \leqslant \frac{c\rho_\ell^E\sqrt{(d_1+d_2)r}}{S_r\sqrt{M\tau_\ell^E}} \log\left(\frac{16(d_1+d_2)rM\tau_\ell^E}{\delta_\ell}\right).$$

The claim of the lemma follows. □

Good Event per Task: We now define the good event $\mathcal{F}'_\ell$ in phase ℓ

that GOBLIN has a good estimate of $\mathbf{S}_{m,*}$ as follows: For any phase $\ell > 0$

$$\mathcal{F}'_\ell := \left\{ \left\| \widehat{\mathbf{S}}_{m,\ell} - \mu^* \mathbf{S}_{m,*} \right\|_F^2 \leqslant \frac{C_1 k_1 k_2 r \log\left(\frac{2(k_1+k_2)}{\delta_\ell}\right)}{\tau^E_{m,\ell}} \right\}, \tag{D.17}$$

where, $C, \mu^* > 0$ are constants and $\widehat{\mathbf{S}}_{m,\ell}$ is the estimate from (5.9). Then define the event

$$\mathcal{F}' := \bigcap_{\ell=1}^{\infty} \mathcal{F}'_\ell. \tag{D.18}$$

We now prove the following lemmas to show the good event $\mathcal{F}'_\ell$ holds with probability $(1 - \delta_\ell)$. Before proving the concentration of $\widehat{\mathbf{S}}_{m,\ell}$ we first need to show that $\sigma_{\min}(\sum_{\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}} \mathbf{b}_{\widetilde{\mathbf{w}}} \widetilde{\mathbf{w}} \widetilde{\mathbf{w}}^\top) > 0$. If this holds true then we can sample following E-optimal design.

Lemma D.22. *For any phase $\ell > 0$ and task $m \in [M]$, let $\left\| \widehat{B}_{1,\ell}^\top \mathbf{B}_1^\perp \right\| \leqslant c_1$ and $\left\| \widehat{B}_{2,\ell}^\top \mathbf{B}_2^\perp \right\| \leqslant c_2$, for some $c_1, c_2 > 0$. Then we have*

$$\sigma_{\min}\Big(\sum_{\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}} \mathbf{b}_{\widetilde{\mathbf{w}}} \widetilde{\mathbf{w}} \widetilde{\mathbf{w}}^\top \Big) > 0$$

Proof. We can show that

$$\sum_{\widetilde{\mathbf{w}} \in \widetilde{\mathcal{W}}} \mathbf{b}_{\widetilde{\mathbf{w}}} \widetilde{\mathbf{w}} \widetilde{\mathbf{w}}^\top \overset{(a)}{=} \sum_{\mathbf{x} \in \mathcal{X}_m, \mathbf{z} \in \mathcal{Z}_m} \mathbf{b}_{x,z} \underbrace{\widehat{\mathbf{U}}_\ell^\top \mathbf{x}_m}_{\widetilde{\mathbf{g}}_m} \underbrace{\mathbf{z}_m \widehat{\mathbf{V}}_\ell^\top}_{\widetilde{\mathbf{v}}_m^\top}$$

where, in (a) the $\mathbf{b}_{x,z}$ is the sampling proportion for the arms $\mathbf{x}$ and $\mathbf{z}$ (they are allocated the same proportion, as they are pulled the same number of times). Also note that from Theorem D.18 we know that $\left\| \widehat{B}_{1,\ell}^\top \mathbf{B}_1^\perp \right\| \leqslant c_1$ and $\left\| \widehat{B}_{2,\ell}^\top \mathbf{B}_2^\perp \right\| \leqslant c_2$ for some $c_1, c_2 > 0$ holds with high probability. This helps us to apply Theorem D.23 to get the claim of the lemma. □

Lemma D.23. (*Restatement of Lemma C.5 from Du et al. (2023)*) *For any phase* $\ell > 0$ *and task* $m \in [M]$, *if* $\left\|\widehat{U}_\ell^\top \overline{U}^\perp\right\| \leqslant c$ *for some* $c > 0$, *then we have*

$$\sigma_{\min}\left(\sum_{i=1}^{n} \mathbf{b}_m^*(\mathbf{x}_i)\, \widehat{\mathbf{U}}_\ell^\top \mathbf{x}_i \mathbf{x}_i^\top \widehat{\mathbf{U}}_\ell\right) > 0$$

where $\mathbf{b}_m^*$ *is a sampling proportion on* $\mathbf{x}$.

Lemma D.24. *Let* $L' : \mathbb{R}^{k_1 \times k_2} \to \mathbb{R}$ *is the loss function defined in* (5.9). *Then by setting*

$$t = \sqrt{2k_1k_2C\,(4+S_0^2)\log\left(\frac{2\,(k_1+k_2)}{\delta_\ell}\right)},$$

$$\nu = \frac{t}{(4+S_0)\,Ck_1k_2\sqrt{\tau_\ell^E}} = \sqrt{\frac{2\log\left(\frac{2(k_1+k_2)}{\delta_\ell}\right)}{\tau_\ell^E k_1 k_2 C\,(4+S_0^2)}},$$

we have with probability at least $1-\delta_\ell$, *it holds that*

$$\mathbb{P}\left(\left\|\nabla L'\left(\mu^* \mathbf{S}_{m,*}\right)\right\|_{op} \geqslant \frac{2t}{\sqrt{\tau_\ell^E}}\right) \leqslant \delta_\ell,$$

where $\mu^* = \mathbb{E}\left[\langle \mathbf{X}_m, \mathbf{S}_{m,*}\rangle\right] > 0$, *and* $\mathbf{X}_m = \widetilde{\mathbf{g}}_m \widetilde{\mathbf{v}}_m^\top$.

Proof. Let $\mathbf{X}_{m,i} = \widetilde{\mathbf{g}}_{m,i}\widetilde{\mathbf{v}}_{m,i}^\top$ Based on the definition of our loss function $L'(\cdot)$ in (5.9), we have that

$$\begin{aligned}\nabla_{x_m} L'\left(\mathbf{S}_{m,*}\right) &= \mu^* \mathbf{S}_{m,*} - \frac{2}{\tau_\ell^E}\sum_{i=1}^{\tau_\ell^E} \widetilde{\psi}_\nu(r_{m,i}\cdot Q(x_m)) \\ &\overset{(a)}{=} \left[\mathbb{E}\left(r_{m,1}\cdot Q\left(\mathbf{X}_{m,1}\right)\right) - \frac{1}{\tau_\ell^E}\sum_{i=1}^{\tau_\ell^E} \widetilde{\psi}_\nu\left(r_{m,i}\cdot Q\left(\mathbf{X}_{m,i}\right)\right)\right]\end{aligned}$$

where (a) follows using the same steps as in Theorem D.17. Similarly,

using the same steps for a single task as in Theorem D.17 we have

$$\mathbb{P}\left(\left\|\nabla L'\left(\mu^{*}\mathbf{S}_{m,*}\right)\right\|_{op} \geqslant \frac{2t}{\sqrt{\tau_{\ell}^{E}}}\right) \leqslant 2\left(k_1+k_2\right)\exp\left(-\nu t\sqrt{\tau_{\ell}^{E}}+\frac{\nu^{2}\left(4+S_0^{2}\right)Ck_1k_2\tau_{\ell}^{E}}{2}\right)$$

By plugging the values of t and ν in Theorem D.17, we finish the proof. □

Lemma D.25. (***Concentration of*** $\widehat{\mathbf{S}}_{m,\ell}$) *For any low-rank linear model with samples* $\mathbf{X}_1 \ldots, \mathbf{X}_{\tau_{\ell}^{E}}$ *drawn from* $\mathcal{X}$ *according to* $\mathcal{D}$ *then for the optimal solution to the nuclear norm regularization problem in* (5.2) *with*

$$\nu = \sqrt{2\log\left(2\left(k_1+k_2\right)/\delta_{\ell}\right)/\left(\left(4+S_0^{2}\right)\tau_{\ell}^{E}k_1k_2\right)}$$

and

$$\gamma_{m,\ell} = 4\sqrt{\frac{2\left(4+S_0^{2}\right)Ck_1k_2\log\left(2\left(k_1+k_2\right)/\delta_{\ell}\right)}{\tau_{m,\ell}^{E}}},$$

with probability at least $1-\delta_{\ell}$ *it holds that:*

$$\left\|\widehat{\mathbf{S}}_{m,\ell}-\mu^{*}\mathbf{S}_{m,*}\right\|_{F}^{2} \leqslant \frac{C_1k_1k_2r\log\left(\frac{2(k_1+k_2)}{\delta_{\ell}}\right)}{\tau_{m,\ell}^{E}},$$

for $C_1 = 36\left(4+S_0^{2}\right)C$, $\|\mathbf{X}\|_F, \|\mathbf{S}_{m,*}\|_F \leqslant S_0$, *some nonzero constant* μ^{*}, *and* $\mathbb{E}\left[\left(S^{p}(\mathbf{X})\right)_{ij}^{2}\right] \leqslant C, \forall i,j$. *Summing over all phases* $\ell \geqslant 1$ *it follows that* $\mathbb{P}(\mathcal{F}') \geqslant 1-\delta/2$.

Proof. Since the estimator $\widehat{\mathbf{S}}_{m,\ell}$ minimizes the regularized loss function defined in Eqn. (6), we have

$$L(\widehat{\mathbf{S}}_{m,\ell})+\gamma_{\ell}\|\widehat{\mathbf{S}}_{m,\ell}\|_{nuc} \leqslant L\left(\mu^{*}\mathbf{S}_{m,*}\right)+\gamma_{\ell}\left\|\mu^{*}\mathbf{S}_{m,*}\right\|_{nuc}.$$

And due to the fact that $L'(\cdot)$ is a quadratic function, we have the following

expression based on multivariate Taylor's expansion:

$$L'(\widehat{\mathbf{S}}_{m,\ell}) - L'(\mu^* \mathbf{S}_{m,*}) = \langle \nabla L'(\mu^* \mathbf{S}_{m,*}), \boldsymbol{\Theta} \rangle + 2\|\boldsymbol{\Theta}\|_F^2, \quad \text{where } \boldsymbol{\Theta} = \widehat{\mathbf{S}}_{m,\ell} - \mu^* \mathbf{S}_{m,*}.$$

By rearranging the above two results, we can deduce that

$$\begin{aligned} 2\|\boldsymbol{\Theta}\|_F^2 &\leqslant -\langle \nabla L'(\mu^* \mathbf{S}_{m,*}), \boldsymbol{\Theta} \rangle + \gamma_\ell \|\mu^* \mathbf{S}_{m,*}\|_{nuc} - \gamma_\ell \|\widehat{\mathbf{S}}_{m,*}\|_{nuc} \\ &\overset{(i)}{\leqslant} \|\nabla L'(\mu^* \mathbf{S}_{m,*})\|_{op} \|\boldsymbol{\Theta}\|_{nuc} + \gamma_\ell \|\mu^* \mathbf{S}_{m,*}\|_{nuc} - \gamma_\ell \|\widehat{\mathbf{S}}_{m,\ell}\|_{nuc}, \end{aligned} \tag{D.19}$$

where (i) comes from the duality between matrix operator norm and nuclear norm. Next, we represent the saturated SVD of $\mathbf{S}_{m,*}$ as $\mathbf{S}_{m,*} = \overline{U} \mathbf{D} \mathbf{V}^\top$ where $\overline{U} \in \mathbb{R}^{k_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{k_2 \times r}$, and here we would work on its full version, i.e.

$$\mathbf{S}_{m,*} = (\overline{U}, \overline{U}_\perp) \begin{pmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{pmatrix} (\mathbf{V}, \mathbf{V}_\perp)^\top = (\overline{U}, \overline{U}_\perp) \mathbf{D}^* (\mathbf{V}, \mathbf{V}_\perp)^\top$$

where we have $\overline{U}_\perp \in \mathbb{R}^{k_1 \times (k_1 - r)}$, $\mathbf{D}^* \in \mathbb{R}^{k_1 \times k_2}$ and $\mathbf{V}_\perp \in \mathbb{R}^{k_2 \times (k_2 - r)}$. Furthermore, we define

$$\boldsymbol{\Lambda} = (\overline{U}, \overline{U}_\perp)^\top \boldsymbol{\Theta} (\mathbf{V}, \mathbf{V}_\perp) = \begin{pmatrix} \overline{U}^\top \boldsymbol{\Theta} \mathbf{V} & \overline{U}^\top \boldsymbol{\Theta} \mathbf{V}_\perp \\ \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V} & \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V}_\perp \end{pmatrix} = \boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2$$

where we write

$$\boldsymbol{\Lambda}_1 = \begin{pmatrix} 0 & 0 \\ 0 & \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V}_\perp \end{pmatrix}, \quad \boldsymbol{\Lambda}_2 = \begin{pmatrix} \overline{U}^\top \boldsymbol{\Theta} \mathbf{V} & \overline{U}^\top \boldsymbol{\Theta} \mathbf{V}_\perp \\ \overline{U}_\perp^\top \boldsymbol{\Theta} \mathbf{V} & 0 \end{pmatrix}.$$

Afterward, it holds that

$$\begin{aligned}\|\widehat{\mathbf{S}}_\ell\|_{\text{nuc}} = \|\mu^*\mathbf{S}_{m,*} + \boldsymbol{\Theta}\|_{\text{nuc}} &\overset{(a)}{=} \left\|\left(\overline{U}, \overline{U}_\perp\right)\left(\mu^*\mathbf{D}^* + \boldsymbol{\Lambda}\right)\left(\mathbf{V}, \mathbf{V}_\perp\right)^\top\right\|_{\text{nuc}} \\ &\overset{(b)}{=} \|\mu^*\mathbf{D}^* + \boldsymbol{\Lambda}\|_{\text{nuc}} + \|\mu^*\mathbf{D}^* + \boldsymbol{\Lambda}_1 + \boldsymbol{\Lambda}_2\|_{\text{nuc}} \\ &\geqslant \|\mu^*\mathbf{D}^* + \boldsymbol{\Lambda}_1\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} \\ &= \|\mu^*\mathbf{D}\|_{\text{nuc}} + \|\boldsymbol{\Lambda}_1\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} \\ &= \|\mu^*\mathbf{S}_{m,*}\|_{\text{nuc}} + \|\boldsymbol{\Lambda}_1\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_2\|_{\text{nuc}}, \end{aligned} \tag{D.20}$$

where, (a) follows from the definition of $\mathbf{S}_{m,*}$, and (b) follows the definition of $\boldsymbol{\Lambda}$. This implies that

$$\|\mu^*\mathbf{S}_{m,*}\|_{\text{nuc}} - \|\widehat{\mathbf{S}}_{m,\ell}\|_{\text{nuc}} \leqslant \|\boldsymbol{\Lambda}_2\|_{\text{nuc}} - \|\boldsymbol{\Lambda}_1\|_{\text{nuc}}.$$

Combining (D.19) and (D.20), we have that

$$2\|\boldsymbol{\Theta}\|_F^2 \leqslant \left(\|\nabla L\left(\mu^*\mathbf{B}\right)\|_{\text{op}} + \gamma_\ell\right)\|\boldsymbol{\Lambda}_2\|_{\text{nuc}} + \left(\|\nabla L\left(\mu^*\mathbf{B}\right)\|_{\text{op}} - \gamma_\ell\right)\|\boldsymbol{\Lambda}_1\|_{\text{nuc}}.$$

Then, we refer to the setting in our Theorem D.17, and we choose $\gamma_\ell = 4t/\sqrt{M\tau_\ell^E}$ where the value of t is determined in Theorem D.17, i.e.

$$\gamma_\ell = 4\sqrt{\frac{2\left(4 + S_0^2\right)Ck_1k_2\log\left(2\left(d_1 + d_2\right)/\delta_\ell\right)}{M\tau_\ell^E}},$$

we know that $\lambda_{T-1} \geqslant 2\left\|\nabla L\left(\mu^*\mathbf{S}_{m,*}\right)\right\|_{\text{op}}$ with probability at least $1 - \delta_\ell$ for any $\delta_\ell \in (0,1)$. Therefore, with a probability at least $1 - \delta_\ell$, we have

$$2\|\boldsymbol{\Theta}\|_F^2 \leqslant \frac{3}{2}\gamma_\ell\|\boldsymbol{\Lambda}_2\|_{\text{nuc}} - \frac{1}{2}\gamma_\ell\|\boldsymbol{\Lambda}_1\|_{\text{nuc}} \leqslant \frac{3}{2}\gamma_\ell\|\boldsymbol{\Lambda}_2\|_{\text{nuc}}.$$

Since we can easily verify that the rank of $\boldsymbol{\Lambda}_2$ is at most $2r$, and by using

Cauchy-Schwarz Inequality we have that

$$2\|\boldsymbol{\Theta}\|_{\mathsf{F}}^2 \leqslant \frac{3}{2}\gamma_\ell\sqrt{2r}\,\|\boldsymbol{\Lambda}_2\|_{\mathsf{F}} \leqslant \frac{3}{2}\gamma_\ell\sqrt{2r}\|\boldsymbol{\Lambda}\|_{\mathsf{F}} = \frac{3}{2}\gamma_\ell\sqrt{2r}\|\boldsymbol{\Theta}\|_{\mathsf{F}}$$

which implies that

$$\|\boldsymbol{\Theta}\|_{\mathsf{F}} \leqslant \frac{3}{4}\sqrt{2r}\gamma_\ell = 6\sqrt{\frac{(4+S_0^2)\,Ck_1k_2r\log\left(\frac{2(k_1+k_2)}{\delta_\ell}\right)}{\tau_\ell^{\mathsf{E}}}}$$

This implies that $\mathbb{P}(\mathcal{F}_\ell') \geqslant 1-\delta_\ell$. Taking a union bound over all phases $\ell \geqslant 1$ and recalling $\delta_\ell := \frac{\delta}{2\ell^2}$, we obtain

$$\begin{aligned}
\mathbb{P}(\mathcal{F}') &\geqslant 1 - \sum_{\ell=1}^{\infty}\mathbb{P}\left((\mathcal{F}')_\ell^{\mathsf{c}}\right) \\
&\geqslant 1 - \sum_{\ell=1}^{\infty}\frac{\delta_\ell}{2} \\
&= 1 - \sum_{\ell=1}^{\infty}\frac{\delta}{4\ell^2} \\
&\geqslant 1 - \frac{\delta}{2}.
\end{aligned}$$

This concludes our proof. □

Final Sample Complexity Bound

We first define the arm elimination event similar to Theorem 1. For any $\mathbb{V} \subseteq \underline{\mathcal{W}}$ be the active set and $\underline{\mathbf{w}} \in \mathbb{V}$ define

$$\mathcal{E}_{\underline{\mathbf{w}},\ell}(\mathbb{V}) = \left\{\left|\left\langle \underline{\mathbf{w}} - \underline{\mathbf{w}}^\star, \widehat{\theta}_\ell(\mathbb{V}) - \theta^*\right\rangle\right| \leqslant \epsilon_\ell\right\} \tag{D.21}$$

where it is implicit that $\widehat{\theta}_\ell := \widehat{\theta}_\ell(\mathbb{V})$ is the design constructed in the algorithm at stage ℓ with respect to $\underline{\mathcal{W}}_\ell = \mathbb{V}$.

Theorem 2. (***Restatement***) *With probability at least* $1-\delta$*, multi-task GOBLIN returns the best arms* $\mathbf{x}_*$, $\mathbf{z}_*$*, and the number of samples used is bounded by*

$$\widetilde{O}\left(\frac{M(k_1+k_2)r}{\Delta^2}+\frac{M\sqrt{k_1k_2r}}{S_r}+\frac{\sqrt{d_1d_2r}}{S_r}\right).$$

Proof. For the rest of the proof we have that the good events $\mathcal{F}_\ell \bigcap \mathcal{F}'_\ell \bigcap \mathcal{E}_{\underline{\mathbf{w}},\ell}(\underline{\mathcal{W}}_\ell)$ holds true for each phase ℓ with probability greater than $(1-\delta)$. The three events are defined in (D.13), (D.17) and (D.21).

Third Stage: Define $\underline{\mathcal{A}}_{m,\ell} = \{\underline{\mathbf{w}} \in \underline{\mathcal{W}}_\ell : \langle \underline{\mathbf{w}}^\star - \underline{\mathbf{w}}, \theta^* \rangle \leqslant 4\epsilon_{m,\ell}\}$. Note that by assumption $\underline{\mathcal{W}} = \underline{\mathcal{W}}_1 = \underline{\mathcal{A}}_1$. The above lemma implies that with probability at least $1-\delta$ we have $\bigcap_{\ell=1}^{\infty}\left\{\underline{\mathcal{W}}_{m,\ell} \subseteq S_{m,\ell}\right\}$. This implies that

$$\begin{aligned}
\rho^G\left(\underline{\mathcal{W}}_{m,\ell}\right) &= \min_{\mathbf{b}\in\Delta_{\underline{\mathcal{W}}_m}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{W}}_{m,\ell}} \|\underline{\mathbf{w}}-\underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}} \\
&\leqslant \min_{\mathbf{b}\in\Delta_{\underline{\mathcal{W}}_m}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in S_{m,\ell}} \|\underline{\mathbf{w}}-\underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\boldsymbol{\Lambda}\right)^{-1}} \\
&= \rho^G\left(\underline{\mathcal{A}}_{m,\ell}\right).
\end{aligned}$$

Let the effective dimension be $k = (k_1+k_2)r$. Define $k_1^\ell = 8k\log(1+\tau^G_{m,\ell-1}/\lambda)$. For $\ell \geqslant \left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil$ we have that $S_{m,\ell} = \{\underline{\mathbf{w}}^\star\}$, thus, the

sample complexity to identify $\underline{\mathbf{w}}_{\mathrm{m}}^{\star}$ is equal to

$$\sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} \sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_{\mathrm{m}}} \left\lceil \tau_{\mathrm{m},\ell}^{\mathrm{G}} \widehat{b}_{\mathrm{m},\ell,\underline{\mathbf{w}}}^{\mathrm{G}} \right\rceil = \sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} \left(\frac{(k_1^{\ell}+1)k_1^{\ell}}{2} + \tau_{\mathrm{m},\ell}^{\mathrm{G}} \right)$$

$$= \sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} \left(\frac{(k_1^{\ell}+1)k_1^{\ell}}{2} + 2\epsilon_{\mathrm{m},\ell}^{-2} \rho^{\mathrm{G}}(\underline{\mathcal{W}}_{\mathrm{m},\ell}) B_{\mathrm{m},*}^{\ell} \log\left(4k_1^{\ell}\ell^2|\underline{\mathcal{W}}_{\mathrm{m}}|/\delta\right) \right)$$

$$\overset{(a)}{\leqslant} 2 \sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} \left(\frac{(k+1)k}{2} \log^2(1+\tau_{\mathrm{m},\ell-1}^{\mathrm{G}}) + 2\epsilon_{\mathrm{m},\ell}^{-2} \rho^{\mathrm{G}}(\underline{\mathcal{W}}_{\mathrm{m},\ell}) B_{\mathrm{m},*}^{\ell} \log\left(4k_1^{\ell}\ell^2|\underline{\mathcal{W}}_{\mathrm{m}}|/\delta\right) \right)$$

$$\overset{(b)}{\leqslant} 2\frac{(k+1)k}{2} \sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} \left(\log^2(1+\tau_{\mathrm{m},\ell-1}^{\mathrm{G}}) + 8\epsilon_{\mathrm{m},\ell}^{-2} \rho^{\mathrm{G}}(\underline{\mathcal{W}}_{\mathrm{m},\ell}) B_{\mathrm{m},*}^{\ell} \log\left(4k_1^{\ell}\ell^2|\underline{\mathcal{W}}_{\mathrm{m}}|/\delta\right) \right)$$

$$\overset{(c)}{\leqslant} (k+1)k \sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} \left(1 + 16\epsilon_{\mathrm{m},\ell}^{-2} \rho^{\mathrm{G}}(\underline{\mathcal{W}}_{\mathrm{m},\ell}) B_{\mathrm{m},*}^{\ell} \log^2\left(4k_1^{\ell}\ell^2|\underline{\mathcal{W}}_{\mathrm{m}}|/\delta\right) \right)$$

$$\overset{(d)}{\leqslant} (k+1)k \left\lceil \log_2\left(4\Delta_{\mathrm{m}}^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta_{\mathrm{m}}^{-1})\rceil} 32\epsilon_{\mathrm{m},\ell}^{-2} \mathrm{f}\left(\underline{\mathcal{A}}_{\mathrm{m},\ell}\right) B_{\mathrm{m},*}^{\ell} \log\left(4k\ell^2|\underline{\mathcal{W}}_{\mathrm{m}}|/\delta\right)$$

$$\overset{(e)}{\leqslant} (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_{\ell}^{-2} \mathrm{f}\left(\underline{\mathcal{A}}_{\mathrm{m},\ell}\right) \left(64\lambda S^2 + 64\tau_{\mathrm{m},\ell-1}^{\mathrm{G}}\right) \log\left(4k\ell^2|\underline{\mathcal{W}}|/\delta\right)$$

$$= (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_{\ell}^{-2} \mathrm{f}\left(\underline{\mathcal{A}}_{\mathrm{m},\ell}\right) (64\lambda S^2) \log\left(4k\ell^2|\underline{\mathcal{W}}|/\delta\right)$$

$$+ (k+1)k \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} 32\epsilon_{\ell}^{-2} \mathrm{f}\left(\underline{\mathcal{A}}_{\mathrm{m},\ell}\right) (64\tau_{\mathrm{m},\ell-1}^{\mathrm{G}}) \log\left(4k\ell^2|\underline{\mathcal{W}}|/\delta\right)$$

$$\overset{(f)}{\leqslant} (k+1)k\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil + 2048\lambda S^2 \log\left(\frac{4k\log_2^2\left(8\Delta_m^{-1}\right)|\underline{\mathcal{W}}_m|}{\delta}\right) \sum_{\ell=1}^{\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} 2^{2\ell} f\left(\underline{\mathcal{A}}_{m,\ell}\right).$$

where, (a) follows as $\log^2(1+\tau^G_{m,\ell-1}/\lambda) \leqslant \log^2(1+\tau^G_{m,\ell-1})$, (b) follows by noting that $\log(x\log(1+x)) \leqslant 2\log(x)$ for any $x > 1$. The (c) follows by subsuming the $\log^2(1+\tau^G_{m,\ell-1})$ into $2\tau^G_\ell$. The (d) follows as $\log(1+\tau^G_{m,\ell-1}) < \tau^G_\ell$ which enables us to replace the k_1^ℓ inside the log with an addtional factor of 2. The (e) follows similarly to (D.11) by noting that

$$\begin{aligned} B^\ell_{m,*} &\leqslant 64(\sqrt{\lambda}S + \sqrt{\lambda^\perp_{m,\ell}}S^\perp_{m,\ell}) \\ &\leqslant 64\lambda S^2 + \left(\frac{64\tau^G_{m,\ell-1}}{8(d_1+d_2)r\log(1+\frac{\tau^G_{m,\ell-1}}{\lambda})}\right)\cdot\left(\frac{8d_1d_2r}{\tau^E_\ell S_r^2}\log\left(\frac{d_1+d_2}{\delta_\ell}\right)\right) \\ &\overset{(a_1)}{\leqslant} 64\lambda S^2 + 64\tau^G_{m,\ell-1}. \end{aligned}$$

Finally the (f) follows by subsuming the $\tau^G_{\ell-1}$ with a factor of 2 into the quantity of τ^G_ℓ. Then it follows that

$$\begin{aligned} \rho^G_{m,*} &= \inf_{b\in\triangle_{\underline{\mathcal{W}}_m}} \max_{\underline{w}\in\underline{\mathcal{W}}_m} \frac{\|\underline{w}-\underline{w}^\star\|^2_{\left(\sum_{\underline{w}\in\underline{\mathcal{W}}_m} b_{\underline{w}}\underline{w}\,\underline{w}^\top+\Lambda\right)^{-1}}}{\left(\langle \underline{w}-\underline{w}^\star,\theta^*\rangle\right)^2} \\ &= \inf_{b\in\triangle_{\underline{\mathcal{W}}_m}} \max_{\ell\leqslant\left\lceil\log_2\left(4\Delta_m^{-1}\right)\right\rceil} \max_{\underline{w}\in S_{m,\ell}} \frac{\|\underline{w}-\underline{w}^\star\|^2_{\left(\sum_{\underline{w}\in\underline{\mathcal{W}}_m} b_{\underline{w}}\underline{w}\,\underline{w}^\top+\Lambda\right)^{-1}}}{\left(\langle \underline{w}-\underline{w}^\star,\theta^*\rangle\right)^2} \\ &\geqslant \frac{1}{\left\lceil\log_2\left(4\Delta_m^{-1}\right)\right\rceil} \inf_{b\in\triangle_{\underline{\mathcal{W}}_m}} \sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta_m^{-1}\right)\right\rceil} \max_{\underline{w}\in\underline{\mathcal{A}}_{m,\ell}} \frac{\|\underline{w}-\underline{w}^\star\|^2_{\left(\sum_{\underline{w}\in\underline{\mathcal{W}}_m} b_{\underline{w}}\underline{w}\,\underline{w}^\top+\Lambda\right)^{-1}}}{\left(\langle \underline{w}-\underline{w}^\star,\theta^*\rangle\right)^2} \\ &\geqslant \frac{1}{16\left\lceil\log_2\left(4\Delta_m^{-1}\right)\right\rceil} \sum_{\ell=1}^{\left\lceil\log_2\left(4\Delta_m^{-1}\right)\right\rceil} 2^{2\ell} \inf_{b\in\triangle_{\underline{\mathcal{W}}_m}} \max_{\underline{w}\in\underline{\mathcal{A}}_{m,\ell}} \|\underline{w}-\underline{w}^\star\|^2_{\left(\sum_{\underline{w}\in\underline{\mathcal{W}}_m} b_{\underline{w}}\underline{w}\,\underline{w}^\top+\Lambda\right)^{-1}} \end{aligned}$$

$$\geqslant \frac{1}{64\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} \sum_{\ell=1}^{\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} 2^{2\ell} \inf_{\mathbf{b}\in\triangle_{\underline{\mathcal{W}}_m}} \max_{\underline{\mathbf{w}},\underline{\mathbf{w}}'\in\underline{\mathcal{A}}_{m,\ell}} \|\underline{\mathbf{w}}-\underline{\mathbf{w}}'\|^2_{\left(\sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_m} \mathbf{b}_{\underline{\mathbf{w}}}\underline{\mathbf{w}}\,\underline{\mathbf{w}}^\top+\mathbf{\Lambda}\right)^{-1}}$$

$$\geqslant \frac{1}{64\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} \sum_{\ell=1}^{\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} 2^{2\ell} f\left(\underline{\mathcal{A}}_{m,\ell}\right).$$

This implies that

$$\sum_{\ell=1}^{\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} 2^{2\ell} f\left(\underline{\mathcal{A}}_{m,\ell}\right) \leqslant \rho^G_{m,*} 64\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil.$$

Plugging this back we get

$$\begin{aligned}
&\sum_{\ell=1}^{\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil} \sum_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}_m} \left\lceil \tau^G_{m,\ell}\widehat{\mathbf{b}}_{\ell,\underline{\mathbf{w}}}\right\rceil \\
&\leqslant (k+1)k\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil \\
&\qquad + 2048\lambda S^2 \log\left(\frac{8k\log_2^2\left(8\Delta_m^{-1}\right)|\underline{\mathcal{W}}_m|}{\delta}\right) 64\rho^G_{m,*}\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil \\
&\leqslant (k+1)k\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil \\
&\qquad + C_2\lambda S^2 \log\left(\frac{8k\log_2^2\left(8\Delta_m^{-1}\right)|\underline{\mathcal{W}}_m|}{\delta}\right) \rho^G_{m,*}\left\lceil \log_2\left(4\Delta_m^{-1}\right)\right\rceil
\end{aligned}$$

where, $C_2 > 0$ is a constant. Summing over each task m, the simplified sample complexity for the third stage is given by

$$N_3 \leqslant O\left(\frac{Mk}{\Delta^2}\log\left(\frac{k\log_2\left(\Delta^{-1}\right)|\underline{\mathcal{W}}_m|}{\delta}\right)\right) = \widetilde{O}\left(\frac{M(k_1+k_2)r}{\Delta^2}\right)$$

where $\Delta = \min_{\underline{\mathbf{w}}\in\underline{\mathcal{W}}}(\underline{\mathbf{w}}_*-\underline{\mathbf{w}})^\top\theta_* \overset{(a1)}{=} \min_{\mathbf{x}\in\mathcal{X}\setminus\{\mathbf{x}_*\},\mathbf{z}\in\mathcal{Z}\setminus\{\mathbf{z}_*\}}(\mathbf{x}_*^\top\mathbf{\Theta}_*\mathbf{z}_*-\mathbf{x}^\top\mathbf{\Theta}_*\mathbf{z})$. The $(a1)$ follows by reshaping the arms in $\underline{\mathcal{W}}$ to recover the arms in $\mathcal{X}$ and

$\mathcal{Z}$.

2nd Stage: Again recall that the E-optimal design in stage 2 of Algorithm 6 satisfies the Assumption 12 as the sample distribution $\mathcal{D}$ has finite second order moments.

For the second stage first observe that by plugging in the definition of $\widetilde{\tau}_E^{\ell}$ we get

$$\begin{aligned}\left\|\theta^*_{m,k+1:p}\right\|_2^2 = \sum_{i>r\wedge j>r} H_{ij}^2 &= \left\|(\widehat{\overline{U}}_\ell^{\perp})^\top \left(\overline{U}^* S^* V^{*\top}\right) \widehat{V}_\ell^{\perp}\right\|_F^2 \\ &\leqslant \left\|(\widehat{\overline{U}}_\ell^{\perp})^\top \overline{U}^*\right\|_F^2 \|S^*\|_2^2 \left\|(\widehat{V}_\ell^{\perp})^\top V^*\right\|_F^2 \leqslant O\left(\frac{k_1 k_2 r}{\tau_\ell^E} \log\left(\frac{k_1+k_2}{\delta}\right)\right) \\ &= O\left(\frac{\sqrt{d_1 d_2 r}}{S_r} \log\left(\frac{d_1+d_2}{\delta_\ell}\right)\right),\end{aligned}$$

which implies $\left\|\theta^*_{k+1:p}\right\|_2 = \widetilde{O}\left(\sqrt{k_1 k_2 r}/S_r\right)$. We also set $\frac{8k_1k_2r}{\tau^E_{m,\ell} S_r^2} \log\left(\frac{k_1+k_2}{\delta_\ell}\right) := S^{\perp}_{m,\ell}$. Now we bound the sample complexity from the second stage. From the second stage we can show that we have for the arm set $\widetilde{\mathcal{W}}_m$

$$\begin{aligned}N_2 &= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \sum_{\tilde{w}\in\widetilde{\mathcal{W}}_m} \left\lceil \widetilde{\tau}^E_{m,\ell} \widehat{b}^E_{m,\ell,\tilde{w}} \right\rceil \\ &= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(\frac{(p+1)p}{2} + \widetilde{\tau}^E_{m,\ell}\right) \\ &= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1})\rceil} \left(\frac{(p+1)p}{2} + \frac{\sqrt{8k_1k_2r\log\left(4\ell^2|\mathcal{W}|/\delta\right)}}{S_r}\right) \\ &\leqslant (p+1)p\left\lceil \log_2\left(4\Delta^{-1}\right)\right\rceil + 32\frac{\sqrt{k_1k_2r}}{S_r}\log\left(\frac{4\log_2^2\left(8\Delta^{-1}\right)|\mathcal{W}|}{\delta}\right)\left\lceil \log_2\left(4\Delta^{-1}\right)\right\rceil \\ &= O\left(\frac{\sqrt{k_1k_2r}}{S_r}\log\left(\frac{4\log_2^2\left(8\Delta^{-1}\right)|\mathcal{W}|}{\delta}\right)\right) \overset{(a)}{=} \widetilde{O}\left(\frac{\sqrt{k_1k_2r}}{S_r}\right)\end{aligned}$$

1st Stage: Finally we also use the E-optimal design in first stage of Algorithm 6. Note that this design satisfies the Assumption 12 as the sample distribution $\mathcal{D}$ has finite second-order moments. Now we bound the sample complexity from the first stage. From the first stage we can show that we have for the arm set $\overline{\mathcal{W}}$

$$
\begin{aligned}
N_1 &= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1}) \rceil} \sum_{m=1}^{M} \sum_{\overline{\mathbf{w}} \in \overline{\mathcal{W}}_m} \left\lceil \tau_\ell^E \widehat{b}_{m,\ell,\overline{\mathbf{w}}}^E \right\rceil \\
&= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1}) \rceil} \left(\frac{M(p+1)p}{2} + \tau_\ell^E \right) \\
&= \sum_{\ell=1}^{\lceil \log_2(4\Delta^{-1}) \rceil} \left(\frac{M(p+1)p}{2} + \frac{\sqrt{8 d_1 d_2 r \log\left(4\ell^2 |\mathcal{W}|/\delta\right)}}{S_r} \right) \\
&\leqslant M(p+1)p \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil + 32 \frac{\sqrt{d_1 d_2 r}}{S_r} \log\left(\frac{4 \log_2^2\left(8\Delta^{-1}\right) |\mathcal{W}|}{\delta} \right) \left\lceil \log_2\left(4\Delta^{-1}\right) \right\rceil \\
&\overset{(a)}{=} O\left(\frac{\sqrt{d_1 d_2 r}}{S_r} \log\left(\frac{4 \log_2^2\left(8\Delta^{-1}\right) |\mathcal{W}|}{\delta} \right) \right) \overset{(a)}{=} \widetilde{O}\left(\frac{\sqrt{d_1 d_2 r}}{S_r} \right)
\end{aligned}
$$

where, (a) follows as $p = d_1 d_2$. Combining N_1, N_2 and N_3 gives the claim of the theorem. □

D.6 Additional Experimental Details

Single Task Unit Ball: This experiment consists of a set of $\{6, 10, 14\}$ left and right arms that are arranged in a unit ball in $\mathbb{R}^6$, and $\|\mathbf{x}\| = 1$, $\|\mathbf{z}\| = 1$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$. Hence, we have $d_1 \in \mathbb{R}^6$ and $d_2 \in \mathbb{R}^6$. We choose a random $\mathbf{\Theta}_* \in \mathbb{R}^{d_1 \times d_2}$ which has rank $r = 2$. We set $\delta = 0.1$. We compare against RAGE (Fiez et al., 2019) that treats this $d_1 d_2$ bilinear bandit as a linear bandit setting and suffers a sample complexity that scales as $\widetilde{O}(d_1 d_2 / \Delta^2)$. We do a continuous relaxation of the algorithm

when implementing it to make this more tractable.

Multi-task Unit Ball: This experiment consists of a set of $\{5, 10, 15, 20, 25, 30\}$ tasks. For each task, we choose left and right arms that are arranged in a unit ball in $\mathbb{R}^8$, and $\|\mathbf{x}\| = 1$, $\|\mathbf{z}\| = 1$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{z} \in \mathcal{Z}$. Hence, we have $d_1 \in \mathbb{R}^8$ and $d_2 \in \mathbb{R}^8$. We choose $k_1 = k_2 = 4$, and feature extractors $\mathbf{B}_1 \in \mathbb{R}^{d_1 \times k_1}$, $\mathbf{B}_2 \in \mathbb{R}^{d_2 \times k_2}$ shared across tasks. We choose a random matrix $\mathbf{S}_{m,*} \in \mathbb{R}^{k_1 \times k_2}$ for each task m such that $\mathbf{S}_{m,*}$ has rank $r = 2$. We set $\delta = 0.1$. We compare against DouExpDes (Du et al., 2023) that treats this setting as M $k_1 k_2$ bilinear bandits (after learning the feature extractors) and suffers a sample complexity that scales as $\widetilde{O}(Mk_1k_2/\Delta^2)$ (see Theorem D.15). Again we do a continuous relaxation of the algorithm when implementing it to make this more tractable.

D.7 Table of Notations

Notations	Definition
$\mathcal{X}$	Left arm set
$\mathcal{Z}$	Right arm set
M	Number of tasks
ℓ	Phase number
$\mathbf{\Theta}_{m,*}$	Hidden parameter matrix for
$\mathbf{b}_{\ell}^{E}$	E-optimal design at the ℓ-th phase
$\mathbf{b}_{m,\ell}^{G}$	G-optimal design at the ℓ-th phase for the m-th task
$S_{m,\ell}^{\perp}$	$\frac{8d_1d_2r}{\tau_{\ell}^{E}S_r^2}\log\left(\frac{d_1+d_2}{\delta_\ell}\right)$
$\lambda_{m,\ell}^{\perp}$	$\tau_{m,\ell-1}^{G}/8(d_1+d_2)r\log(1+\frac{\tau_{m,\ell-1}^{G}}{\lambda})$
$B_{m,*}^{\ell}$	$(8\sqrt{\lambda}S+\sqrt{\lambda_{m,\ell}^{\perp}}S_{m,\ell}^{\perp})$
$\mathbf{B}_1$	Left feature extractor
$\mathbf{B}_2$	Right feature extractor
S_r	r-th largest singular value of $\mathbf{\Theta}_*$
$\Delta(\mathbf{x},\mathbf{z})$	$\mathbf{x}_*^{\top}\mathbf{\Theta}_*\mathbf{z}_*-\mathbf{x}^{\top}\mathbf{\Theta}_*\mathbf{z}$
Δ	$\min_{\mathbf{x}\neq\mathbf{x}_*,\mathbf{z}\neq\mathbf{z}_*}\Delta(\mathbf{x},\mathbf{z})$
$\mathcal{Y}(\mathcal{W})$	$\{\mathbf{w}-\mathbf{w}':\forall\mathbf{w},\mathbf{w}'\in\mathcal{W},\mathbf{w}\neq\mathbf{w}'\}$
$\mathcal{Y}^*(\mathcal{W})$	$\{\mathbf{w}_*-\mathbf{w}:\forall\mathbf{w}\in\mathcal{W}\backslash\mathbf{w}_*\}$
δ	confidence level

Table D.1: Table of Notations for GOBLIN

E APPENDIX: PRETRAINING DECISION TRANSFORMERS WITH REWARD PREDICTION FOR IN-CONTEXT STRUCTURED BANDIT LEARNING

E.1 Experimental Setting Information and Details of Baselines

In this section, we describe in detail the experimental settings and some baselines.

Experimental Details

Linear Bandit: We consider the setting when $f(\mathbf{x}, \theta_*) = \mathbf{x}^\top \theta_*$. Here $\mathbf{x} \in \mathbb{R}^d$ is the action feature and $\theta_* \in \mathbb{R}^d$ is the hidden parameter. For every experiment, we first generate tasks from $\mathcal{T}_{\text{pre}}$. Then we sample a fixed set of actions from $\mathcal{N}(\mathbf{0}, \mathbf{I}_d/d)$ in $\mathbb{R}^d$ and this constitutes the features. Then for each task $m \in [M]$ we sample $\theta_{m,*} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d/d)$ to produce the means $\mu(m, a) = \langle \theta_{m,*}, \mathbf{x}(m, a) \rangle$ for $a \in \mathcal{A}$ and $m \in [M]$. Finally, note that we do not shuffle the data as the order matters. Also in this setting $\mathbf{x}(m, a)$ for each $a \in \mathcal{A}$ is fixed for all tasks m.

Non-Linear Bandit: We now consider the setting when $f(\mathbf{x}, \theta_*) = 1/(1 + 0.5 \cdot \exp(2 \cdot \exp(-\mathbf{x}^\top \theta_*)))$. Again, here $\mathbf{x} \in \mathbb{R}^d$ is the action feature, and $\theta_* \in \mathbb{R}^d$ is the hidden parameter. Note that this is different than the generalized linear bandit setting (Filippi et al., 2010a; Li et al., 2017b). Again for every experiment, we first generate tasks from $\mathcal{T}_{\text{pre}}$. Then we sample a fixed set of actions from $\mathcal{N}(\mathbf{0}, \mathbf{I}_d/d)$ in $\mathbb{R}^d$ and this constitutes the features. Then for each task $m \in [M]$ we sample $\theta_{m,*} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d/d)$ to produce the means $\mu(m, a) = 1/(1 + 0.5 \cdot \exp(2 \cdot \exp(-\mathbf{x}(m, a)^\top \theta_{m,*})))$ for $a \in \mathcal{A}$ and $m \in [M]$. Again note that in this setting $\mathbf{x}(m, a)$ for each

$a \in \mathcal{A}$ is fixed for all tasks m.

We use NVIDIA GeForce RTX 3090 GPU with 24GB RAM to load the GPT 2 Large Language Model. This requires less than 2GB RAM without data, and with large context may require as much as 20GB RAM.

Details of Baselines

(1) Thomp: This baseline is the stochastic A-action bandit Thompson Sampling algorithm from Thompson (1933); Agrawal and Goyal (2012); Russo et al. (2018); Zhu and Tan (2020). We briefly describe the algorithm below: At every round t and each action a, Thomp samples $\gamma_{m,t}(a) \sim \mathcal{N}(\widehat{\mu}_{m,t-1}(a), \sigma^2/N_{m,t-1}(a))$, where $N_{m,t-1}(a)$ is the number of times the action a has been selected till $t-1$, and $\widehat{\mu}_{m,t-1}(a) = \frac{\sum_{s=1}^{t-1} \hat{r}_{m,s} \mathbf{1}(I_s = a)}{N_{m,t-1}(a)}$ is the empirical mean. Then the action selected at round t is $I_t = arg\,max_a\, \gamma_{m,t}(a)$. Observe that Thomp is not a deterministic algorithm like UCB (Auer et al., 2002). So we choose Thomp as the weak demonstrator π^w because it is more exploratory than UCB and also chooses the optimal action, $a_{m,*}$, a sufficiently large number of times. Thomp is a weak demonstrator as it does not have access to the feature set $\mathcal{X}$ for any task m.

(2) LinUCB: (Linear Upper Confidence Bound): This baseline is the Upper Confidence Bound algorithm for the linear bandit setting that selects the action I_t at round t for task m that is most optimistic and reduces the uncertainty of the task unknown parameter $\theta_{m,*}$. To balance exploitation and exploration between choosing different items the LinUCB computes an upper confidence value to the estimated mean of each action $\mathbf{x}_{m,a} \in \mathcal{X}$. This is done as follows: At every round t for task m, it calculates the ucb value $B_{m,a,t}$ for each action $\mathbf{x}_{m,a} \in \mathcal{X}$ such that $B_{m,a,t} = \mathbf{x}_{m,a}^\top \widehat{\theta}_{m,t-1} + \alpha \|\mathbf{x}_{m,a}\|_{\mathbf{\Sigma}_{m,t-1}^{-1}}$ where $\alpha > 0$ is a constant and $\widehat{\theta}_{m,t}$ is the estimate of the model parameter $\theta_{m,*}$ at round t. Here, $\mathbf{\Sigma}_{m,t-1} = \sum_{s=1}^{t-1} \mathbf{x}_{m,s}\mathbf{x}_{m,s}^\top + \lambda \mathbf{I}_d$ is the data covariance matrix or the arms already tried. Then it chooses $I_t = arg\,max_a\, B_{m,a,t}$. Note that LinUCB is a *strong* demonstrator that

we give oracle access to the features of each action; other algorithms do not observe the features. Hence, in linear bandits, LinUCB provides an approximate upper bound on the performance of all algorithms.

(3) MLinGreedy: This is the multi-task linear regression bandit algorithm proposed by Yang et al. (2021a). This algorithm assumes that there is a common low dimensional feature extractor $\mathbf{B} \in \mathbb{R}^{k \times d}$, $k \leqslant d$ shared between the tasks and the rewards per task m are linearly dependent on a hidden parameter $\theta_{m,*}$. Under a diversity assumption (which may not be satisfied in real data) and $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_M]$ they assume $\mathbf{\Theta} = [\theta_{1,*}, \ldots, \theta_{M,*}] = \mathbf{BW}$. During evaluation MLinGreedy estimates the $\widehat{\mathbf{B}}$ and $\widehat{\mathbf{W}}$ from training data and fit $\widehat{\theta}_m = \widehat{\mathbf{B}}\widehat{\mathbf{w}}_m$ per task and selects action greedily based on $I_{m,t} = \arg\max_a \mathbf{x}_{m,a}^\top \widehat{\theta}_{m,*}$. Finally, note that MLinGreedy requires access to the action features to estimate $\widehat{\theta}_m$ and select actions as opposed to DPT, AD, and PreDeToR.

E.2 Empirical Study: Bilinear Bandits

In this section, we discuss the performance of PreDeToR against the other baselines in the bilinear setting. Again note that the number of tasks $M_{\text{pre}} \gg A \geqslant n$. Through this experiment, we want to evaluate the performance of PreDeToR to exploit the underlying latent structure and reward correlation when the horizon is small, the number of tasks is large, and understand its performance in the bilinear bandit setting (Jun et al., 2019; Lu et al., 2021; Kang et al., 2022; Mukherjee et al., 2023b). Note that this setting also goes beyond the linear feedback model (Abbasi-Yadkori et al., 2011; Lattimore and Szepesvári, 2020a) and is related to matrix bandits (Yang and Wang, 2020).

Bilinear bandit setting: In the bilinear bandits the learner is provided with two sets of action sets, $\mathcal{X} \subseteq \mathbb{R}^{d_1}$ and $\mathcal{Z} \subseteq \mathbb{R}^{d_2}$ which are referred to as the left and right action sets. At every round t the learner chooses a pair

of actions $\mathbf{x}_t \in \mathcal{X}$ and $\mathbf{z}_t \in \mathcal{Z}$ and observes a reward

$$r_t = \mathbf{x}_t^\top \boldsymbol{\Theta}_* \mathbf{z}_t + \eta_t$$

where $\boldsymbol{\Theta}_* \in \mathbb{R}^{d_1 \times d_2}$ is the unknown hidden matrix which is also low-rank. The η_t is a σ^2 sub-Gaussian noise. In the multi-task bilinear bandit setting we now have a set of M tasks where the reward for the m-th task at round t is given by

$$r_{m,t} = \mathbf{x}_{m,t}^\top \boldsymbol{\Theta}_{m,*} \mathbf{z}_{m,t} + \eta_{m,t}.$$

Here $\boldsymbol{\Theta}_{m,*} \in \mathbb{R}^{d_1 \times d_2}$ is the unknown hidden matrix for each task m, which is also low-rank. The $\eta_{m,t}$ is a σ^2 sub-Gaussian noise. Let κ be the rank of each of these matrices $\boldsymbol{\Theta}_{m,*}$.

A special case is the rank 1 structure where $\boldsymbol{\Theta}_{m,*} = \theta_{m,*}\theta_{m,*}^\top$ where $\boldsymbol{\Theta}_{m,*} \in \mathbb{R}^{d \times d}$ and $\theta_{m,*} \in \mathbb{R}^d$ for each task m. Let the left and right action sets be also same such that $\mathbf{x}_{m,t} \in \mathcal{X} \subseteq \mathbb{R}^d$. Observe then that the reward for the m-th task at round t is given by

$$r_{m,t} = \mathbf{x}_{m,t}^\top \boldsymbol{\Theta}_{m,*} \mathbf{x}_{m,t} + \eta_{m,t} = (\mathbf{x}_{m,t}^\top \theta_{m,*})^2 + \eta_{m,t}.$$

This special case is studied in Chaudhuri et al. (2017).

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, and Thomp. Note that we do not implement the LinUCB and MLinGreedy for the bilinear bandit setting. However, we now implement the LowOFUL (Jun et al., 2019) which is optimal in the bilinear bandit setting.

LowOFUL: The LowOFUL algorithm first estimates the unknown parameter $\boldsymbol{\Theta}_{m,*}$ for each task m using E-optimal design (Pukelsheim, 2006; Fedorov, 2013; Jun et al., 2019) for n_1 rounds. Let $\widehat{\boldsymbol{\Theta}}_{m,n_1}$ be the estimate of $\boldsymbol{\Theta}_{m,*}$ at the end of n_1 rounds. Let the SVD of $\widehat{\boldsymbol{\Theta}}_{m,n_1}$ be given

by $\text{SVD}(\widehat{\mathbf{\Theta}}_{m,n_1}) = \widehat{\mathbf{U}}_{m,n_1}\widehat{\mathbf{S}}_{m,n_1}\widehat{\mathbf{V}}_{m,n_1}$. Then LowOFUL rotates the actions as follows:

$$\mathcal{X}'_m = \left\{ \left[\widehat{\mathbf{U}}_{m,n_1}\widehat{\mathbf{U}}^{\perp}_{m,n_1}\right]^\top \mathbf{x}_m : \mathbf{x}_m \in \mathcal{X} \right\} \textbf{ and}$$
$$\mathcal{Z}' = \left\{ \left[\widehat{\mathbf{V}}_{m,n_1}\widehat{\mathbf{V}}^{\perp}_{m,n_1}\right]^\top \mathbf{z}_m : \mathbf{z}_m \in \mathcal{Z} \right\}.$$

Then defines a vectorized action set for each task m so that the last $(d_1 - \kappa)\cdot(d_2 - \kappa)$ components are from the complementary subspaces:

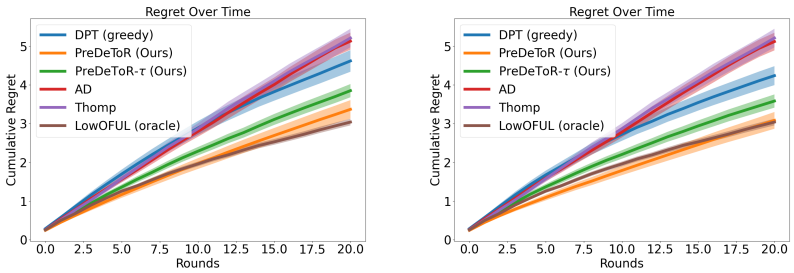
$$\widetilde{\mathcal{A}}_m = \{ [\text{vec}\left(\mathbf{x}_{m,1:\kappa}\mathbf{z}^\top_{m,1:\kappa}\right); \text{vec}\left(\mathbf{x}_{m,\kappa+1:d_1}\mathbf{z}^\top_{m,1:\kappa}\right); \text{vec}\left(\mathbf{x}_{m,1:\kappa}\mathbf{z}^\top_{m,\kappa+1:d_2}\right); \text{vec}\left(\mathbf{x}_{m,\kappa+1:d_1}\mathbf{z}^\top_{m,\kappa+1:d_2}\right)] \in \mathbb{R}^{d_1 d_2} : \mathbf{x}_m \in \mathcal{X}'_m, \mathbf{z}_m \in \mathcal{Z}'_m \}.$$

Finally for $n_2 = n - n_1$ rounds, LowOFUL invokes the specialized OFUL algorithm (Abbasi-Yadkori et al., 2011) for the rotated action set $\widetilde{\mathcal{A}}_m$ with the low dimension $k = (d_1 + d_2)\,\kappa - \kappa^2$. Note that the LowOFUL runs the per-task low dimensional OFUL algorithm rather than learning the underlying structure across the tasks (Mukherjee et al., 2023b).

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

> **Finding 5:** PreDeToR (-τ) outperforms DPT-greedy, AD, and matches the performance of LowOFUL in bilinear bandit setting.

Experimental Result: We observe these outcomes in Figure E.1. In Figure E.1a we experiment with rank 1 hidden parameter $\mathbf{\Theta}_{m,*}$ and set horizon $n = 20$, $M_{\text{pre}} = 200000$, $M_{\text{test}} = 200$, $A = 30$, and $d = 5$. In Figure E.1b we experiment with rank 2 hidden parameter $\mathbf{\Theta}_{m,*}$ and set horizon $n = 20$, $M_{\text{pre}} = 250000$, $M_{\text{test}} = 200$, $A = 25$, and $d = 5$. Again, the demonstrator π^w is the Thomp algorithm. We observe that PreDeToR has lower cumulative regret than DPT-greedy, AD and Thomp. Note



(a) Rank 1 $\Theta_{m,*}$ (b) Rank 2 $\Theta_{m,*}$

Figure E.1: Experiment with bilinear bandits. The y-axis shows the cumulative regret.

that for any task m for the horizon 20 the Thomp will be able to sample all the actions at most once. Note that for this small horizon setting the DPT-greedy does not have a good estimation of $\widehat{a}_{m,*}$ which results in a poor prediction of optimal action $\widehat{a}_{m,t,*}$. In contrast PreDeToR learns the correlation of rewards across tasks and can perform well. Observe from Figure E.1a, and E.1b that PreDeToR has lower regret than Thomp and matches LowOFUL. Also, in this low-data regime it is not enough for LowOFUL to learn the underlying $\Theta_{m,*}$ with high precision. Hence, PreDeToR also has slightly lower regret than LowOFUL. Note that the main objective of AD is to match the performance of its demonstrator. Most importantly it shows that PreDeToR can exploit the underlying latent structure and reward correlation better than DPT-greedy, and AD.

E.3 Empirical Study: Latent Bandits

In this section, we discuss the performance of PreDeToR (-τ) against the other baselines in the latent bandit setting and create a generalized bilinear bandit setting. Note that the number of tasks $M_{pre} \gg A \geqslant n$. Using this experiment, we want to evaluate the ability of PreDeToR (-τ) to exploit the underlying reward correlation when the horizon is small, the number of

tasks is large, and understand its performance in the latent bandit setting (Hong et al., 2020; Maillard and Mannor, 2014; Pal et al., 2023; Kveton et al., 2017). We create a latent bandit setting which generalizes the bilinear bandit setting (Jun et al., 2019; Lu et al., 2021; Kang et al., 2022; Mukherjee et al., 2023b). Again note that this setting also goes beyond the linear feedback model (Abbasi-Yadkori et al., 2011; Lattimore and Szepesvári, 2020a) and is related to matrix bandits (Yang and Wang, 2020).

Latent bandit setting: In this special multi-task latent bandits the learner is again provided with two sets of action sets, $\mathcal{X} \subseteq \mathbb{R}^{d_1}$ and $\mathcal{Z} \subseteq \mathbb{R}^{d_2}$ which are referred to as the left and right action sets. The reward for the m-th task at round t is given by

$$r_{m,t} = \mathbf{x}_{m,t}^\top \underbrace{(\mathbf{\Theta}_{m,*} + \overline{U}\mathbf{V}^\top)}_{\mathbf{Z}_{m_*}} \mathbf{z}_{m,t} + \eta_{m,t}.$$

Here $\mathbf{\Theta}_{m,*} \in \mathbb{R}^{d_1 \times d_2}$ is the unknown hidden matrix for each task m, which is also low-rank. Additionally, all the tasks share a *common latent parameter matrix* $\overline{U}\mathbf{V}^\top \in \mathbb{R}^{d_1 \times d_2}$ which is also low rank. Hence the learner needs to learn the latent parameter across the tasks hence the name latent bandits. Finally, the $\eta_{m,t}$ is a σ^2 sub-Gaussian noise. Let κ be the rank of each of these matrices $\mathbf{\Theta}_{m,*}$ and $\overline{U}\mathbf{V}^\top$. Again special case is the rank 1 structure where the reward for the m-th task at round t is given by

$$r_{m,t} = \mathbf{x}_{m,t}^\top \underbrace{(\theta_{m,*}\theta_{m,*}^\top + \mathbf{u}\mathbf{x}^\top)}_{\mathbf{Z}_{m,*}} \mathbf{x}_{m,t} + \eta_{m,t}.$$

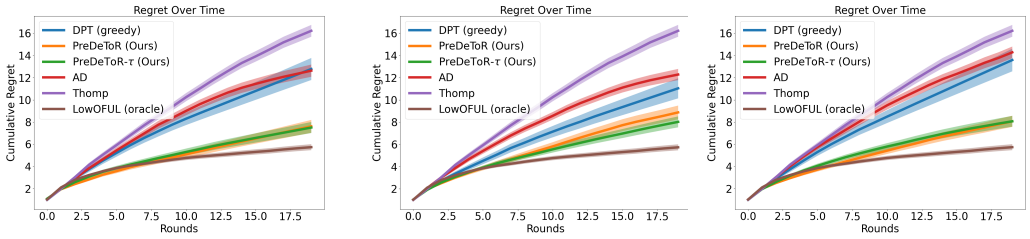
where $\theta_{m,*} \in \mathbb{R}^d$ for each task m and $\mathbf{u}, \mathbf{x} \in \mathbb{R}^d$. Note that the left and right action sets are the same such that $\mathbf{x}_{m,t} \in \mathcal{X} \subseteq \mathbb{R}^d$.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LowOFUL. However, we now implement a special LowOFUL

(stated in Section E.2) which has knowledge of the shared latent parameters $\overline{U}$, and $\mathbf{V}$. We call this the LowOFUL (oracle) algorithm. Therefore LowOFUL (oracle) has knowledge of the problem parameters in the latent bandit setting and hence the name. Again note that we do not implement the LinUCB and MLinGreedy for the latent bandit setting.

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

> **Finding 6:** PreDeToR (-τ) outperforms DPT-greedy, AD, and matches the performance of LowOFUL (oracle) in latent bandit setting.



(a) Rank 1 $\mathbf{Z}_{m,*}$ (b) Rank 2 $\mathbf{Z}_{m,*}$ (c) Rank 3 $\mathbf{Z}_{m,*}$

Figure E.2: Experiment with latent bandits. The y-axis shows the cumulative regret.

Experimental Result: We observe these outcomes in Figure E.2. In Figure E.2a we experiment with rank 1 hidden parameter $\theta_{m,*}\theta_{m,*}^\top$ and latent parameters $\mathbf{u}\mathbf{x}^\top$ shared across the tasks and set horizon $n = 20$, $M_{pre} = 200000$, $M_{test} = 200$, $A = 30$, and $d = 5$. In Figure E.2b we experiment with rank 2 hidden parameter $\mathbf{\Theta}_{m,*}$, and latent parameters $\overline{U}\mathbf{V}^\top$ and set horizon $n = 20$, $M_{pre} = 250000$, $M_{test} = 200$, $A = 25$, and $d = 5$. In Figure E.2c we experiment with rank 3 hidden parameter $\mathbf{\Theta}_{m,*}$, and latent parameters $\overline{U}\mathbf{V}^\top$ and set horizon $n = 20$, $M_{pre} = 300000$, $M_{test} = 200$, $A = 25$, and $d = 5$. Again, the demonstrator π^w is the

Thomp algorithm. We observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, AD and Thomp. Note that for any task m for the horizon 20 the Thomp will be able to sample all the actions at most once. Note that for this small horizon setting the DPT-greedy does not have a good estimation of $\widehat{a}_{m,*}$ which results in a poor prediction of optimal action $\widehat{a}_{m,t,*}$. In contrast PreDeToR (-τ) learns the correlation of rewards across tasks and is able to perform well. Observe from Figure E.2a, E.2b, and E.2c that PreDeToR has lower regret than Thomp and has regret closer to LowOFUL (oracle)which has access to the problem-dependent parameters. Hence. LowOFUL (oracle) outperforms PreDeToR (-τ) in this setting. This shows that PreDeToR is able to exploit the underlying latent structure and reward correlation better than DPT-greedy, and AD.

E.4 Connection between PreDeToR and Linear Multivariate Gaussian Model

In this section, we try to understand the behavior of PreDeToR and its ability to exploit the reward correlation across tasks under a *linear multivariate Gaussian model*. In this model, the hidden task parameter, θ_*, is a random variable drawn from a multi-variate Gaussian distribution (Bishop, 2006a) and the feedback follows a linear model. We study this setting since we can estimate the Linear Minimum Mean Square Estimator (LMMSE) in this setting (Carlin and Louis, 2008; Box and Tiao, 2011). This yields a posterior prediction for the mean of each action over all tasks on average, by leveraging the linear structure when θ_* is drawn from a multi-variate Gaussian distribution. So we can compare the performance of PreDeToR against such an LMMSE and evaluate whether it is exploiting the underlying linear structure and the reward correlation across tasks. We summarize this as follows:

Finding 7: PreDeToR learns the reward correlation covariance matrix from the in-context training data $\mathcal{H}_{\text{train}}$ and acts greedily on it.

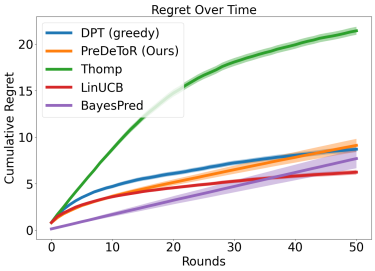


Figure E.3: BayesPred Performance

Consider the linear feedback setting consisting of A actions and the hidden task parameter $\theta_* \sim \mathcal{N}(0, \sigma_\theta^2 \mathbf{I}_d)$. The reward of the action $\mathbf{x}_t$ at round t is given by $r_t = \mathbf{x}_t^\top \theta_* + \eta_t$, where η_t is σ^2 sub-Gaussian. Let π^w collect n rounds of pretraining in-context data and observe $\{I_t, r_t\}_{t=1}^n$. Let $N_n(a)$ denote the total number of times the action a is sampled for n rounds. Note that we drop the task index m in these notations as the random variable θ_* corresponds to the task. Define the matrix $\mathbf{H}_n \in \mathbb{R}^{n \times A}$ where the t-th row represents the action I_t for $t \in [n]$. The t-th row of $\mathbf{H}_n$ is a one-hot vector with the I_t-th component being 1. We represent each action by one hot vector because we assume that this LMMSE does not have access to the feature vectors of the actions similar to the PreDeToR for fair comparison. Then define the reward vector $\mathbf{Y}_n \in \mathbb{R}^n$ where the t-th component is the reward r_t observed for the action I_t for $t \in [n]$ in the pretraining data. Define the diagonal matrix $\mathbf{D}_A \in \mathbb{R}^{A \times A}$ estimated from pretraining data as follows

$$\mathbf{D}_A(i,i) = \begin{cases} \frac{\sigma^2}{N_n(a)}, \text{ if } N_n(a) > 0 \\ = 0, \text{ if } N_n(a) = 0 \end{cases} \tag{E.1}$$

where the reward noise being σ^2 sub-Gaussian is known. Finally define the estimated reward covariance matrix $\mathbf{S}_A \in \mathbb{R}^{A \times A}$ as $\mathbf{S}_A(a, a') = \widehat{\mu}_n(a)\widehat{\mu}_n(a')$, where $\widehat{\mu}_n(a)$ is the empirical mean of action a estimated

from the pretraining data. This matrix captures the reward correlation between the pairs of actions $a, a' \in [A]$. Then the posterior average mean estimator $\widehat{\mu} \in \mathbb{R}^A$ over all tasks is given by the following lemma. The proof is given in Section E.15.

Lemma 1. *Let $\mathbf{H}_n$ be the action matrix, $\mathbf{Y}_n$ be the reward vector and $\mathbf{S}_A$ be the estimated reward covariance matrix. Then the posterior prediction of the average mean reward vector $\widehat{\mu}$ over all tasks is given by*

$$\widehat{\mu} = \sigma_\theta^2 \mathbf{S}_A \mathbf{H}_n^\top \left(\sigma_\theta^2 \mathbf{H}_n (\mathbf{S}_A + \mathbf{D}_A) \mathbf{H}_n^\top\right)^{-1} \mathbf{Y}_n. \tag{E.2}$$

The $\widehat{\mu}$ in (E.2) represents the posterior mean vector averaged on all tasks. So if some action $a \in [A]$ consistently yields high rewards in the pretraining data then $\widehat{\mu}(a)$ has high value. Since the test distribution is the same as pretraining, this action on average will yield a high reward during test time.

We hypothesize that the PreDeToR is learning the reward correlation covariance matrix from the training data $\mathcal{H}_{\text{train}}$ and acting greedily on it. To test this hypothesis, we consider the greedy BayesPred algorithm that first estimates $\mathbf{S}_A$ from the pretraining data. It then uses the LMMSE estimator in Lemma 1 to calculate the posterior mean vector $\widehat{\mu}$, and then selects $I_t = \arg\max_a \widehat{\mu}(a)$ at each round t. Note that BayesPred is a greedy algorithm that always selects the most rewarding action (exploitation) without any exploration of sub-optimal actions. Also the BayesPred is an LMMSE estimator that leverages the linear reward structure and estimates the reward covariance matrix, and therefore can be interpreted as a lower bound to the regret of PreDeToR. The hypothesis that BayesPred is a lower bound to PreDeToR is supported by Figure E.3. In Figure E.3 the reward covariance matrix for BayesPred is estimated from the $\mathcal{H}_{\text{train}}$ by first running the Thomp (π^w). Observe that the BayesPred has a lower cumulative regret than PreDeToR and almost matches the regret of PreDeToR towards

the end of the horizon. Also note that LinUCB has lower cumulative regret towards the end of horizon as it leverages the linear structure and the feature of the actions in selecting the next action.

E.5 Empirical Study: Increasing number of Actions

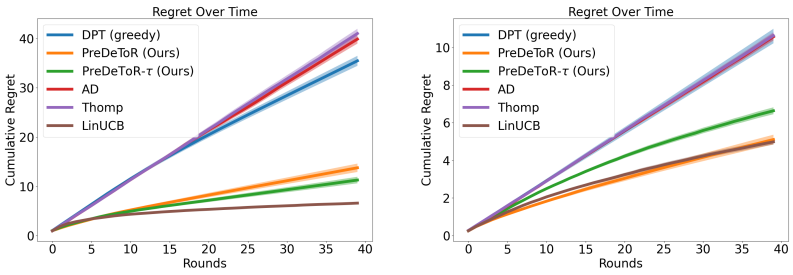
In this section, we discuss the performance of PreDeToR when the number of actions is very high so that the weak demonstrator π^w does not have sufficient samples for each action. However, the number of tasks $M_{pre} \gg A > n$.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LinUCB.

Outcomes: We first discuss the main outcomes from our experimental results of introducing more actions than the horizon (or more dimensions than actions) during data collection and evaluation:

> **Finding 8:** PreDeToR (-τ) outperforms DPT-greedy, and AD, even when $A > n$ but $M_{pre} \gg A$.

Experimental Result: We observe these outcomes in Figure E.4. In Figure E.4a we show the linear bandit setting for $M_{pre} = 250000$, $M_{test} = 200$, $A = 100$, $n = 50$ and $d = 5$. Again, the demonstrator π^w is the Thomp algorithm. We observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy and AD. Note that for any task m the Thomp will not be able to sample all the actions even once. The weak performance of DPT-greedy can be attributed to both short horizons and the inability to estimate the optimal action for such a short horizon $n < A$. The AD performs similar to the demonstrator Thomp because of its training. Observe that



(a) Linear Bandit (b) Non-linear Bandit

Figure E.4: Testing the limit experiments. The horizontal axis is the number of rounds. Confidence bars show one standard error.

PreDeToR (-τ) has similar regret to LinUCB and lower regret than Thomp which also shows that PreDeToR is exploiting the latent linear structure of the underlying tasks. In Figure E.4b we show the non-linear bandit setting for horizon $n = 40$, $M_{pre} = 200000$, $A = 60$, $d = 2$, and $|\mathcal{A}^{inv}| = 5$. The demonstrator π^w is the Thomp algorithm. Again we observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, AD and LinUCB which fails to perform well in this non-linear setting due to its algorithmic design.

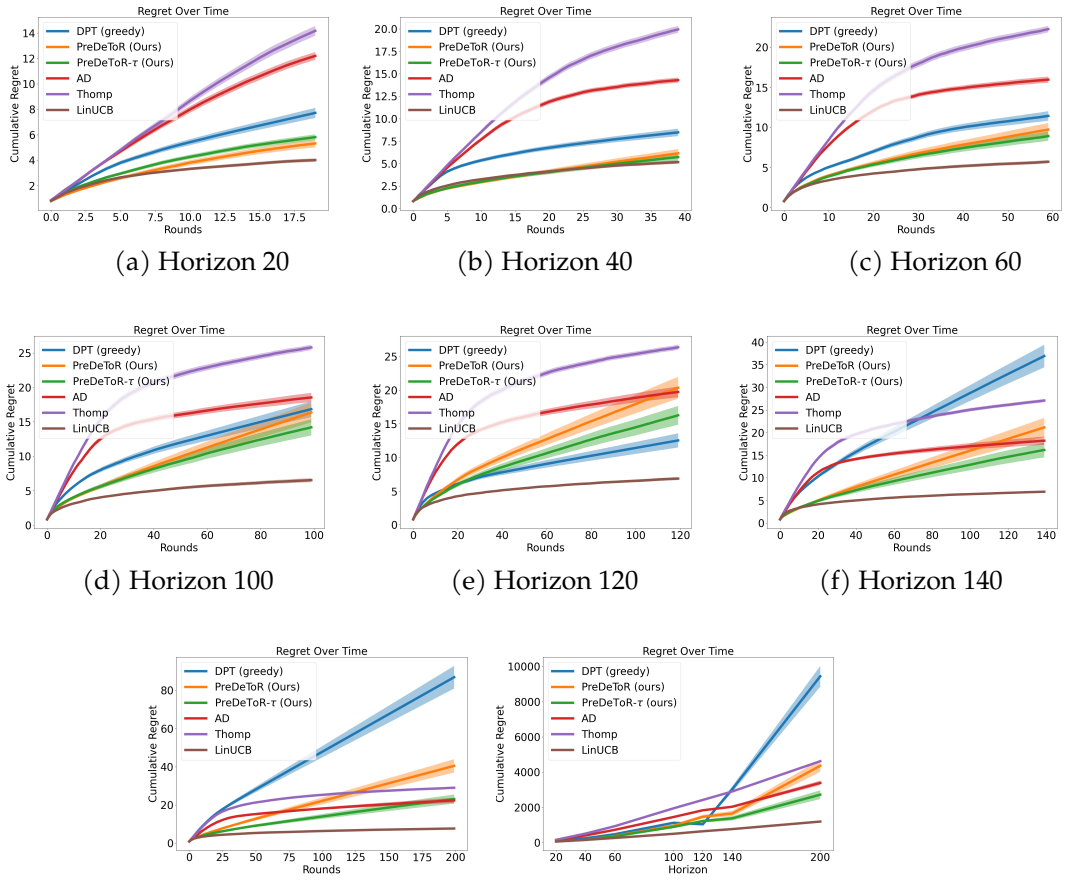
E.6 Empirical Study: Increasing Horizon

In this section, we discuss the performance of PreDeToR with respect to an increasing horizon for each task $m \in [M]$. However, note that the number of tasks $M_{pre} \geqslant n$. Note that Lee et al. (2023) studied linear bandit setting for $n = 200$. We study the setting up to a similar horizon scale.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LinUCB.

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

Finding 9: PreDeToR (-τ) outperforms DPT-greedy, and AD with increasing horizon.



(a) Horizon 20 (b) Horizon 40 (c) Horizon 60

(d) Horizon 100 (e) Horizon 120 (f) Horizon 140

(g) Horizon 200 (h) Increasing Horizon

Figure E.5: Experiment with increasing horizon. The y-axis shows the cumulative regret.

Experimental Result: We observe these outcomes in Figure E.5. In Figure E.5 we show the linear bandit setting for $M_{pre} = 150000$, $M_{test} = 200$, $A = 20$, $n = \{20, 40, 60, 100, 120, 140, 200\}$ and $d = 5$. Again, the demonstrator π^w is the Thomp algorithm. We observe that PreDeToR (-τ)

has lower cumulative regret than DPT-greedy, and AD. Note that for any task m for the horizon 20 the Thomp will be able to sample all the actions at most once. Observe from Figure E.5a, E.5b, E.5c, Figure E.5d, E.5e, E.5f and E.5g that PreDeToR (-τ) is closer to LinUCB and outperforms Thomp which also shows that PreDeToR (-τ) is learning the latent linear structure of the underlying tasks. In Figure E.5h we plot the regret of all the baselines with respect to the increasing horizon. Again we see that PreDeToR (-τ) is closer to LinUCB and outperforms DPT-greedy, AD and Thomp. This shows that PreDeToR (-τ) is able to exploit the latent structure and reward correlation across the tasks for varying horizon length.

E.7 Empirical Study: Increasing Dimension

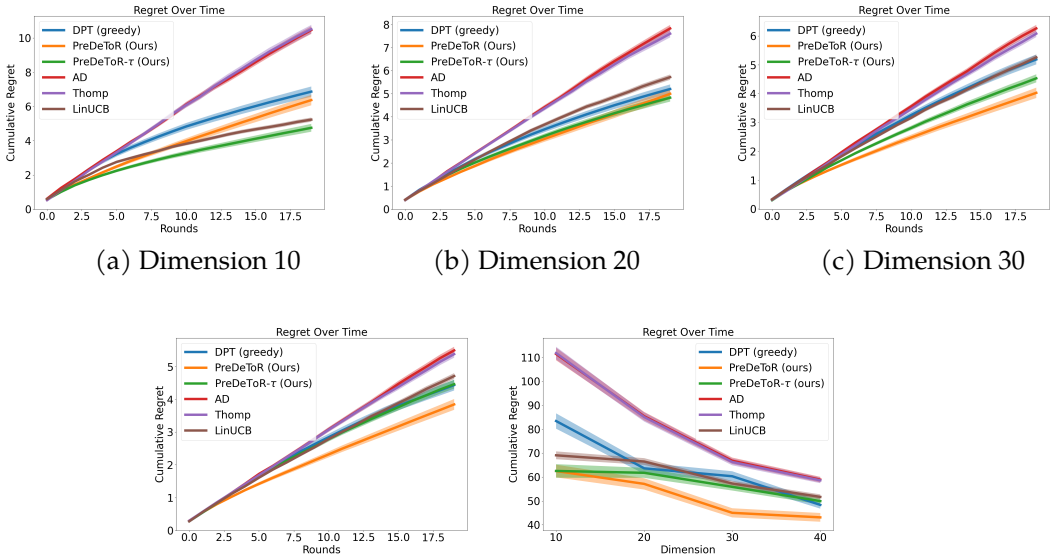
In this section, we discuss the performance of PreDeToR with respect to an increasing dimension for each task $m \in [M]$. Again note that the number of tasks $M_{pre} \gg A \geqslant n$. Through this experiment, we want to evaluate the performance of PreDeToR and see how it exploits the underlying reward correlation when the horizon is small as well as for increasing dimensions.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ DPT-greedy, AD, Thomp, and LinUCB.

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

> **Finding 10:** PreDeToR (-τ) outperforms DPT-greedy, AD with increasing dimension and has lower regret than LinUCB for larger dimension.

Experimental Result: We observe these outcomes in Figure E.5. In Figure E.5 we show the linear bandit setting for horizon $n = 20$, $M_{pre} = 160000$, $M_{test} = 200$, $A = 20$, and $d = \{10, 20, 30, 40\}$. Again, the demon-



(a) Dimension 10 (b) Dimension 20 (c) Dimension 30

(d) Dimension 40 (e) Increasing Dimension

Figure E.6: Experiment with increasing dimension. The y-axis shows the cumulative regret.

strator π^w is the Thomp algorithm. We observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, AD. Note that for any task m for the horizon 20 the Thomp will be able to sample all the actions at most once. Observe from Figure E.6a, E.6b, E.6c, and E.6d that PreDeToR (-τ) is closer to LinUCB and has lower regret than Thomp which also shows that PreDeToR (-τ) is exploiting the latent linear structure of the underlying tasks. In Figure E.6e we plot the regret of all the baselines with respect to the increasing dimension. Again we see that PreDeToR (-τ) has lower regret than DPT-greedy, AD and Thomp. Observe that with increasing dimension PreDeToR is able to outperform LinUCB. This shows that the PreDeToR (-τ) is able to exploit reward correlation across tasks for varying dimensions.

E.8 Empirical Study: Increasing Attention Heads

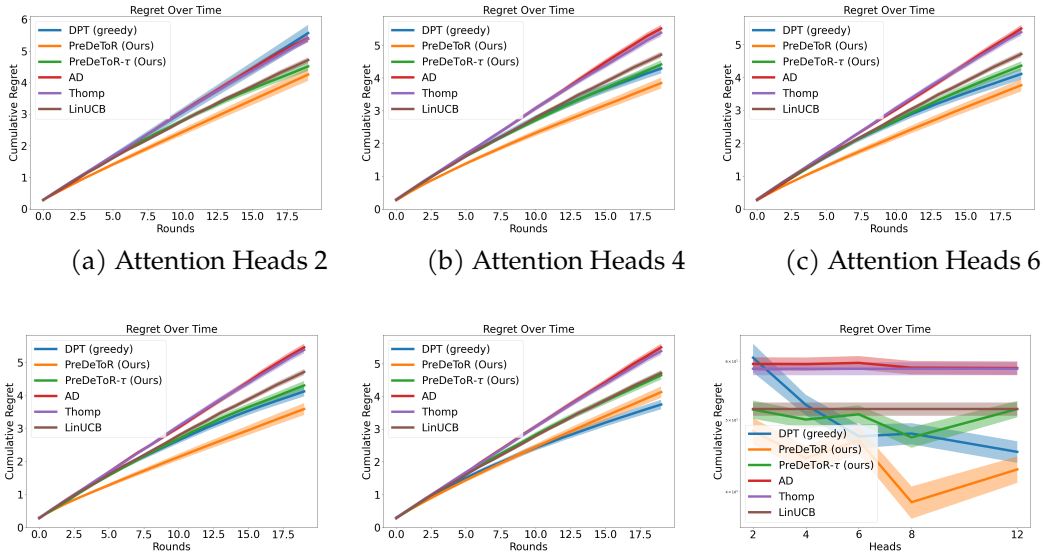
In this section, we discuss the performance of PreDeToR with respect to an increasing attention heads for the transformer model for the non-linear feedback model. Again note that the number of tasks $M_{pre} \gg A \geqslant n$. Through this experiment, we want to evaluate the performance of PreDeToR to exploit the underlying reward correlation when the horizon is small and understand the representative power of the transformer by increasing the attention heads. Note that we choose the non-linear feedback model and low data regime to leverage the representative power of the transformer.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LinUCB.

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

> **Finding 11:** PreDeToR (-τ) outperforms DPT-greedy, and AD with increasing attention heads.

Experimental Result: We observe these outcomes in Figure E.7. In Figure E.7 we show the non-linear bandit setting for horizon $n = 20$, $M_{pre} = 160000$, $M_{test} = 200$, $A = 20$, heads $= \{2, 4, 6, 8\}$ and $d = 5$. Again, the demonstrator π^w is the Thomp algorithm. We observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, AD. Note that for any task m for the horizon 20 the Thomp will be able to sample all the actions atmost once. Observe from Figure E.7a, E.7b, E.7c, and E.7d that PreDeToR (-τ) has lower regret than AD, Thomp and LinUCB which also shows that PreDeToR (-τ) is exploiting the latent linear structure of the underlying



(a) Attention Heads 2 (b) Attention Heads 4 (c) Attention Heads 6

(d) Attention Heads 8 (e) Attention Heads 12 (f) Increasing Attention Heads

Figure E.7: Experiment with increasing attention heads. The y-axis shows the cumulative regret.

tasks for the non-linear setting. In Figure E.7f we plot the regret of all the baselines with respect to the increasing attention heads. Again we see that PreDeToR (-τ) regret decreases as we increase the attention heads.

E.9 Empirical Study: Increasing Number of Tasks

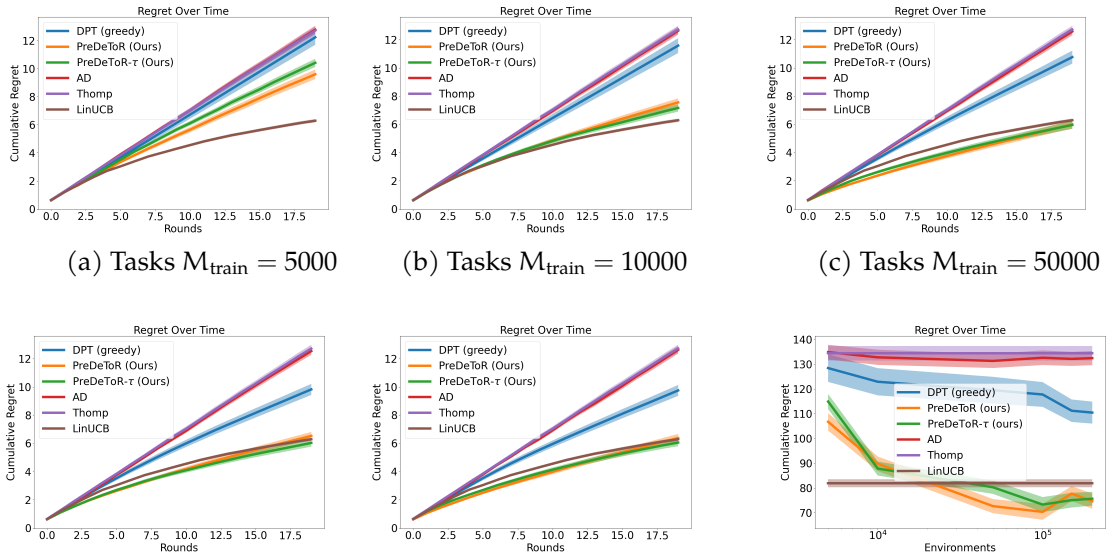
In this section, we discuss the performance of PreDeToR with respect to the increasing number of tasks for the linear bandit setting. Again note that the number of tasks $M_{\text{pre}} \gg A \geqslant n$. Through this experiment, we want to evaluate the performance of PreDeToR to exploit the underlying reward correlation when the horizon is small and the number of tasks is changing. Finally, recall that when the horizon is small the weak demonstrator π^w

does not have sufficient samples for each action. This leads to a poor approximation of the greedy action.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LinUCB.

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

> **Finding 12:** PreDeToR (-τ) fails to exploit the underlying latent structure and reward correlation from in-context data when the number of tasks is small.



(a) Tasks $M_{\text{train}} = 5000$ (b) Tasks $M_{\text{train}} = 10000$ (c) Tasks $M_{\text{train}} = 50000$

(d) Tasks $M_{\text{train}} = 100000$ (e) Tasks $M_{\text{train}} = 150000$ (f) Increasing tasks

Figure E.8: Experiment with an increasing number of tasks. The y-axis shows the cumulative regret.

Experimental Result: We observe these outcomes in Figure E.8. In Figure E.8 we show the linear bandit setting for horizon $n = 20$, $M_{\text{pre}} \in$

$\{5000, 10000, 50000, 100000, 150000\}$, $M_{\text{test}} = 200$, $A = 20$, and $d = 40$. Again, the demonstrator π^w is the Thomp algorithm. We observe that PreDeToR (-τ), AD and DPT-greedy suffer more regret than the LinUCB when the number of tasks is small ($M_{\text{train}} \in \{5000, 10000\}$ in Figure E.8a, and E.8b. However in Figure E.8c, E.8d, E.8e, and E.8f we show that PreDeToR has lower regret than Thomp and matches LinUCB. This shows that PreDeToR (-τ) is exploiting the latent linear structure of the underlying tasks for the non-linear setting. Moreover, observe that as M_{train} increases the PreDeToR has lower cumulative regret than DPT-greedy, AD. Note that for any task m for the horizon 20 the Thomp will be able to sample all the actions at most once. Therefore DPT-greedy does not perform as well as PreDeToR. Finally, note that the result shows that PreDeToR (-τ) is able to exploit the reward correlation across the tasks better as the number of tasks increases.

E.10 Exploration of PreDeToR(-τ)

In this section, we discuss the exploration of PreDeToR in the linear bandit setting discussed in Section 6.3. Recall that the linear bandit setting consist of horizon $n = 25$, $M_{\text{pre}} = 200000$, $M_{\text{test}} = 200$, $A = 10$, and $d = 2$. The demonstrator π^w is the Thomp algorithm and we observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, AD and matches the performance of LinUCB. Therefore PreDeToR (-τ) behaves almost optimally in this setting and so we analyze how PreDeToR conducts exploration for this setting.

Outcomes: We first discuss the main outcomes of our analysis of exploration in the low-data regime:

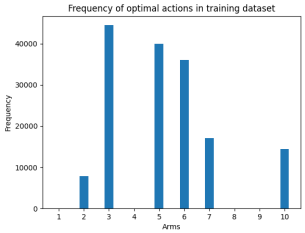
Finding 13: The PreDeToR (-τ) has a two phase exploration. In the first phase, it explores with a strong prior over the in-context training data. In the second phase, once the task data has been observed for a few rounds (in-context) it switches to task-based exploration.

We first show in Figure E.9a the training distribution of the optimal actions. For each bar, the frequency indicates the number of tasks where the action (shown in the x-axis) is the optimal action.

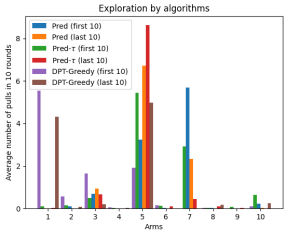
Then in Figure E.9b we show how the sampling distribution of DPT-greedy, PreDeToR and PreDeToR-τ change in the first 10 and last 10 rounds for all the tasks where action 5 is optimal. To plot this graph we first sum over the individual pulls of the action taken by each algorithm over the first 10 and last 10 rounds. Then we average these counts over all test tasks where action 5 is optimal. From the figure Figure E.9b we see that PreDeToR(-τ) consistently pulls the action 5 more than DPT-greedy. It also explores other optimal actions like $\{2, 3, 6, 7, 10\}$ but discards them quickly in favor of the optimal action 5 in these tasks. This shows that PreDeToR (-τ) only considers the optimal actions seen from the training data. Once sufficient observation have been observed for the task it switches to task-based exploration and samples the optimal action more than DPT-greedy.

Finally, we plot the feasible action set considered by DPT-greedy, PreDeToR, and PreDeToR-τ in Figure E.9c. To plot this graph again we consider the test tasks where the optimal action is 5. Then we count the number of distinct actions that are taken from round t up until horizon n. Finally we average this over all the considered tasks where the optimal action is 5. We call this the candidate action set considered by the algorithm. From the Figure E.9c we see that DPT-greedy explores the least and gets stuck with few actions quickly (by round 10). Note that the actions DPT-greedy samples are sub-optimal and so it suffers a high cumulative regret (see Figure 6.2). PreDeToR explore slightly more than DPT-greedy,

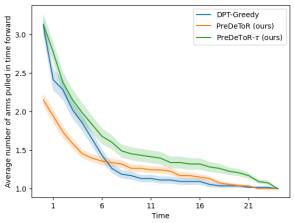
but PreDeToR-τ explores the most.



(a) Train Optimal Action Distribution



(b) Distribution of action sampling in all tasks where action 5 is optimal



(c) Candidate Action Set in Time averaged over all tasks where action 5 is optimal

Figure E.9: Exploration Analysis of PreDeToR(-τ)

E.11 Exploration of PreDeToR(-τ) in New Arms Setting

In this section, we discuss the exploration of PreDeToR (-τ) in the linear and non-linear new arms bandit setting discussed in Section 6.5. Recall that we consider the linear bandit setting of horizon $n = 50$, $M_{pre} = 200000$, $M_{test} = 200$, $A = 20$, and $d = 5$. Here during data collection and during collecting the test data, we randomly select one new action from $\mathbb{R}^d$ for each task m. So the number of invariant actions is $|\mathcal{A}^{inv}| = 19$.

Outcomes: We first discuss the main outcomes of our analysis of exploration in the low-data regime:

> **Finding 14:** The PreDeToR (-τ) is robust to changes when the number of in-variant actions is large. PreDeToR (-τ) performance drops as shared structure breaks down.

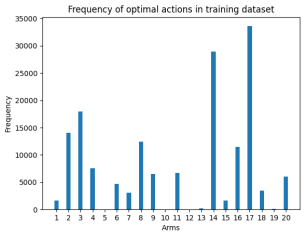
We first show in Figure E.10a the training distribution of the optimal actions. For each bar, the frequency indicates the number of tasks where the action (shown in the x-axis) is the optimal action.

Then in Figure E.10b we show how the sampling distribution of DPT-greedy, PreDeToR and PreDeToR-τ change in the first 10 and last 10 rounds for all the tasks where action 17 is optimal. We plot this graph the same way as discussed in Section E.10. From the figure Figure E.10b we see that PreDeToR(-τ) consistently pulls the action 17 more than DPT-greedy. It also explores other optimal actions like $\{1, 2, 3, 8, 9, 15\}$ but discards them quickly in favor of the optimal action 17 in these tasks.

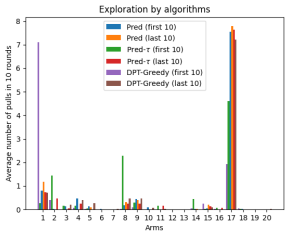
Finally, we plot the feasible action set considered by DPT-greedy, PreDeToR, and PreDeToR-τ in Figure E.10c. To plot this graph again we consider the test tasks where the optimal action is 17. Then we count the number of distinct actions that are taken from round t up until horizon n. Finally we average this over all the considered tasks where the optimal action is 17. We call this the candidate action set considered by the algorithm. From the Figure E.10c we see that PreDeToR-τ explores more than PreDeToR in this setting.

We also show how the prediction error of the optimal action by PreDeToR compared to LinUCB in this 1 new arm linear bandit setting. In Figure E.11a we first show how the 20 actions are distributed in the $M_{\text{test}} = 200$ test tasks. In Figure E.11a for each bar, the frequency indicates the number of tasks where the action (shown in the x-axis) is the optimal action. Then in Figure E.11b we show the prediction error of PreDeToR (-τ) for each task $m \in [M_{\text{test}}]$. The prediction error is calculated the same way as stated in Section 6.5 From the Figure E.11b we see that for most actions the prediction error of PreDeToR (-τ) is closer to LinUCB showing that the introduction of 1 new action does not alter the prediction error much. Note that LinUCB estimates the empirical mean directly from the test task, whereas PreDeToR has a strong prior based on the training data. Therefore

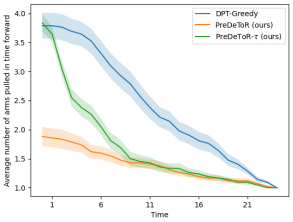
we see that PreDeToR is able to estimate the reward of the optimal action quite well from the training dataset $\mathcal{D}_{\text{pre}}$.



(a) Train Optimal Action Distribution

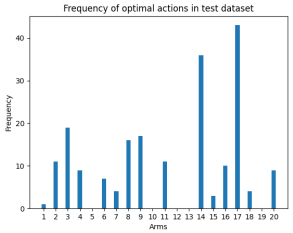


(b) Distribution of action sampling in all tasks where action 17 is optimal

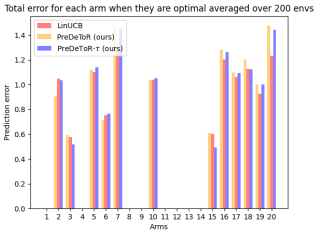


(c) Candidate Action Set in Time averaged over all tasks where action 17 is optimal

Figure E.10: Exploration Analysis of PreDeToR(-τ) in linear 1 new arm setting



(a) Test action distribution



(b) Test Prediction Error

Figure E.11: Prediction error of PreDeToR(-τ) in linear 1 new arm setting

We now consider the setting where the number of invariant actions is $|\mathcal{A}^{\text{inv}}| = 15$. We again show in Figure E.12a the training distribution of the optimal actions. For each bar, the frequency indicates the number of tasks where the action (shown in the x-axis) is the optimal action. Then in Figure E.12b we show how the sampling distribution of DPT-greedy, PreDeToR and PreDeToR-τ change in the first 10 and last 10 rounds for all the tasks where action 17 is optimal. We plot this graph the same way

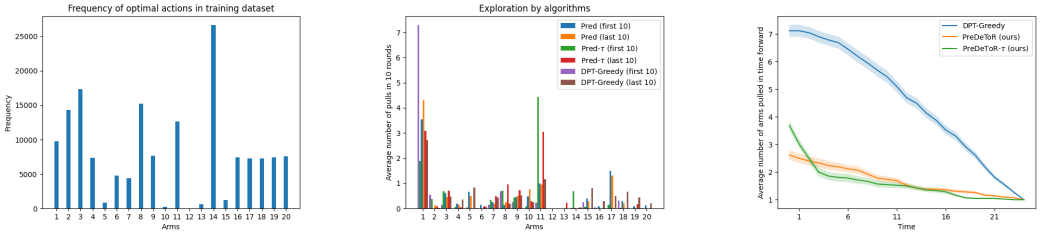
as discussed in Section E.10. From the figure Figure E.12b we see that none of the algorithms PreDeToR, PreDeToR-τ, DPT-greedy consistently pulls the action 17 more than other actions. This shows that the common underlying actions across the tasks matter for learning the epxloration.

Finally, we plot the feasible action set considered by DPT-greedy, PreDeToR, and PreDeToR-τ in Figure E.12c. To plot this graph again we consider the test tasks where the optimal action is 17. We build the candidate set the same way as before. From the Figure E.12c we see that none of the three algorithms DPT-greedy, PreDeToR, PreDeToR-τ, is able to sample the optimal action 17 sufficiently high number of times.

We also show how the prediction error of the optimal action by PreDeToR compared to LinUCB in this 1 new arm linear bandit setting. In Figure E.13a we first show how the 20 actions are distributed in the $M_{\text{test}} = 200$ test tasks. In Figure E.13a for each bar, the frequency indicates the number of tasks where the action (shown in the x-axis) is the optimal action. Then in Figure E.13b we show the prediction error of PreDeToR (-τ) for each task $m \in [M_{\text{test}}]$. The prediction error is calculated the same way as stated in Section 6.5. From the Figure E.13b we see that for most actions the prediction error is higher than LinUCB showing that the introduction of 5 new actions (and thereby decreasing the invariant action set) significantly alters the prediction error.

E.12 Data Collection Analysis

In this section, we analyze the performance of PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LinUCBwhen the weak demonstrator π^w is Thomp, LinUCB, or Uniform. We again consider the linear bandit setting discussed in Section 6.3. Recall that the linear bandit setting consist of horizon $n = 25$, $M_{\text{pre}} = 200000$, $M_{\text{test}} = 200$, $A = 10$, and $d = 2$. Finally, we show the cumulative regret by the above baselines in Figure E.14a,

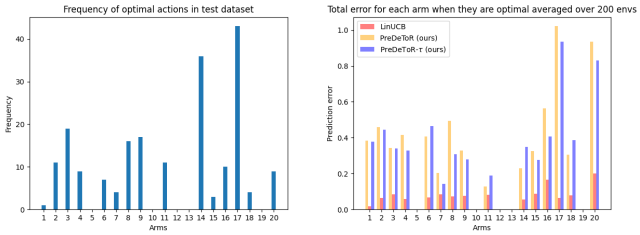


(a) Train Optimal Action Distribution

(b) Distribution of action sampling in all tasks where action 17 is optimal

(c) Candidate Action Set in Time averaged all tasks where action 17 is optimal

Figure E.12: Exploration Analysis of PreDeToR(-τ) in linear 5 new arm setting



(a) Test action distribution (b) Test Prediction Error

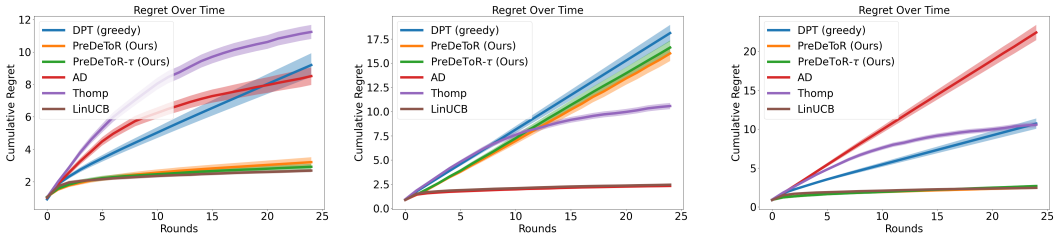
Figure E.13: Prediction error of PreDeToR(-τ) in linear 1 new arm setting

E.14b, and E.14b when data is collected through Thomp, LinUCB, and Uniform respectively.

Outcomes: We first discuss the main outcomes of our experimental results for different data collection:

> **Finding 15:** The PreDeToR (-τ) excels in exploiting the underlying latent structure and reward correlation from in-context data when the data diversity is high.

Experimental Result: We observe these outcomes in Figure E.14. In Figure E.14a we see that the A-actioned Thomp is explorative enough as



(a) Thomp data collection (b) LinUCB data collection (c) Uniform data collection

Figure E.14: Data Collection with various algorithms and Performance analysis

it does not explore with the knowledge of feature representation. So it pulls the sub-optimal actions sufficiently high number of times before discarding them in favor of the optimal action. Therefore the training data is diverse enough so that PreDeToR (-τ) can predict the reward vectors for actions sufficiently well. Consequently, PreDeToR (-τ) almost matches the LinUCB algorithm. Both DPT-greedy and AD perform poorly in this setting.

In Figure E.14b we see that the LinUCB algorithm is not explorative enough as it explores with the knowledge of feature representation and quickly discards the sub-optimal actions in favor of the optimal action. Therefore the training data is not diverse enough so that PreDeToR (-τ) is not able to correctly predict the reward vectors for actions. Note that DPT-greedy also performs poorly in this setting when it is not provided with the optimal action information during training. The AD matches the performance of its demonstrator LinUCB because of its training procedure of predicting the next action of the demonstrator.

Finally, in Figure E.14c we see that the A-armed Uniform is fully explorative as it does not intend to minimize regret (as opposed to Thomp) and does not explore with the knowledge of feature representation. Therefore the training data is very diverse which results in PreDeToR (-τ) being able to predict the reward vectors for actions very well. Consequently,

PreDeToR (-τ) perfectly matches the LinUCB algorithm. Note that AD performs the worst as it matches the performance of its demonstrator whereas the performance of DPT-greedy suffers due to the lack of information on the optimal action during training.

E.13 Empirical Validation of Theoretical Result

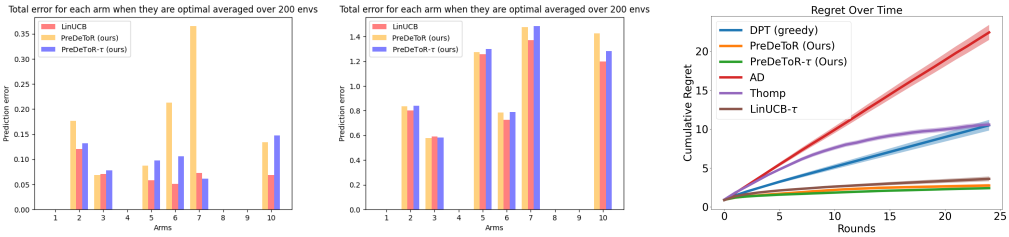
In this section, we empirically validate the theoretical result proved in Section 6.6. We again consider the linear bandit setting discussed in Section 6.3. Recall that the linear bandit setting consist of horizon $n = 25$, $M_{pre} = \{100000, 200000\}$, $M_{test} = 200$, $A = 10$, and $d = 2$. The demonstrator π^w is the Thomp algorithm and we observe that PreDeToR (-τ) has lower cumulative regret than DPT-greedy, AD and matches the performance of LinUCB.

Baseline (LinUCB-τ): We define soft LinUCB (LinUCB-τ) as follows: At every round t for task m, it calculates the ucb value $B_{m,a,t}$ for each action $\mathbf{x}_{m,a} \in \mathcal{X}$ such that $B_{m,a,t} = \mathbf{x}_{m,a}^\top \widehat{\theta}_{m,t-1} + \alpha \|\mathbf{x}_{m,a}\|_{\Sigma_{m,t-1}^{-1}}$ where $\alpha > 0$ is a constant and $\widehat{\theta}_{m,t}$ is the estimate of the model parameter $\theta_{m,*}$ at round t. Here, $\mathbf{\Sigma}_{m,t-1} = \sum_{s=1}^{t-1} \mathbf{x}_{m,s}\mathbf{x}_{m,s}^\top + \lambda \mathbf{I}_d$ is the data covariance matrix or the arms already tried. Then it chooses $I_t \sim \text{softmax}_a^\tau(B_{m,a,t})$, where $\text{softmax}_a^\tau(\cdot) \in \triangle^A$ denotes a softmax distribution over the actions and τ is a temperature parameter (See Section 6.3 for definition of $\text{softmax}_a^\tau(\cdot)$).

Outcomes: We first discuss the main outcomes of our experimental results:

> **Finding 16:** PreDeToR (-τ) excels in predicting the rewards for test tasks when the number of training (source) tasks is large.

Experimental Result: These findings are reported in Figure E.15. In Figure E.15a we show the prediction error of PreDeToR (-τ) for each task



(a) Prediction Error for 10^5 tasks (b) Prediction Error for 2×10^5 tasks (c) Cumulative Regret of PreDeToR (-τ) compared against LinUCB-τ

Figure E.15: Empirical validation of theoretical analysis

$m \in [M_{\text{test}}]$. The prediction error is calculated as $(\widehat{\mu}_{m,n,*}(a) - \mu_{m,*}(a))^2$ where $\widehat{\mu}_{m,n,*}(a) = \max_a \widehat{\theta}_{m,n}^\top \mathbf{x}_m(a)$ is the empirical mean at the end of round n, and $\mu_{*,m}(a) = \max_a \theta_{m,*}^\top \mathbf{x}_m(a)$ is the true mean of the optimal action in task m. Then we average the prediction error for the action $a \in [A]$ by the number of times the action a is the optimal action in some task m. We see that when the source tasks are 100000 the reward prediction falls short of LinUCB prediction for all actions except action 2.

In Figure E.15b we again show the prediction error of PreDeToR (-τ) for each task $m \in [M_{\text{test}}]$ when the source tasks are 200000. Note that in both these settings, we kept the horizon $n = 25$, and the same set of actions. We now observe that the reward prediction almost matches LinUCB prediction in almost all the optimal actions.

In Figure E.15c we compare PreDeToR (-τ) against LinUCB-τ and show that they almost match in the linear bandit setting discussed in Section 6.3 when the source tasks are 100000.

E.14 Empirical Study: Offline Performance

In this section, we discuss the offline performance of PreDeToR when the number of tasks $M_{\text{pre}} \gg A \geqslant n$.

We first discuss how PreDeToR (-τ) is modified for the offline setting. In the offline setting, the PreDeToR first samples a task $m \sim \mathcal{T}_{\text{test}}$, then the test dataset $\mathcal{H}_m \sim \mathcal{D}_{\text{test}}(\cdot|m)$. Then PreDeToR and PreDeToR-τ act similarly to the online setting, but based on the entire offline dataset $\mathcal{H}_m$. The full pseudocode of PreDeToR is in Algorithm 13.

Algorithm 13 **Pre**-trained **De**cision **T**rans**f**o**r**mer with **R**eward Estimation (PreDeToR)

Collecting Pretraining Dataset
Initialize empty pretraining dataset $\mathcal{H}_{\text{train}}$
for i in $[M_{\text{pre}}]$ **do**
    Sample task $m \sim \mathcal{T}_{\text{pre}}$, in-context dataset $\mathcal{H}_m \sim \mathcal{D}_{\text{pre}}(\cdot|m)$ and add this to $\mathcal{H}_{\text{train}}$.
Pretraining model on dataset
Initialize model $\mathbf{T}_{\boldsymbol{\Theta}}$ with parameters $\boldsymbol{\Theta}$
while not converged **do**
    Sample $\mathcal{H}_m$ from $\mathcal{H}_{\text{train}}$ and predict $\widehat{r}_{m,t}$ for action $(I_{m,t})$ for all $t \in [n]$
    Compute loss in (6.3) with respect to $r_{m,t}$ and backpropagate to update model parameter $\boldsymbol{\Theta}$.
Offline test-time deployment
Sample unknown task $m \sim \mathcal{T}_{\text{test}}$, sample dataset $\mathcal{H}_m \sim \mathcal{D}_{\text{test}}(\cdot|m)$
Use $\mathbf{T}_{\boldsymbol{\Theta}}$ on m at round t to choose

$$I_t \begin{cases} = \arg\max_{a \in \mathcal{A}} \mathbf{T}_{\boldsymbol{\Theta}}\left(\widehat{r}_{m,t}(a) \mid \mathcal{H}_m\right), & \textbf{PreDeToR} \\ \sim \text{softmax}_a^{\tau} \mathbf{T}_{\boldsymbol{\Theta}}\left(\widehat{r}_{m,t}(a) \mid \mathcal{H}_m\right), & \textbf{PreDeToR-}\tau \end{cases}$$

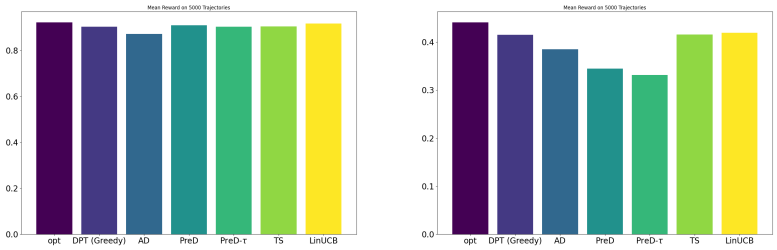
Recall that $\mathcal{D}_{\text{test}}$ denote a distribution over all possible interactions that can be generated by π^w during test time. For offline testing, first, a test task $m \sim \mathcal{T}_{\text{test}}$, and then an in-context test dataset $\mathcal{H}_m$ is collected such that $\mathcal{H}_m \sim \mathcal{D}_{\text{test}}(\cdot|m)$. Observe from Algorithm 13 that in the offline setting, PreDeToR first samples a task $m \sim \mathcal{T}_{\text{test}}$, and then a test dataset $\mathcal{H}_m \sim \mathcal{D}_{\text{test}}(\cdot|m)$ and acts greedily. Crucially in the offline setting the PreDeToR does not add the observed reward r_t at round t to the dataset. Through

this experiment, we want to evaluate the performance of PreDeToR to learn the underlying latent structure and reward correlation when the horizon is small. Finally, recall that when the horizon is small the weak demonstrator π^w does not have sufficient samples for each action. This leads to a poor approximation of the greedy action.

Baselines: We again implement the same baselines discussed in Section 6.3. The baselines are PreDeToR, PreDeToR-τ, DPT-greedy, AD, Thomp, and LinUCB. During test time evaluation for offline setting the DPT selects $I_t = \widehat{a}_{m,t,*}$ where $\widehat{a}_{m,t,*} = \arg\max_a \mathbf{T}_{\Theta}(a|\mathcal{H}_m^t)$ is the predicted optimal action.

Outcomes: We first discuss the main outcomes of our experimental results for increasing the horizon:

Finding 17: PreDeToR (-τ) performs comparably to DPT-greedy and AD in the offline setting.



(a) Offline for Linear setting (b) Offline for Non-linear setting

Figure E.16: Offline experiment. The y-axis shows the cumulative reward.

Experimental Result: We observe these outcomes in Figure E.16. In Figure E.16 we show the linear bandit setting for horizon $n = 20$, $M_{pre} = 200000$, $M_{test} = 5000$, $A = 20$, and $d = 5$ for the low data regime. Again, the demonstrator π^w is the Thomp algorithm. We observe that PreDeToR (-τ) has comparable cumulative regret to DPT-greedy. Note that for any

task m for the horizon $n = 20$ the Thomp will be able to sample all the actions at most once. In the non-linear setting of Figure E.16b the $n = 40$, $M_{pre} = 100000$, $A = 6$, $d = 2$. Observe that in all of these results, the performance of PreDeToR (-τ) is comparable with respect to cumulative regret against DPT-greedy.

E.15 Theoretical Analysis

Proof of Lemma 1

Proof. The learner collects n rounds of data following π^w. The weak demonstrator π^w only observes the $\{I_t, r_t\}_{t=1}^n$. Recall that $N_n(a)$ denotes the total number of times the action a is sampled for n rounds. Define the matrix $\mathbf{H}_n \in \mathbb{R}^{n \times A}$ where the t-th row represents the action sampled at round $t \in [n]$. The t-th row is a one-hot vector with 1 as the a-th component in the vector for $a \in [A]$. Then define the reward vector $\mathbf{Y}_n \in \mathbb{R}^n$ as the reward vector where the t-th component is the observed reward for the action I_t for $t \in [n]$. Finally define the diagonal matrix $\mathbf{D}_A \in \mathbb{R}^{A \times A}$ as in (E.1) and the estimated reward covariance matrix as $\mathbf{S}_A \in \mathbb{R}^{A \times A}$ such that $\mathbf{S}_A(a, a') = \widehat{\mu}_n(a)\widehat{\mu}_n(a')$. This matrix captures the reward correlation between the pairs of actions $a, a' \in [A]$.

Assume $\mu \sim \mathcal{N}(0, \mathbf{S}_*)$ where $\mathbf{S}_* \in \mathbb{R}^{A \times A}$. Then the observed mean vector $\mathbf{Y}_n$ is

$$\mathbf{Y}_n = \mathbf{H}_n \mu + \mathbf{H}_n \mathbf{D}_A^{1/2} \eta_n$$

where, η_n is the noise vector over the $[n]$ training data. Then the posterior mean of $\widehat{\mu}$ by Gauss Markov Theorem (Johnson et al., 2002) is given by

$$\widehat{\mu} = \mathbf{S}_* \mathbf{H}_n^\top \left(\mathbf{H}_n (\mathbf{S}_* + \mathbf{D}_A) \mathbf{H}_n^\top \right)^{-1} \mathbf{Y}_n. \tag{E.3}$$

However, the learner does not know the true reward co-variance matrix. Hence it needs to estimate the $\mathbf{S}_*$ from the observed data. Let the estimate be denoted by $\mathbf{S}_A$.

Assumption 13. *We assume that π^w is sufficiently exploratory so that each action is sampled at least once.*

The Assumption 13 ensures that the matrix $\left(\sigma_\theta^2 \mathbf{H}_n(\mathbf{S}_A + \mathbf{D}_A)\mathbf{H}_n^\top\right)^{-1}$ is invertible. Under Assumption 13, plugging the estimate $\mathbf{S}_A$ back in (E.3) shows that the average posterior mean over all the tasks is

$$\widehat{\mu} = \mathbf{S}_A \mathbf{H}_n^\top \left(\mathbf{H}_n(\mathbf{S}_A + \mathbf{D}_A)\mathbf{H}_n^\top\right)^{-1} \mathbf{Y}_n. \tag{E.4}$$

The claim of the lemma follows. □

E.16 Generalization and Transfer Learning Proof for PreDeToR

Generalization Proof

Alg is the space of algorithms induced by the transformer $\mathbf{T}_\Theta$.

Theorem E.1. **(*PreDeToR risk*)** *Suppose error stability Assumption 9 holds and assume loss function $\ell(\cdot,\cdot)$ is C-Lipschitz for all $r_t \in [0, B]$ and horizon $n \geqslant 1$. Let $\widehat{\mathbf{T}}$ be the empirical solution of (ERM) and $\mathcal{N}(\mathcal{A}, \rho, \epsilon)$ be the covering number of the algorithm space* Alg *following Definition E.2 and E.3. Then with probability at least $1 - 2\delta$, the excess Multi-task learning (MTL) risk of PreDeToR-τ is bounded by*

$$\mathcal{R}_{\text{MTL}}(\widehat{\mathbf{T}}) \leqslant 4\frac{C}{\sqrt{nM}} + 2(B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\text{Alg}, \rho, \varepsilon)/\delta)}{cnM}}$$

where, $\mathcal{N}(\text{Alg}, \rho, \varepsilon)$ is the covering number of transformer $\widehat{\mathbf{T}}$.

Proof. We consider a meta-learning setting. Let M source tasks are i.i.d. sampled from a task distribution $\mathcal{T}$, and let $\widehat{\mathbf{T}}$ be the empirical Multi-task (MTL) solution. Define $\mathcal{H}_{\text{all}} = \bigcup_{m=1}^{M} \mathcal{H}_m$. We drop the $\boldsymbol{\Theta}, \mathbf{r}$ from transformer notation $\text{TF}^{\mathbf{r}}_{\boldsymbol{\Theta}}$ as we keep the architecture fixed as in Lin et al. (2023). Note that this transformer predicts a reward vector over the actions. To be more precise we denote the reward predicted by the transformer at round t after observing history $\mathcal{H}_m^{t-1}$ and then sampling the action a_{mt} as $\mathbf{T}\left(\widehat{r}_{mt}(a_{mt})|\mathcal{H}_m^{t-1}, a_{mt}\right)$. Define the training risk

$$\widehat{\mathcal{L}}_{\mathcal{H}_{\text{all}}}(\mathbf{T}) = \frac{1}{nM} \sum_{m=1}^{M} \sum_{t=1}^{n} \ell\left(r_{mt}(a_{mt}), \mathbf{T}\left(\widehat{r}_{mt}(a_{mt})|\mathcal{H}_m^{t-1}, a_{mt}\right)\right)$$

and the test risk

$$\mathcal{L}_{\text{MTL}}(\mathbf{T}) = \mathbb{E}\left[\widehat{\mathcal{L}}_{\mathcal{H}_{\text{all}}}(\mathbf{T})\right].$$

Define empirical risk minima $\widehat{\mathbf{T}} = \arg\min_{\mathbf{T}\in\text{Alg}} \widehat{\mathcal{L}}_{\mathcal{H}_{\text{all}}}(\mathbf{T})$ and population minima

$$\mathbf{T}^* = \arg\min_{\mathbf{T}\in\text{Alg}} \mathcal{L}_{\text{MTL}}(\mathbf{T})$$

In the following discussion, we drop the subscripts MTL and $\mathcal{H}_{\text{all}}$. The excess MTL risk is decomposed as follows:

$$\begin{aligned}\mathcal{R}_{\text{MTL}}(\widehat{\mathbf{T}}) &= \mathcal{L}(\widehat{\mathbf{T}}) - \mathcal{L}(\mathbf{T}^*) \\ &= \underbrace{\mathcal{L}(\widehat{\mathbf{T}}) - \widehat{\mathcal{L}}(\widehat{\mathbf{T}})}_{a} + \underbrace{\widehat{\mathcal{L}}(\widehat{\mathbf{T}}) - \widehat{\mathcal{L}}(\mathbf{T}^*)}_{b} + \underbrace{\widehat{\mathcal{L}}(\mathbf{T}^*) - \mathcal{L}(\mathbf{T}^*)}_{c}.\end{aligned}$$

Since $\widehat{\mathbf{T}}$ is the minimizer of empirical risk, we have $b \leqslant 0$.

Step 1: (Concentration bound $|\mathcal{L}(\mathbf{T}) - \widehat{\mathcal{L}}(\mathbf{T})|$ for a fixed $\mathbf{T} \in$ Alg)

Define the test/train risks of each task as follows:

$$\widehat{\mathcal{L}}_m(\mathbf{T}) := \frac{1}{n}\sum_{t=1}^{n} \ell\left(r_{mt}(a_{mt}), \mathbf{T}\left(\widehat{r}_{mt}(a_{mt})|\mathcal{H}_m^{t-1}, a_{mt}\right)\right), \quad \text{and}$$

$$\mathcal{L}_m(\mathbf{T}) := \mathbb{E}_{\mathcal{H}_m}\left[\widehat{\mathcal{L}}_m(\mathbf{T})\right]$$

$$= \mathbb{E}_{\mathcal{H}_m}\left[\frac{1}{n}\sum_{t=1}^{n} \ell\left(r_{mt}(a_{mt}), \mathbf{T}\left(\widehat{r}_{mt}(a_{mt})|\mathcal{H}_m^{t-1}, a_{mt}\right)\right)\right], \quad \forall m \in [M].$$

Define the random variables $X_{m,t} = \mathbb{E}\left[\widehat{\mathcal{L}}_t(\mathbf{T}) \mid \mathcal{H}_m^t\right]$ for $t \in [n]$ and $m \in [M]$, that is, $X_{m,t}$ is the expectation over $\widehat{\mathcal{L}}_t(\mathbf{T})$ given training sequence $\mathcal{H}_m^t = \{(a_{mt'}, r_{mt'})\}_{t'=1}^t$ (which are the filtrations). With this, we have that $X_{m,n} = \mathbb{E}\left[\widehat{\mathcal{L}}_m(\mathbf{T}) \mid \mathcal{H}_m^n\right] = \widehat{\mathcal{L}}_m(\mathbf{T})$ and $X_{m,0} = \mathbb{E}\left[\widehat{\mathcal{L}}_m(\mathbf{T})\right] = \mathcal{L}_m(\mathbf{T})$. More generally, $(X_{m,0}, X_{m,1}, \ldots, X_{m,n})$ is a martingale sequence since, for every $m \in [M]$, we have that $\mathbb{E}\left[X_{m,i} \mid \mathcal{H}_m^{t-1}\right] = X_{m,t-1}$. For notational simplicity, in the following discussion, we omit the subscript m from a, r and $\mathcal{H}$ as they will be clear from the left-hand-side variable $X_{m,t}$. We have that

$$\begin{aligned} X_{m,t} &= \mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n} \ell\left(r_{t'}, \mathrm{TF}\left(\widehat{r}_{t'}|\mathcal{H}^{t'-1}, a_{t'}\right)\right) \middle| \mathcal{H}^t\right] \\ &= \frac{1}{n}\sum_{t'=1}^{t} \ell\left(r_{t'}, \mathrm{TF}\left(\widehat{r}_{t'}|\mathcal{H}^{t'-1}, a_{t'}\right)\right) \\ &\quad + \frac{1}{n}\sum_{t'=t+1}^{n} \mathbb{E}\left[\ell\left(r_{t'}, \mathrm{TF}\left(\widehat{r}_{t'}|\mathcal{H}^{t'-1}, a_{t'}\right)\right) \mid \mathcal{H}^t\right] \end{aligned}$$

Using the similar steps as in Li et al. (2023) we can show that

$$|X_{m,t} - X_{m,t-1}| \overset{(a)}{\leqslant} \frac{B}{n} + \sum_{t'=t+1}^{n} \frac{K}{t'n} \leqslant \frac{B + K\log n}{n}.$$

where, (a) follows by using the fact that loss function $\ell(\cdot,\cdot)$ is bounded by

B, and error stability assumption.

Recall that $\left|\mathcal{L}_m(\mathbf{T}) - \widehat{\mathcal{L}}_m(\mathbf{T})\right| = |X_{m,0} - X_{m,n}|$ and for every $m \in [M]$, we have $\sum_{t=1}^{n} |X_{m,t} - X_{m,t-1}|^2 \leqslant \frac{(B+K\log n)^2}{n}$. As a result, applying Azuma-Hoeffding's inequality, we obtain

$$\mathbb{P}\left(\left|\mathcal{L}_m(\mathbf{T}) - \widehat{\mathcal{L}}_m(\mathbf{T})\right| \geqslant \tau\right) \leqslant 2e^{-\frac{n\tau^2}{2(B+K\log n)^2}}, \quad \forall m \in [M] \tag{E.5}$$

Let us consider $Y_m := \mathcal{L}_m(\mathbf{T}) - \widehat{\mathcal{L}}_m(\mathbf{T})$ for $m \in [M]$. Then, $(Y_m)_{m=1}^{M}$ are i.i.d. zero mean sub-Gaussian random variables. There exists an absolute constant $c_1 > 0$ such that, the subgaussian norm, denoted by $\|\cdot\|_{\psi_2}$, obeys $\|Y_m\|_{\psi_2}^2 < \frac{c_1(B+K\log n)^2}{n}$ via Proposition 2.5.2 of (Vershynin, 2018). Applying Hoeffding's inequality, we derive

$$\mathbb{P}\left(\left|\frac{1}{M}\sum_{m=1}^{M} Y_t\right| \geqslant \tau\right) \leqslant 2e^{-\frac{cnM\tau^2}{(B+K\log n)^2}} \implies \mathbb{P}(|\widehat{\mathcal{L}}(\mathbf{T}) - \mathcal{L}(\mathbf{T})| \geqslant \tau) \leqslant 2e^{-\frac{cnM\tau^2}{(B+K\log n)^2}}$$

where $c > 0$ is an absolute constant. Therefore, we have that for any $\mathbf{T} \in \text{Alg}$, with probability at least $1 - 2\delta$,

$$|\widehat{\mathcal{L}}(\mathbf{T}) - \mathcal{L}(\mathbf{T})| \leqslant (B + K\log n)\sqrt{\frac{\log(1/\delta)}{cnM}} \tag{E.6}$$

Step 2: (Bound $\sup_{\mathbf{T}\in\text{Alg}} |\mathcal{L}(\mathbf{T}) - \widehat{\mathcal{L}}(\mathbf{T})|$ where Alg is assumed to be a continuous search space). Let

$$h(\mathbf{T}) := \mathcal{L}(\mathbf{T}) - \widehat{\mathcal{L}}(\mathbf{T})$$

and we aim to bound $\sup_{\mathbf{T}\in\text{Alg}} |h(\mathbf{T})|$. Following Theorem E.3, for $\varepsilon > 0$, let Alg_ε be a minimal ε-cover of Alg in terms of distance metric ρ. Therefore, Alg_ε is a discrete set with cardinality $\left|\text{Alg}_\varepsilon\right| := \mathcal{N}(\text{Alg}, \rho, \varepsilon)$. Then, we

have

$$\sup_{\mathbf{T}\in\mathrm{Alg}} |\mathcal{L}(\mathbf{T}) - \widehat{\mathcal{L}}(\mathbf{T})| \leqslant \sup_{\mathbf{T}\in\mathrm{Alg}'} \min_{\mathbf{T}\in\mathrm{Alg}_\varepsilon} |h(\mathbf{T}) - h(\mathbf{T}')| + \max_{\mathbf{T}\in\mathrm{Alg}_\varepsilon} |h(\mathbf{T})|.$$

We will first bound the quantity $\sup_{\mathbf{T}\in\mathrm{Alg}'} \min_{\mathbf{T}\in\mathrm{Alg}_\varepsilon} |h(\mathbf{T}) - h(\mathbf{T}')|$. We will utilize that loss function $\ell(\cdot,\cdot)$ is C-Lipschitz. For any $\mathbf{T} \in \mathrm{Alg}$, let $\mathbf{T} \in \mathrm{Alg}_\varepsilon$ be its neighbor following Theorem E.3. Then we can show that

$$\begin{aligned}
&\left|\widehat{\mathcal{L}}(\mathrm{TF}) - \widehat{\mathcal{L}}(\mathrm{TF}')\right| \\
&= \left| \frac{1}{nM} \sum_{m=1}^{M} \sum_{t=1}^{n} \left(\ell\left(r_{mt}(a_{mt}), \mathbf{T}\left(\widehat{r}_{mt}(a_{mt}) | \mathcal{H}_m^{t-1}, a_{mt}\right)\right) \right.\right. \\
&\qquad \left.\left. - \ell\left(r_{mt}(a_{mt}), \mathbf{T}'\left(\widehat{r}_{mt}(a_{mt}) | \mathcal{H}_m^{t-1}, a_{mt}\right)\right)\right) \right| \\
&\leqslant \frac{L}{nM} \sum_{m=1}^{M} \sum_{t=1}^{n} \left\| \mathbf{T}\left(\widehat{r}_{mt}(a_{mt}) | \mathcal{H}_m^{t-1}, a_{mt}\right) - \mathbf{T}'\left(\widehat{r}_{mt}(a_{mt}) | \mathcal{H}_m^{t-1}, a_{mt}\right) \right\|_{\ell_2} \\
&\leqslant L\varepsilon.
\end{aligned}$$

Note that the above bound applies to all data-sequences, we also obtain that for any $\mathbf{T} \in \mathrm{Alg}$,

$$|\mathcal{L}(\mathrm{TF}) - \mathcal{L}(\mathrm{TF}')| \leqslant L\varepsilon.$$

Therefore we can show that,

$$\begin{aligned}
&\sup_{\mathbf{T}\in\mathrm{Alg}} \min_{\mathbf{T}} \in \mathrm{Alg}_\varepsilon |h(\mathbf{T}) - h(\mathbf{T}\mathrm{F}')| \\
&\qquad \leqslant \sup_{\mathbf{T}\in\mathrm{Alg}} \min_{\mathbf{T}} \in \mathrm{Alg}_\varepsilon \left|\widehat{\mathcal{L}}(\mathbf{T}) - \widehat{\mathcal{L}}(\mathbf{T}')\right| + |\mathcal{L}(\mathbf{T}) - \mathcal{L}(\mathbf{T}')| \leqslant 2L\varepsilon. \quad \text{(E.7)}
\end{aligned}$$

Next we bound the second term $\max_{\mathbf{T}\in\mathrm{Alg}_\varepsilon} |h(\mathbf{T})|$. Applying union bound directly on Alg_ε and combining it with (E.6), then we will have

that with probability at least $1-2\delta$,

$$\max_{\mathbf{T}\in \mathrm{Alg}_\varepsilon} |h(\mathbf{T})| \leqslant (B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)/\delta)}{cnM}}$$

Combining the upper bound above with the perturbation bound (E.7), we obtain that

$$\max_{\mathbf{T}\in \mathrm{Alg}} |h(\mathbf{T})| \leqslant 2C\varepsilon + (B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)/\delta)}{cnM}}.$$

It follows then that

$$\mathcal{R}_{\mathrm{MTL}}(\widehat{\mathbf{T}}) \leqslant 2\sup_{\mathbf{T}\in \mathrm{Alg}} |\mathcal{L}(\mathbf{T}) - \widehat{\mathcal{L}}(\mathbf{T})| \leqslant 4C\varepsilon + 2(B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)/\delta)}{cnM}}$$

Again by setting $\varepsilon = 1/\sqrt{nM}$

$$\mathcal{L}(\widehat{\mathbf{T}}) - \mathcal{L}(\mathbf{T}^*) \leqslant \frac{4C}{\sqrt{nM}} + 2(B + K\log n)\sqrt{\frac{\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)/\delta)}{cnM}}$$

The claim of the theorem follows. □

Definition E.2. (*Covering number*) *Let* Q *be any hypothesis set and* $d(q, q') \geqslant 0$ *be a distance metric over* $q, q' \in Q$. *Then,* $\bar{Q} = \{q_1, \ldots, q_N\}$ *is an* ε*-cover of* Q *with respect to* $d(\cdot,\cdot)$ *if for any* $q \in Q$, *there exists* $q_i \in \bar{Q}$ *such that* $d(q, q_i) \leqslant \varepsilon$. *The* ε*-covering number* $\mathcal{N}(Q, d, \varepsilon)$ *is the cardinality of the minimal* ε*-cover.*

Definition E.3. (*Algorithm distance*). *Let* Alg *be an algorithm hypothesis set and* $\mathcal{H} = (a_t, r_t)_{t=1}^n$ *be a sequence that is admissible for some task* $m \in [M]$. *For any pair* $\mathbf{T}, \mathbf{T}' \in \mathrm{Alg}$, *define the distance metric* $\rho(\mathbf{T}, \mathbf{T}') := \sup_{\mathcal{H}} \frac{1}{n}\sum_{t=1}^n \left\|\mathbf{T}\left(\widehat{r}_t|\mathcal{H}^{t-1}, a_t\right) - \mathbf{T}'\left(\widehat{r}_t|\mathcal{H}^{t-1}, a_t\right)\right\|_{\ell_2}$.

Remark E.4. (***Stability Factor***) *The work of Li et al. (2023) also characterizes the stability factor* K *in Assumption 9 with respect to the transformer architecture.*

Assuming loss $\ell(\cdot,\cdot)$ is C-Lipschitz, the algorithm induced by $\mathbf{T}(\cdot)$ obeys the stability assumption with $K = 2C\left((1+\Gamma)e^{\Gamma}\right)^{L}$, where the norm of the transformer weights are upper bounded by $O(\Gamma)$ and there are L-layers of the transformer.

Remark E.5. (***Covering Number***) *From Lemma 16 of Lin et al. (2023) we have the following upper bound on the covering number of the transformer class $\mathbf{T}_{\Theta}$ as*

$$\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)) \leqslant O(L^2D^2J)$$

where L is the total number of layers of the transformer and J and, D denote the upper bound to the number of heads and hidden neurons in all the layers respectively. Note that this covering number holds for the specific class of transformer architecture discussed in section 2 of (Lin et al., 2023).

Generalization Error to New Task

Theorem E.6. (***Transfer Risk***) *Consider the setting of Theorem 6.1 and assume the source tasks are independently drawn from task distribution $\mathcal{T}$. Let $\widehat{TF}$ be the empirical solution of (ERM) and $g \sim \mathcal{T}$. Then with probability at least $1-2\delta$, the expected excess transfer learning risk is bounded by*

$$\mathbb{E}_g\left[\mathcal{R}_g(\widehat{\mathbf{T}})\right] \leqslant 4\frac{C}{\sqrt{M}} + B\sqrt{\frac{2\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)/\delta)}{M}}$$

where, $\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)$ is the covering number of transformer $\widehat{\mathbf{T}}$.

Proof. Let the target task g be sampled from $\mathcal{T}$, and the test set $\mathcal{H}_g = \{a_t, r_t\}_{t=1}^{n}$. Define empirical and population risks on g as

$$\widehat{\mathcal{L}}_g(\mathbf{T}) = \frac{1}{n}\sum_{t=1}^{n} \ell\left(r_t(a_{mt}), \mathbf{T}\left(\widehat{r}_t(a_{mt})|\mathcal{H}_g^{t-1}, a_t\right)\right)$$

and $\mathcal{L}_g(\mathbf{T}) = \mathbb{E}_{\mathcal{H}_g}\left[\widehat{\mathcal{L}}_g(\mathbf{T})\right]$. Again we drop $\boldsymbol{\Theta}$ from the transformer notation. Then the expected excess transfer risk following (ERM) is defined as

$$\mathbb{E}_g\left[\mathcal{R}_g(\widehat{\mathbf{T}})\right] = \mathbb{E}_{\mathcal{H}_g}\left[\mathcal{L}_g(\widehat{\mathbf{T}})\right] - \arg\min_{\mathbf{T}\in\mathrm{Alg}} \mathbb{E}_{\mathcal{H}_g}\left[\mathcal{L}_g(\mathbf{T})\right]. \tag{E.8}$$

where $\mathcal{A}$ is the set of all algorithms. The goal is to show a bound like this

$$\mathbb{E}_g\left[\mathcal{R}_g(\widehat{\mathbf{T}})\right] \leqslant \min_{\varepsilon \geqslant 0}\left\{4C\varepsilon + B\sqrt{\frac{2\log(\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)/\delta)}{T}}\right\}$$

where $\mathcal{N}(\mathrm{Alg},\rho,\varepsilon)$ is the covering number.

Step 1 ((Decomposition): Let $\mathbf{T}^* = \arg\min_{\mathbf{T}\in\mathrm{Alg}} \mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right]$. The expected transfer learning excess test risk of given algorithm $\widehat{\mathbf{T}} \in \mathrm{Alg}$ is formulated as

$$\begin{aligned}
&\widehat{\mathcal{L}}_m(\mathbf{T}) := \frac{1}{n}\sum_{t=1}^{n} \ell\left(r_{mt}(a_{mt}), \mathbf{T}\left(\widehat{r}_{mt}(a_{mt})|\mathcal{D}_m^{t-1}, a_{mt}\right)\right), \quad \text{and} \\
&\mathcal{L}_m(\mathbf{T}) := \mathbb{E}_{\mathcal{H}_m}\left[\widehat{\mathcal{L}}_t(\mathbf{T})\right] \\
&= \mathbb{E}_{\mathcal{H}_m}\left[\frac{1}{n}\sum_{t=1}^{n} \ell\left(r_{mt}(a_{mt}), \mathbf{T}\left(\widehat{r}_{mt}(a_{mt})|\mathcal{D}_m^{t-1}, a_{mt}\right)\right)\right], \quad \forall m \in [M].
\end{aligned}$$

Then we can decompose the risk as

$$\begin{aligned}
\mathbb{E}_g\left[\mathcal{R}_g(\widehat{\mathbf{T}})\right] &= \mathbb{E}_g\left[\mathcal{L}_g(\widehat{\mathbf{T}})\right] - \mathbb{E}_g\left[\mathcal{L}_g\left(\mathbf{T}^*\right)\right] \\
&= \underbrace{\mathbb{E}_g\left[\mathcal{L}_g(\widehat{\mathbf{T}})\right] - \widehat{\mathcal{L}}_{\mathcal{H}_{\mathrm{all}}}(\widehat{\mathbf{T}})}_{a} + \underbrace{\widehat{\mathcal{L}}_{\mathcal{H}_{\mathrm{all}}}(\widehat{\mathbf{T}}) - \widehat{\mathcal{L}}_{\mathcal{H}_{\mathrm{all}}}\left(\mathbf{T}^*\right)}_{b} \\
&\quad + \underbrace{\widehat{\mathcal{L}}_{\mathcal{H}_{\mathrm{all}}}\left(\mathbf{T}^*\right) - \mathbb{E}_g\left[\mathcal{L}_g\left(\mathbf{T}^*\right)\right]}_{c}.
\end{aligned}$$

Here since $\widehat{\mathbf{T}}$ is the minimizer of training risk, $b < 0$. Then we obtain

$$\mathbb{E}_g\left[\mathcal{R}_g(\widehat{\mathbf{T}})\right] \leqslant 2 \sup_{\mathbf{T}\in \mathrm{Alg}} \left|\mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right] - \frac{1}{M}\sum_{m=1}^{M} \widehat{\mathcal{L}}_m(\mathbf{T})\right|. \tag{E.9}$$

Step 2 (Bounding (E.9)**)**For any $\mathbf{T} \in \mathrm{Alg}$, let $X_t = \widehat{\mathcal{L}}_t(\mathbf{T})$ and we observe that

$$\mathbb{E}_{m\sim\mathcal{T}}\left[X_t\right] = \mathbb{E}_{m\sim\mathcal{T}}\left[\widehat{\mathcal{L}}_m(\mathbf{T})\right] = \mathbb{E}_{m\sim\mathcal{T}}\left[\mathcal{L}_m(\mathbf{T})\right] = \mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right]$$

Since $X_m, m \in [M]$ are independent, and $0 \leqslant X_m \leqslant B$, applying Hoeffding's inequality obeys

$$\mathbb{P}\left(\left|\mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right] - \frac{1}{M}\sum_{m=1}^{M} \widehat{\mathcal{L}}_m(\mathbf{T})\right| \geqslant \tau\right) \leqslant 2e^{-\frac{2M\tau^2}{B^2}}.$$

Then with probability at least $1-2\delta$, we have that for any $\mathbf{T} \in \mathrm{Alg}$,

$$\left|\mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right] - \frac{1}{M}\sum_{m=1}^{M} \widehat{\mathcal{L}}_m(\mathbf{T})\right| \leqslant B\sqrt{\frac{\log(1/\delta)}{2M}}. \tag{E.10}$$

Next, let Alg_ε be the minimal ε-cover of Alg following Theorem E.2, which implies that for any task $g \sim \mathcal{T}$, and any $\mathbf{T} \in \mathrm{Alg}$, there exists $\mathbf{T}' \in \mathrm{Alg}_\varepsilon$

$$\left|\mathcal{L}_g(\mathbf{T}) - \mathcal{L}_g\left(\mathbf{T}'\right)\right|, \left|\widehat{\mathcal{L}}_g(\mathbf{T}) - \widehat{\mathcal{L}}_g\left(\mathbf{T}'\right)\right| \leqslant C\varepsilon. \tag{E.11}$$

Since the distance metric following Definition 3.4 is defined by the worst-case datasets, then there exists $\mathbf{T}' \in \mathrm{Alg}_\varepsilon$ such that

$$\left|\mathbb{E}_g\left[\mathcal{L}_g(\mathbf{T})\right] - \frac{1}{M}\sum_{m=1}^{M} \widehat{\mathcal{L}}_m(\mathbf{T})\right| \leqslant 2C\varepsilon.$$

Let $\mathcal{N}(\text{Alg}, \rho, \varepsilon) = \left|\text{Alg}_\varepsilon\right|$ be the ε-covering number. Combining the above inequalities ((E.9), (E.10), and (E.11)), and applying union bound, we have that with probability at least $1 - 2\delta$,

$$\mathbb{E}_g\left[\mathcal{R}_g(\widehat{\mathbf{T}})\right] \leqslant \min_{\varepsilon \geqslant 0}\left\{4C\varepsilon + B\sqrt{\frac{2\log(\mathcal{N}(\text{Alg}, \rho, \varepsilon)/\delta)}{M}}\right\}$$

Again by setting $\varepsilon = 1/\sqrt{M}$

$$\mathcal{L}(\widehat{\mathbf{T}}) - \mathcal{L}\left(\mathbf{T}^*\right) \leqslant \frac{4C}{\sqrt{M}} + 2B\sqrt{\frac{\log(\mathcal{N}(\text{Alg}, \rho, \varepsilon)/\delta)}{cM}}$$

The claim of the theorem follows. □

Remark E.7. (*Dependence on* n) *In this remark, we briefly discuss why the expected excess risk for target task* $\mathcal{T}$ *does not depend on samples* n. *The work of Li et al.* (2023) *pointed out that the MTL pretraining process identifies a favorable algorithm that lies in the span of the* M *source tasks. This is termed as inductive bias* (*see section 4 of Li et al.* (2023)) (*Soudry et al.*, 2018; *Neyshabur et al.*, 2017). *Such bias would explain the lack of dependence of the expected excess transfer risk on* n *during transfer learning. This is because given a target task* $g \sim \mathcal{T}$, *the* $\mathbf{T}$ *can use the learnt favorable algorithm to conduct a discrete search over span of the* M *source tasks and return the source task that best fits the new target task. Due to the discrete search space over the span of* M *source tasks, it is not hard to see that, we need* $n \propto \log(M)$ *samples* (*which is guaranteed by the* M *source tasks*) *rather than* $n \propto d$ (*for the linear setting*).

E.17 Table of Notations

Notations	Definition
M	Total number of tasks
d	Dimension of the feature
$\mathcal{A}_m$	Action set of the m-th task
$\mathcal{X}_m$	Feature space of m-th task
M_{test}	Tasks for testing
M_{pre}	Total Tasks for pretraining
$\mathbf{x}(m, a)$	Feature of action a in task m
$\theta_{m,*}$	Hidden parameter for the task m
$\mathcal{T}_{\text{pre}}$	Pretraning distribution on tasks
$\mathcal{T}_{\text{test}}$	Testing distribution on tasks
n	Total horizon for each task m
$\mathcal{H}_m = \{I_t, r_t\}_{t=1}^n$	Dataset sampled for the m-th task containing n samples
$\mathcal{H}_m^t = \{I_s, r_s\}_{s=1}^t$	Dataset sampled for the m-th task containing samples from round $s = 1$ to t
$\mathbf{w}$	Transformer model parameter
$\mathbf{T}_{\mathbf{w}}$	Transformer with model parameter $\mathbf{w}$
$\mathcal{D}_{\text{pre}}$	Pretraining in-context distribution
$\mathcal{H}_{\text{train}}$	Training in-context dataset
$\mathcal{D}_{\text{test}}$	Testing in-context distribution

Table E.1: Table of Notations for PreDeToR

F APPENDIX: OPTIMAL DESIGN FOR HUMAN PREFERENCE ELICITATION

F.1 Proofs

This section contains proofs of our main claims.

Proof of theorem 7.1

We follow the sketch of the proof in Section 21.1 of Lattimore and Szepesvári (2020a) and adapt it to matrices. Before we start, we prove several helpful claims.

First, using (43) in Petersen and Pedersen (2012), we have

$$\frac{\partial}{\partial \pi(i)} f(\pi) = \frac{\partial}{\partial \pi(i)} \log\det(\mathbf{V}_\pi) = \mathbf{Tr}\left(\mathbf{V}_\pi^{-1} \frac{\partial}{\partial \pi(i)} \mathbf{V}_\pi\right)$$
$$= \mathbf{Tr}(\mathbf{V}_\pi^{-1} \mathbf{A}_i \mathbf{A}_i^\top) = \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)\,.$$

In the last step, we use the cyclic property of the trace. We define the gradient of $f(\pi)$ with respect to π as $\nabla f(\pi) = (\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i))_{i=1}^L$. Second, using basic properties of the trace, we have

$$\sum_{i=1}^L \pi(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i) = \sum_{i=1}^L \pi(i) \mathbf{Tr}(\mathbf{V}_\pi^{-1} \mathbf{A}_i \mathbf{A}_i^\top) = \mathbf{Tr}\left(\sum_{i=1}^L \pi(i) \mathbf{V}_\pi^{-1} \mathbf{A}_i \mathbf{A}_i^\top\right) \tag{F.1}$$
$$= \mathbf{Tr}\left(\mathbf{V}_\pi^{-1} \sum_{i=1}^L \pi(i) \mathbf{A}_i \mathbf{A}_i^\top\right) = \mathbf{Tr}(I_d) = d\,.$$

Finally, for any distribution π, (F.1) implies

$$g(\pi) = \max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i) \geqslant \sum_{i=1}^{L} \pi(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i) = d\,. \tag{F.2}$$

Now we are ready to start the proof.

(b) $\Rightarrow$ (a): Let π_* be a maximizer of $f(\pi)$. By first-order optimality conditions, for any distribution π, we have

$$\begin{aligned} 0 \geqslant \langle \nabla f(\pi_*), \pi - \pi_* \rangle &= \sum_{i=1}^{L} \pi(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) - \sum_{i=1}^{L} \pi_*(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) \\ &= \sum_{i=1}^{L} \pi(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) - d\,. \end{aligned}$$

In the last step, we use (F.1). Since this inequality holds for any distribution π, including Dirac at i for any $i \in [L]$, we have that $d \geqslant \max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) = g(\pi_*)$. Finally, (F.2) says that for any distribution π, $g(\pi) \geqslant d$. Therefore, π_* must be a minimizer of $g(\pi)$ and $g(\pi_*) = d$.

(c) $\Rightarrow$ (b): Note that

$$\begin{aligned} \langle \nabla f(\pi_*), \pi - \pi_* \rangle &= \sum_{i=1}^{L} \pi(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) - d \\ &\leqslant \max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) - d = g(\pi_*) - d \end{aligned}$$

holds for any distributions π and π_*. Since $g(\pi_*) = d$, we have $\langle \nabla f(\pi_*), \pi - \pi_* \rangle \leqslant 0$. Therefore, by first-order optimality conditions, π_* is a maximizer of $f(\pi)$.

(a) $\Rightarrow$ (c): This follows from the same argument as in (b) $\Rightarrow$ (a). In particular, we show there that the maximizer π_* of $f(\pi)$ is the minimizer of $g(\pi)$, and that $g(\pi_*) = d$.

To prove that $|\mathrm{supp}\,(\pi_*)| \leqslant d(d+1)/2$, we argue that π_* can be sub-

stituted for a distribution with a lower support whenever $|\text{supp}\,(\pi_*)| > d(d+1)/2$. The claim then follows by induction.

Let $S = \text{supp}\,(\pi_*)$ and suppose that $|S| > d(d+1)/2$. We start with designing a suitable family of optimal solutions. Since the space of $d \times d$ symmetric matrices has $d(d+1)/2$ dimensions, there must exist an L-dimensional vector η such that $\text{supp}\,(\eta) \subseteq S$ and

$$\sum_{i \in S} \eta(i) \mathbf{A}_i \mathbf{A}_i^\top = \mathbf{0}_{d,d}\,, \tag{F.3}$$

where $\mathbf{0}_{d,d}$ is a $d \times d$ zero matrix. Let $\pi_t = \pi_* + t\eta$ for $t \geqslant 0$. An important property of π_t is that

$$\log\det(\mathbf{V}_{\pi_t}) = \log\det\left(\mathbf{V}_{\pi_*} + t\sum_{i \in S} \eta(i) \mathbf{A}_i \mathbf{A}_i^\top\right) = \log\det(\mathbf{V}_{\pi_*})\,.$$

Therefore, π_t is also an optimal solution. However, it may not be a distribution.

We prove that $\pi_t \in \Delta^L$, for some $t > 0$, as follows. First, note that $\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) = d$ holds for any $i \in S$. Otherwise π_* could be improved. Using this observation, we have

$$\begin{aligned} d\sum_{i \in S} \eta(i) &= \sum_{i \in S} \eta(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) = \sum_{i \in S} \eta(i)\mathbf{Tr}(\mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i \mathbf{A}_i^\top) \\ &= \mathbf{Tr}\left(\mathbf{V}_{\pi_*}^{-1} \sum_{i \in S} \eta(i) \mathbf{A}_i \mathbf{A}_i^\top\right) = 0\,, \end{aligned}$$

where the last equality follows from (F.3). This further implies that $\sum_{i \in S} \eta(i) = 0$, and in turn that $\pi_t \in \Delta^L$, for as long as $\pi_t \geqslant \mathbf{0}_L$.

Finally, we take the largest feasible t, $\tau = \max\{t > 0 : \pi_t \in \Delta^L\}$, and note that π_τ has at least one more non-zero entry than π_* while having the same value. This concludes the proof.

Proof of theorem 7.2

We start by noting that for any list $i \in [L]$,

$$\sum_{a \in A_i} \|a\|^2_{\overline{\Sigma}_n^{-1}} = \mathbf{Tr}(A_i^\top \overline{\Sigma}_n^{-1} A_i) = \mathbf{Tr}\left(A_i^\top \left(\sum_{t=1}^{n}\sum_{k=1}^{K} x_{I_t,k} x_{I_t,k}^\top\right)^{-1} A_i\right)$$
$$= \frac{1}{n}\mathbf{Tr}\left(A_i^\top \left(\sum_{i=1}^{L} \pi_*(i) \sum_{k=1}^{K} x_{i,k} x_{i,k}^\top\right)^{-1} A_i\right) = \frac{1}{n}\mathbf{Tr}(A_i^\top V_{\pi_*}^{-1} A_i)\,.$$

The third equality holds because all $n\pi_*(i)$ are integers and $n > 0$. In this case, the optimal design is exact and $\overline{\Sigma}_n$ invertible, because all of its eigenvalues are positive. Now we use the definition of $g(\pi_*)$, apply theorem 7.1, and get that

$$\max_{i \in [L]} \mathbf{Tr}(A_i^\top V_{\pi_*}^{-1} A_i) = g(\pi_*) = d\,.$$

This concludes the proof.

Proof of theorem 7.3

For any list $i \in [L]$, we have

$$\begin{aligned}
&\mathbf{Tr}(A_i^\top(\widehat{\theta}_n - \theta_*)(\widehat{\theta}_n - \theta_*)^\top A_i) \\
&= \sum_{a \in A_i} (a^\top(\widehat{\theta}_n - \theta_*))^2 = \sum_{a \in A_i} (a^\top \overline{\Sigma}_n^{-1/2} \overline{\Sigma}_n^{1/2}(\widehat{\theta}_n - \theta_*))^2 \\
&\leqslant \sum_{a \in A_i} \|a\|^2_{\overline{\Sigma}_n^{-1}} \|\widehat{\theta}_n - \theta_*\|^2_{\overline{\Sigma}_n}\,,
\end{aligned}$$

where the last step follows from the Cauchy-Schwarz inequality. It follows that

$$\max_{i\in[L]} \mathbf{Tr}(\mathbf{A}_i^\top(\widehat{\theta}_n-\theta_*)(\widehat{\theta}_n-\theta_*)^\top\mathbf{A}_i) \leqslant \max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\|\mathbf{a}\|^2_{\overline{\Sigma}_n^{-1}}\|\widehat{\theta}_n-\theta_*\|^2_{\overline{\Sigma}_n}$$
$$= \underbrace{\max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\|\mathbf{a}\|^2_{\overline{\Sigma}_n^{-1}}}_{\textbf{Part I}}\underbrace{n\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n}}_{\textbf{Part II}},$$

where we use $\overline{\Sigma}_n = n\Sigma_n$ in the last step.

Part I captures the efficiency of data collection and depends on the optimal design. By theorem 7.2,

$$\max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\|\mathbf{a}\|^2_{\overline{\Sigma}_n^{-1}} = \frac{d}{n}.$$

Part II measures how the estimated model parameter $\widehat{\theta}_n$ differs from the true model parameter θ_*, under the empirical covariance matrix Σ_n. For Part II, we use theorem F.1 and get that

$$\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n} \leqslant \frac{16d+8\log(1/\delta)}{n}$$

holds with probability at least $1-\delta$. The main claim follows from combining the upper bounds on Parts I and II.

Proof of theorem 7.4

From the definition of ranking loss, we have

$$\mathbb{E}[R_n] = \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K}\mathbb{E}[\mathbb{1}\{\hat{\sigma}_{n,i}(j) > \hat{\sigma}_{n,i}(k)\}] = \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K}\mathbb{P}\left(\mathbf{x}_{i,j}^\top\widehat{\theta}_n < \mathbf{x}_{i,k}^\top\widehat{\theta}_n\right).$$

In the rest of the proof, we bound each term separately. Specifically, for any list $i \in [L]$ and items $(j,k) \in \Pi_2(K)$ in it, we have

$$\begin{aligned}
\mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\theta}_n < \mathbf{x}_{i,k}^\top \widehat{\theta}_n\right) &= \mathbb{P}\left(\mathbf{x}_{i,k}^\top \widehat{\theta}_n - \mathbf{x}_{i,j}^\top \widehat{\theta}_n > 0\right) \\
&= \mathbb{P}\left(\mathbf{x}_{i,k}^\top \widehat{\theta}_n - \mathbf{x}_{i,j}^\top \widehat{\theta}_n + \Delta_{i,j,k} > \Delta_{i,j,k}\right) \\
&= \mathbb{P}\left(\mathbf{x}_{i,k}^\top \widehat{\theta}_n - \mathbf{x}_{i,j}^\top \widehat{\theta}_n + \mathbf{x}_{i,j}^\top \theta_* - \mathbf{x}_{i,k}^\top \theta_* > \Delta_{i,j,k}\right) \\
&= \mathbb{P}\left(\mathbf{x}_{i,k}^\top (\widehat{\theta}_n - \theta_*) + \mathbf{x}_{i,j}^\top (\theta_* - \widehat{\theta}_n) > \Delta_{i,j,k}\right) \\
&\leqslant \mathbb{P}\left(\mathbf{x}_{i,k}^\top (\widehat{\theta}_n - \theta_*) > \frac{\Delta_{i,j,k}}{2}\right) + \mathbb{P}\left(\mathbf{x}_{i,j}^\top (\theta_* - \widehat{\theta}_n) > \frac{\Delta_{i,j,k}}{2}\right).
\end{aligned}$$

In the third equality, we use that $\Delta_{i,j,k} = (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \theta_*$. The last step follows from the fact that event $A + B > c$ occurs only if $A > c/2$ or $B > c/2$.

Now we bound $\mathbb{P}\left(\mathbf{x}_{i,k}^\top (\widehat{\theta}_n - \theta_*) > \Delta_{i,j,k}/2\right)$ and note that the other term can be bounded analogously. Specifically, we apply theorems F.2 and 7.2, and get

$$\mathbb{P}\left(\mathbf{x}_{i,k}^\top (\widehat{\theta}_n - \theta_*) > \frac{\Delta_{i,j,k}}{2}\right) \leqslant \exp\left[-\frac{\Delta_{i,j,k}^2}{4\|\mathbf{x}_{i,k}\|^2_{\Sigma_n^{-1}}}\right] \leqslant \exp\left[-\frac{\Delta_{i,j,k}^2 n}{4d}\right].$$

This concludes the proof.

The above approach relies on the concentration of $\mathbf{x}_{i,k}^\top (\widehat{\theta}_n - \theta_*)$, which is proved in theorem F.2. An alternative way of proving is a similar result is through the Cauchy-Schwarz inequality. This is useful when a high-probability bound on $\|\widehat{\theta}_n - \theta_*\|^2_{\Sigma_n}$ is available, as in section F.1. Specifically,

by the Cauchy-Schwarz inequality

$$\begin{aligned}\mathbb{P}\left(\mathbf{x}_{i,k}^\top(\widehat{\theta}_n-\theta_*)>\frac{\Delta_{i,j,k}}{2}\right)&\leqslant\mathbb{P}\left(\|\mathbf{x}_{i,k}\|_{\Sigma_n^{-1}}\|\widehat{\theta}_n-\theta_*\|_{\Sigma_n}>\frac{\Delta_{i,j,k}}{2}\right)\\&=\mathbb{P}\left(\|\mathbf{x}_{i,k}\|^2_{\Sigma_n^{-1}}\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n}>\frac{\Delta^2_{i,j,k}}{4}\right)\\&\leqslant\mathbb{P}\left(\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n}>\frac{\Delta^2_{i,j,k}}{4d}\right).\end{aligned}$$

In the second inequality, we use that $\|\mathbf{x}_{i,k}\|^2_{\Sigma_n^{-1}}=n\|\mathbf{x}_{i,k}\|^2_{\widehat{\Sigma}_n^{-1}}\leqslant d$, which follows from theorem 7.2. Finally, theorem F.1 says that

$$\mathbb{P}\left(\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n}\geqslant\frac{16d+8\log(1/\delta)}{n}\right)\leqslant\delta$$

holds for any $\delta>0$. To apply this bound, we let

$$\frac{16d+8\log(1/\delta)}{n}=\frac{\Delta^2_{i,j,k}}{4d}$$

and express δ. This leads to

$$\mathbb{P}\left(\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n}>\frac{\Delta^2_{i,j,k}}{4d}\right)\leqslant\delta=\exp\left[-\frac{\Delta^2_{i,j,k}n}{32d}+2d\right],$$

which concludes the alternative proof.

Proof of theorem 7.5

Following the same steps as in section F.1, we have

$$\max_{i\in[L]}\mathbf{Tr}(\mathbf{A}_i^\top(\widehat{\theta}_n-\theta_*)(\widehat{\theta}_n-\theta_*)^\top\mathbf{A}_i)\leqslant\underbrace{\max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\|\mathbf{a}\|^2_{\widehat{\Sigma}_n^{-1}}}_{\textbf{Part I}}\underbrace{\frac{K(K-1)n}{2}\|\widehat{\theta}_n-\theta_*\|^2_{\Sigma_n}}_{\textbf{Part II}}.$$

Part I captures the efficiency of data collection and depends on the optimal design. By theorem 7.2,

$$\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\mathbf{\Sigma}_n^{-1}} = \frac{d}{n}.$$

Part II measures how the estimated model parameter $\widehat{\theta}_n$ differs from the true model parameter θ_*, under the empirical covariance matrix $\mathbf{\Sigma}_n$. For Part II, we use theorem F.3 (restatement of Theorem 4.1 in Zhu et al. (2023b)) and get that

$$\|\widehat{\theta}_n - \theta_*\|^2_{\mathbf{\Sigma}_n} \leqslant \frac{CK^4(d + \log(1/\delta))}{n}$$

holds with probability at least $1 - \delta$, where $C > 0$ is some constant. The main claim follows from combining the upper bounds on Parts I and II.

Proof of theorem 7.6

Following the same steps as in section F.1, we get

$$\begin{aligned}
\mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\theta}_n < \mathbf{x}_{i,k}^\top \widehat{\theta}_n\right) &= \mathbb{P}\left(\mathbf{x}_{i,k}^\top(\widehat{\theta}_n - \theta_*) + \mathbf{x}_{i,j}^\top(\theta_* - \widehat{\theta}_n) > \Delta_{i,j,k}\right) \\
&= \mathbb{P}\left(\mathbf{z}_{i,j,k}^\top(\theta_* - \widehat{\theta}_n) > \Delta_{i,j,k}\right) \\
&\leqslant \mathbb{P}\left(\|\mathbf{z}_{i,j,k}\|_{\mathbf{\Sigma}_n^{-1}} \|\widehat{\theta}_n - \theta_*\|_{\mathbf{\Sigma}_n} > \Delta_{i,j,k}\right) \\
&= \mathbb{P}\left(\|\mathbf{z}_{i,j,k}\|^2_{\mathbf{\Sigma}_n^{-1}} \|\widehat{\theta}_n - \theta_*\|^2_{\mathbf{\Sigma}_n} > \Delta^2_{i,j,k}\right) \\
&\leqslant \mathbb{P}\left(\|\widehat{\theta}_n - \theta_*\|^2_{\mathbf{\Sigma}_n} > \frac{\Delta^2_{i,j,k}}{d}\right).
\end{aligned}$$

In the first inequality, we use the Cauchy-Schwarz inequality. In the second inequality, we use that $\|\mathbf{z}_{i,j,k}\|^2_{\mathbf{\Sigma}_n^{-1}} = n\|\mathbf{z}_{i,j,k}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}} \leqslant d$, which follows from

theorem 7.2. Finally, theorem F.3 says that

$$\mathbb{P}\left(\|\widehat{\theta}_n - \theta_*\|^2_{\Sigma_n} \geqslant \frac{CK^4(d + \log(1/\delta))}{n}\right) \leqslant \delta$$

holds for any $\delta > 0$. To apply this bound, we let

$$\frac{CK^4(d + \log(1/\delta))}{n} = \frac{\Delta^2_{i,j,k}}{d}$$

and express δ. This leads to

$$\mathbb{P}\left(\|\widehat{\theta}_n - \theta_*\|^2_{\Sigma_n} > \frac{\Delta^2_{i,j,k}}{d}\right) \leqslant \delta = \exp\left[-\frac{\Delta^2_{i,j,k} n}{CK^4 d} + d\right],$$

which concludes the proof.

F.2 Supporting Lemmas

This section contains our supporting lemmas and their proofs.

Lemma F.1. *Consider the absolute feedback model in section 7.4. Fix $\delta \in (0,1)$. Then*

$$\|\widehat{\theta}_n - \theta_*\|^2_{\Sigma_n} \leqslant \frac{16d + 8\log(1/\delta)}{n}$$

holds with probability at least $1 - \delta$.

Proof. Let $\mathbf{X} \in \mathbb{R}^{Kn \times d}$ be a matrix of Kn feature vectors in (7.7) and $\mathbf{y} \in \mathbb{R}^{Kn}$ be a vector of the corresponding Kn observations. Under 1-sub-Gaussian noise in (7.1), we can rewrite $\widehat{\theta}_n - \theta_*$ as

$$\widehat{\theta}_n - \theta_* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top (\mathbf{y} - \mathbf{X}\theta_*) = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \eta,$$

where $\eta \in \mathbb{R}^{Kn}$ is a vector of independent 1-sub-Gaussian noise. Now note that $\mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ is a fixed vector of length Kn for any fixed $\mathbf{a} \in \mathbb{R}^d$. Therefore, $\mathbf{a}^\top(\widehat{\theta}_n - \theta_*)$ is a sub-Gaussian random variable with a variance proxy

$$\mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a} = \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a} = \|\mathbf{a}\|^2_{(\mathbf{X}^\top\mathbf{X})^{-1}} = \|\mathbf{a}\|^2_{\Sigma_n^{-1}}/n\,.$$

From standard concentration inequalities for sub-Gaussian random variables (Boucheron et al., 2013),

$$\mathbb{P}\left(\mathbf{a}^\top(\widehat{\theta}_n - \theta_*) \geqslant \sqrt{\frac{2\|\mathbf{a}\|^2_{\Sigma_n^{-1}}\log(1/\delta)}{n}}\right) \leqslant \delta \tag{F.4}$$

holds for any fixed $\mathbf{a} \in \mathbb{R}^d$. To bound $\|\widehat{\theta}_n - \theta_*\|_{\Sigma_n}$, we take advantage of the fact that

$$\|\widehat{\theta}_n - \theta_*\|_{\Sigma_n} = \langle \widehat{\theta}_n - \theta_*, \Sigma_n^{1/2} A\rangle\,, \quad A = \frac{\Sigma_n^{1/2}(\widehat{\theta}_n - \theta_*)}{\|\widehat{\theta}_n - \theta_*\|_{\Sigma_n}}\,. \tag{F.5}$$

While $A \in \mathbb{R}^d$ is random, it has two important properties. First, its length is 1. Second, if it was fixed, we could apply (F.4) and would get

$$\mathbb{P}\left(\langle \widehat{\theta}_n - \theta_*, \Sigma_n^{1/2} A\rangle \geqslant \sqrt{\frac{2\log(1/\delta)}{n}}\right) \leqslant \delta\,.$$

To eliminate the randomness in A, we use a coverage argument.

Let $\mathcal{S} = \{\mathbf{a} \in \mathbb{R}^d : \|a\|_2 = 1\}$ be a unit sphere. Lemma 20.1 in Lattimore and Szepesvári (2020a) says that there exists an ε-cover $\mathcal{C}_\varepsilon \subset \mathbb{R}^d$ of $\mathcal{S}$ that has at most $|\mathcal{C}_\varepsilon| \leqslant (3/\varepsilon)^d$ points. Specifically, for any $\mathbf{a} \in \mathcal{S}$, there exists $\mathbf{y} \in \mathcal{C}_\varepsilon$ such that $\|\mathbf{a} - \mathbf{y}\|_2 \leqslant \varepsilon$. By a union bound applied to all points in

$\mathcal{C}_\varepsilon$, we have that

$$\mathbb{P}\left(\exists \mathbf{y} \in \mathcal{C}_\varepsilon : \langle \widehat{\theta}_n - \theta_*, \mathbf{\Sigma}_n^{1/2}\mathbf{y}\rangle \geqslant \sqrt{\frac{2\log(|\mathcal{C}_\varepsilon|/\delta)}{n}}\right) \leqslant \delta\,. \tag{F.6}$$

Now we are ready to complete the proof. Specifically, note that

$$\begin{aligned}
\|\widehat{\theta}_n - \theta_*\|_{\mathbf{\Sigma}_n} &\overset{(a)}{=} \max_{\mathbf{a}\in\mathcal{S}} \langle \widehat{\theta}_n - \theta_*, \mathbf{\Sigma}_n^{1/2}\mathbf{a}\rangle \\
&= \max_{\mathbf{a}\in\mathcal{S}} \min_{\mathbf{y}\in\mathcal{C}_\varepsilon} \left[\langle \widehat{\theta}_n - \theta_*, \mathbf{\Sigma}_n^{1/2}(\mathbf{a}-\mathbf{y})\rangle + \langle \widehat{\theta}_n - \theta_*, \mathbf{\Sigma}_n^{1/2}\mathbf{y}\rangle\right] \\
&\overset{(b)}{\leqslant} \max_{\mathbf{a}\in\mathcal{S}} \min_{\mathbf{y}\in\mathcal{C}_\varepsilon} \left[\|\widehat{\theta}_n - \theta_*\|_{\mathbf{\Sigma}_n}\|\mathbf{a}-\mathbf{y}\|_2 + \sqrt{\frac{2\log(|\mathcal{C}_\varepsilon|/\delta)}{n}}\right] \\
&\overset{(c)}{\leqslant} \varepsilon\|\widehat{\theta}_n - \theta_*\|_{\mathbf{\Sigma}_n} + \sqrt{\frac{2\log(|\mathcal{C}_\varepsilon|/\delta)}{n}}
\end{aligned}$$

holds with probability at least $1-\delta$. In this derivation, (a) follows from (F.5), (b) follows from the Cauchy-Schwarz inequality and (F.6), and (c) follows from the definition of ε-cover $\mathcal{C}_\varepsilon$. Finally, we rearrange the terms, choose $\varepsilon = 1/2$, and get that

$$\|\widehat{\theta}_n - \theta_*\|_{\mathbf{\Sigma}_n} \leqslant 2\sqrt{\frac{2\log(|\mathcal{C}_\varepsilon|/\delta)}{n}} \leqslant 2\sqrt{\frac{(2\log 6)d + 2\log(1/\delta)}{n}}\,.$$

This concludes the proof. □

Lemma F.2. *Consider the absolute feedback model in section 7.4. Fix* $\mathbf{x} \in \mathbb{R}^d$ *and* $\Delta > 0$*. Then*

$$\mathbb{P}\left(\mathbf{x}^\top(\widehat{\theta}_n - \theta_*) > \Delta\right) \leqslant \exp\left[-\frac{\Delta^2}{2\|\mathbf{x}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}}}\right]\,.$$

Proof. The proof is from Section 2.2 in Jamieson and Jain (2022). Let $\mathbf{X} \in \mathbb{R}^{Kn\times d}$ be a matrix of Kn feature vectors in (7.7) and $\mathbf{y} \in \mathbb{R}^{Kn}$ be a vector of the corresponding Kn observations. Under 1-sub-Gaussian noise

in (7.1), we can rewrite $\mathbf{x}^\top(\widehat{\theta}_n - \theta_*)$ as

$$\mathbf{x}^\top(\widehat{\theta}_n - \theta_*) = \underbrace{\mathbf{x}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top}_{\mathbf{w}}\eta = \mathbf{w}^\top\eta = \sum_{t=1}^{Kn}\mathbf{w}_t\eta_t\,,$$

where $\mathbf{w} \in \mathbb{R}^{Kn}$ is a fixed vector and $\eta \in \mathbb{R}^{Kn}$ is a vector of independent sub-Gaussian noise. Then, for any $\Delta > 0$ and $\lambda > 0$, we have

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{x}^\top(\widehat{\theta}_n - \theta_*) > \Delta\right) \\
&= \mathbb{P}\left(\mathbf{w}^\top\eta > \Delta\right) = \mathbb{P}\left(\exp[\lambda\mathbf{w}^\top\eta] > \exp[\lambda\Delta]\right) \\
&\overset{(a)}{\leqslant} \exp[-\lambda\Delta]\mathbb{E}\left[\exp\left[\lambda\sum_{t=1}^{Kn}\mathbf{w}_t\eta_t\right]\right] \overset{(b)}{\leqslant} \exp[-\lambda\Delta]\prod_{t=1}^{Kn}\mathbb{E}[\exp[\lambda\mathbf{w}_t\eta_t]] \\
&\overset{(c)}{\leqslant} \exp[-\lambda\Delta]\prod_{t=1}^{Kn}\exp[\lambda^2\mathbf{w}_t^2/2] = \exp[-\lambda\Delta + \lambda^2\|\mathbf{w}\|_2^2/2] \\
&\overset{(d)}{\leqslant} \exp\left[-\frac{\Delta^2}{2\|\mathbf{w}\|_2^2}\right] \overset{(e)}{=} \exp\left[-\frac{\Delta^2}{2\mathbf{x}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}}\right] \\
&= \exp\left[-\frac{\Delta^2}{2\|\mathbf{x}\|_{\bar{\Sigma}_n^{-1}}^2}\right]\,.
\end{aligned}$$

In the above derivation, (a) follows from Markov's inequality, (b) follows from independent noise, (c) follows from sub-Gaussianity, (d) follows from setting $\lambda = \Delta/\|\mathbf{w}\|_2^2$, and (e) follows from

$$\|\mathbf{w}\|_2^2 = \mathbf{x}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x} = \mathbf{x}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}\,.$$

This concludes the proof. □

Lemma F.3. *Consider the ranking feedback model in section 7.5. Fix* $\delta \in (0,1)$.

Then there exists a constant $C > 0$ *such that*

$$\|\widehat{\theta}_n - \theta_*\|_{\Sigma_n}^2 \leqslant \frac{CK^4(d + \log(1/\delta))}{n}$$

holds with probability at least $1 - \delta$.

Proof. The proof has two main steps.

Step 1: We first prove that $\ell_n(\theta)$ is strongly convex with respect to the norm $\|\cdot\|_{\Sigma_n}$ at θ_*. This means that there exists $\gamma > 0$ such that

$$\ell_n(\theta_* + \Delta) \geqslant \ell_n(\theta_*) + \langle \nabla \ell_n(\theta_*), \Delta \rangle + \gamma \|\Delta\|_{\Sigma_n}^2$$

for all perturbations Δ such that $\theta_* + \Delta \in \Theta$. To show this, we derive the Hessian of $\ell_n(\theta)$,

$$\nabla^2 \ell_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j}^{K} \sum_{k'=j}^{K} \frac{\exp[x_{I_t,\sigma_t(k)}^\top \theta + x_{I_t,\sigma_t(k')}^\top \theta]}{2\left(\sum_{\ell=j}^{K} \exp[x_{I_t,\sigma_t(\ell)}^\top \theta]\right)^2} z_{I_t,\sigma_t(k),\sigma_t(k')} z_{I_t,\sigma_t(k),\sigma_t(k')}^\top .$$

Since $\|x\| \leqslant 1$ and $\|\theta\| \leqslant 1$, we have $\exp[x^\top \theta] \in [e^{-1}, e]$, and thus

$$\frac{\exp[x_{I_t,\sigma_t(k)}^\top \theta + x_{I_t,\sigma_t(k')}^\top \theta]}{2\left(\sum_{\ell=j}^{K} \exp[x_{I_t,\sigma_t(\ell)}^\top \theta]\right)^2} \geqslant \frac{e^{-4}}{2(K-j)^2} \geqslant \frac{e^{-4}}{2K(K-1)} .$$

We further have for any $t \in [n]$ that

$$\begin{aligned}\sum_{j=1}^{K} \sum_{k=j}^{K} \sum_{k'=j}^{K} z_{I_t,\sigma_t(k),\sigma_t(k')} z_{I_t,\sigma_t(k),\sigma_t(k')}^\top &\succeq \sum_{k=1}^{K} \sum_{k'=1}^{K} z_{I_t,\sigma_t(k),\sigma_t(k')} z_{I_t,\sigma_t(k),\sigma_t(k')}^\top \\ &\succeq 2 \sum_{k=1}^{K} \sum_{k'=k+1}^{K} z_{I_t,k,k'} z_{I_t,k,k'}^\top .\end{aligned}$$

The last step follows from the fact that σ_t is a permutation. Simply put, we replace the sum of K^2 outer products by all but the ones between the

same vectors. Now we combine all claims and get

$$\nabla^2 \ell_n(\theta) \succeq \frac{e^{-4}}{K(K-1)n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbf{z}_{I_t,j,k} \mathbf{z}_{I_t,j,k}^\top = \gamma \boldsymbol{\Sigma}_n$$

for $\gamma = e^{-4}/2$. Therefore, $\ell_n(\theta)$ is strongly convex at θ_* with respect to the norm $\|\cdot\|_{\boldsymbol{\Sigma}_n}$.

Step 2: Now we rearrange the strong convexity inequality and get

$$\begin{aligned} \gamma \|\Delta\|_{\boldsymbol{\Sigma}_n}^2 &\leqslant \ell_n(\theta_* + \Delta) - \ell_n(\theta_*) - \langle \nabla \ell_n(\theta_*), \Delta \rangle \leqslant -\langle \nabla \ell_n(\theta_*), \Delta \rangle \quad \text{(F.7)} \\ &\leqslant \|\nabla \ell_n(\theta_*)\|_{\boldsymbol{\Sigma}_n^{-1}} \|\Delta\|_{\boldsymbol{\Sigma}_n} . \end{aligned}$$

In the second inequality, we use that $\hat{\theta}$ minimizes ℓ_n, and hence $\ell_n(\theta_* + \Delta) - \ell_n(\theta_*) \leqslant 0$. In the last inequality, we use the Cauchy-Schwarz inequality.

Next we write the gradient of the loss function

$$\nabla \ell_n(\theta) = -\frac{1}{n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j}^{K} \frac{\exp[\mathbf{x}_{I_t,\sigma_t(k)}^\top \theta]}{\sum_{\ell=j}^{K} \exp[\mathbf{x}_{I_t,\sigma_t(\ell)}^\top \theta]} \mathbf{z}_{I_t,\sigma_t(j),\sigma_t(k)} .$$

Zhu et al. (2023b) note that is a sub-Gaussian random variable and prove that

$$\|\nabla \ell_n(\theta_*)\|_{\boldsymbol{\Sigma}_n^{-1}}^2 \leqslant \frac{CK^4(d + \log(1/\delta))}{n}$$

holds with probability at least $1 - \delta$, where $C > 0$ is a constant. Finally, we plug the above bound into (F.7) and get that

$$\gamma \|\Delta\|_{\boldsymbol{\Sigma}_n}^2 \leqslant \sqrt{\frac{CK^4(d + \log(1/\delta))}{n}} \|\Delta\|_{\boldsymbol{\Sigma}_n}$$

holds with probability at least $1 - \delta$. We rearrange the inequality and since

γ is a constant,

$$\|\widehat{\theta}_n - \theta_*\|^2_{\Sigma_n} = \|\Delta\|^2_{\Sigma_n} \leqslant \frac{CK^4(d + \log(1/\delta))}{n}$$

holds with probability at least $1 - \delta$ for some $C > 0$. This concludes the proof. □

F.3 Optimal Design for Ranking Feedback

The optimal design for (7.10) is derived as follows. First, we compute the Hessian of $\ell_n(\theta)$,

$$\nabla^2 \ell_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j}^{K} \sum_{k'=j}^{K} \frac{\exp[\mathbf{x}^\top_{I_t,\sigma_t(k)}\theta + \mathbf{x}^\top_{I_t,\sigma_t(k')}\theta]}{2\left(\sum_{\ell=j}^{K} \exp[\mathbf{x}^\top_{I_t,\sigma_t(\ell)}\theta]\right)^2} \mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')} \mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')}.$$

In this work, we maximize the log determinant of relaxed $\nabla^2 \ell_n(\theta_*)$. Note that the exact optimization is impossible because θ_* is unknown. This can be addressed in two ways.

Worst-case design: Solve an approximation where θ_*-dependent terms are replaced with a lower bound. We take this approach. Specifically, we show in the proof of theorem F.3 that

$$\nabla^2 \ell_n(\theta) \succeq \frac{e^{-4}}{K(K-1)n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbf{z}_{I_t,j,k} \mathbf{z}^\top_{I_t,j,k} = \gamma \Sigma_n$$

for $\gamma = e^{-4}/2$. Then we maximize the log determinant of a relaxed problem. This solution is sound and justified, because we maximize a lower bound on the original objective.

Plug-in design: Solve an approximation where θ_* is replaced with a plug-in estimate.

We discuss the pluses and minuses of both approaches next.

Method	Maximum prediction error	Ranking loss
Dope (ours)	15.79 ± 1.08	0.107 ± 0.002
Plug-in (400)	19.75 ± 1.48	0.104 ± 0.003
Plug-in (300)	30.52 ± 3.00	0.103 ± 0.002
Plug-in (200)	65.75 ± 13.71	0.114 ± 0.003
Plug-in (100)	100.39 ± 10.72	0.142 ± 0.003
Optimal	9.22 ± 0.82	0.092 ± 0.002

Table F.1: Comparison of Dope with plug-in designs Plug-in and optimal solution Optimal.

Prior works: Mason et al. (2022) used a plug-in estimate to design a cumulative regret minimization algorithm for logistic bandits. Recent works on preference-based learning (Zhu et al., 2023b; Das et al., 2024; Zhan et al., 2024), which are the closet related works, used worst-case designs. Interestingly, Das et al. (2024) analyzed an algorithm with a plug-in estimate but empirically evaluated a worst-case design. This indicates that their plug-in design is not practical.

Ease of implementation: Worst-case designs are easier to implement. This is because the plug-in estimate does not need to be estimated. Note that this requires solving an exploration problem with additional hyper-parameters, such as the number of exploration rounds for the plug-in estimation.

Theory: Worst-case designs can be analyzed similarly to linear models. Plug-in designs require an analysis of how the plug-in estimate concentrates. The logging policy for the plug-in estimate can be non-trivial as well. For instance, the plug-in estimate exploration in Mason et al. (2022) is over $\tilde{O}(d)$ special arms, simply to get pessimistic per-arm estimates. Their exploration budget is reported in Table 1. The lowest one, for a 3-dimensional problem, is 6 536 rounds. This is an order of magnitude more than in our fig. 7.1b for a larger 36-dimensional problem.

Based on the above discussion, we believe that worst-case designs strike

a good balance between *practicality and a theory support*. Nevertheless, plug-in designs are intriguing because they may perform well with a decent plug-in estimate. To investigate if this happens in our experiments, we repeat Experiment 2 in section 7.6 with $K = 2$ (logistic regression). We compare three methods:

(1) Dope: This is our method. The exploration policy is π_* in (7.6).

(2) Plug-in (m): This is a plug-in baseline. For the first m rounds, it explores using policy π_* in (7.6). After that, we compute the plug-in estimate of θ_* using (7.10) and solve the D-optimal design with it. The resulting policy is used for exploration for the remaining $n - m$ rounds. Finally, θ_* is estimated from all feedback using (7.10).

(3) Optimal: The exploration policy π_* is computed using the D-optimal design with the unknown θ_*. This validates our implementation and also shows the gap from the optimal solution.

We report both the prediction errors and ranking losses at $n = 500$ rounds in table F.1. The results are averaged over 100 runs. We observe that the prediction error of Dope is always smaller than that of Plug-in (6 times at $m = 100$). Optimal outperforms Dope but cannot be implemented in practice. The gap between Optimal and Plug-in shows that an optimal design with a plug-in estimate of θ_* can be much worse than that with θ_*. Dope has a comparable ranking loss to Plug-in at $m = 300$ and $m = 400$. Plug-in has a higher ranking loss otherwise.

Based on our discussion and experiments, we do not see any strong evidence to adopt plug-in designs. They would be more complex than worst-case designs, harder to analyze, and we do not see benefits in our experiments. This also follows the principle of Occam's razor, which tells us to design with a minimal needed complexity.

Number of lists L	100	200	300	400	500	600	700	800
Time (seconds)	4.71	8.31	15.63	21.00	26.60	35.00	41.25	49.72

Table F.2: Computation time of policy π_* in (7.6) as a function of the number of lists L.

	$L = 50$	$L = 100$	$L = 200$	$L = 500$
$K = 2$	0.12 ± 0.06	0.28 ± 0.12	0.37 ± 0.14	0.57 ± 0.21
$K = 3$	0.14 ± 0.06	0.24 ± 0.10	0.37 ± 0.15	0.50 ± 0.19
$K = 4$	0.13 ± 0.05	0.24 ± 0.08	0.35 ± 0.14	0.47 ± 0.18
$K = 5$	0.12 ± 0.04	0.21 ± 0.08	0.34 ± 0.12	0.45 ± 0.15

Table F.3: The ranking loss of Dope as a function of the number of lists L and items K.

F.4 Ablation Studies

We conduct two ablation studies on Experiment 2 in section 7.6.

In table F.2, we report the computation time of policy π_* in (7.6). We vary the number of lists L and use CVXPY (Diamond and Boyd, 2016) to solve (7.6). For $L = 100$, the computation takes 4 seconds; and for $L = 800$, it takes 50. Therefore, it scales roughly linearly with the number of lists L.

In table F.3, we vary the number of lists L and items K, and report the ranking loss of Dope. We observe that the problems get harder as L increases (more lists to rank) and easier as K increases (longer lists with more feedback).

G APPENDIX: OPTIMAL DESIGN FOR ADAPTIVE IN-CONTEXT PROMPT DESIGN IN LARGE LANGUAGE MODELS

G.1 Proofs

This section contains the properties of our objective and proofs of our main claims.

Properties of our objective

By the total variance decomposition for a linear model with observation variance σ^2, we get

$$\begin{aligned}\max_{k\in[K]} \operatorname{var}[Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}] &= \max_{k\in[K]} \operatorname{var}[\mathbb{E}[Y_{*,k} \mid \mathbf{x}_{*,k}, \theta_*, H_{T+1}] \mid \mathbf{x}_{*,k}, H_{T+1}] \\ &\quad + \max_{k\in[K]} \mathbb{E}[\operatorname{var}[Y_{*,k} \mid \mathbf{x}_{*,k}, \theta_*, H_{T+1}] \mid \mathbf{x}_{*,k}, H_{T+1}] \\ &= \max_{k\in[K]} \sigma^2 \mathbf{x}_{*,k} \mathbf{T} \widehat{\mathbf{\Sigma}}_t^{-1} \mathbf{x}_{*,k} + \sigma^2 \,.\end{aligned}$$

Therefore, the variance minimization of $\max_{k\in[K]} Y_{*,k} \mid \mathbf{x}_{*,k}, H_{T+1}$ is a combinatorial optimization problem. It can be formulated as follows

$$S_* = \arg\min_{S\subseteq[n],|S|=T} f(S)\,, \tag{G.1}$$

where the function $f(S) = \max_{k\in[K]} \mathbf{x}_{*,k} \mathbf{T} \left(\mathbf{\Sigma}_0^{-1} + \sigma^{-2} \sum_{i\in S} \mathbf{x}_i \mathbf{x}_i \mathbf{T}\right)^{-1} \mathbf{x}_{*,k}$ represents the uncertainty associated with any subset S, and S_* denotes the optimal subset of size T. If $f(S)$ was monotone and supermodular in S, we would have $(1-1/e)$-optimality guarantees for GO (Nemhauser et al., 1978). We start with proving the monotonicity of $f(S)$.

Lemma G.1. *Function $f(S)$ is monotonically decreasing.*

The claim is proved in section G.1. Now we state the definition of a supermodular function and show that $f(S)$ is not supermodular.

Definition G.2 (Supermodular function)**.** *Let* $[n]$ *be a set of* n *elements. A function* $f : 2^{[n]} \to \mathbb{R}$ *is supermodular if it satisfies the property of diminishing marginal returns. For any* $S_1 \subseteq S_2 \subseteq [n]$ *and* $\mathbf{x} \notin S_2$*, we have* $f(S_1 \cup \{\mathbf{x}\}) - f(S_1) \leqslant f(S_2 \cup \{\mathbf{x}\}) - f(S_2)$.

Lemma G.3. *The function* $f(S)$ *is not supermodular.*

Proof. Take $\boldsymbol{\Sigma}_0 = I_d$, fix $K = 1$, $\mathbf{x}_1 = (1/\sqrt{2}, 1/\sqrt{2})$, $\mathbf{x}_2 = (0, 1)$, and $\mathbf{x}_* = (1, 0)$. Then $f(\{2\}) - f(\emptyset) = 0 < 0.035714 \approx f(\{1, 2\}) - f(\{1\})$. □

Proof of theorem G.1

Recall that the objective function is

$$f(S) = \max_{k \in [K]} \mathbf{x}_{*,k} \mathbf{T} \left(\boldsymbol{\Sigma}_0^{-1} + \sigma^{-2} \sum_{i \in S} \mathbf{x}_i \mathbf{x}_i \mathbf{T} \right)^{-1} \mathbf{x}_{*,k}.$$

Without loss of generality, let $\sigma^2 = 1$. Fix any $j \in [n] \setminus S$ and let $S_+ = S \cup \{j\}$. Then the objective value at S_+ can be written as

$$f(S_+) = \max_{k \in [K]} \mathbf{x}_{*,k} \mathbf{T} \left(\boldsymbol{\Sigma}_0^{-1} + \sum_{i \in S_+} \mathbf{x}_i \mathbf{x}_i \mathbf{T} \right)^{-1} \mathbf{x}_{*,k} = \max_{k \in [K]} \mathbf{x}_{*,k} \mathbf{T} \left(\mathbf{A} + \mathbf{x}_j \mathbf{x}_j \mathbf{T} \right)^{-1} \mathbf{x}_{*,k},$$

where $\mathbf{A} = \boldsymbol{\Sigma}_0^{-1} + \sum_{i \in S} \mathbf{x}_i \mathbf{x}_i \mathbf{T}$. By the Sherman–Morrison formula, we have

$$\left(\mathbf{A} + \mathbf{x}_j \mathbf{x}_j \mathbf{T} \right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \mathbf{x}_j \mathbf{x}_j \mathbf{T} \mathbf{A}^{-1}}{1 + \mathbf{x}_j \mathbf{T} \mathbf{A}^{-1} \mathbf{x}_j}.$$

Now note that $\frac{\mathbf{A}^{-1}\mathbf{x}_j\mathbf{x}_j\mathbf{T}\mathbf{A}^{-1}}{1+\mathbf{x}_j\mathbf{T}\mathbf{A}^{-1}\mathbf{x}_j}$ is a positive semi-definite matrix for any $\mathbf{x}_j$ and any positive semi-definite matrix $\mathbf{A}$. As a result, $\mathbf{x}_{*,k}\mathbf{T}\frac{\mathbf{A}^{-1}\mathbf{x}_j\mathbf{x}_j\mathbf{T}\mathbf{A}^{-1}}{1+\mathbf{x}_j\mathbf{T}\mathbf{A}^{-1}\mathbf{x}_j}\mathbf{x}_{*,k} \geqslant 0$ holds for any $\mathbf{x}_{*,k}$ and thus

$$\begin{aligned} f(S_+) &= \max_{k\in[K]} \mathbf{x}_{*,k}\mathbf{T}\mathbf{A}^{-1}\mathbf{x}_{*,k}\mathbf{T} - \mathbf{x}_{*,k}\mathbf{T}\frac{\mathbf{A}^{-1}\mathbf{x}_j\mathbf{x}_j\mathbf{T}\mathbf{A}^{-1}}{1+\mathbf{x}_j\mathbf{T}\mathbf{A}^{-1}\mathbf{x}_j}\mathbf{x}_{*,k} \\ &\leqslant \max_{k\in[K]} \mathbf{x}_{*,k}\mathbf{T}\mathbf{A}^{-1}\mathbf{x}_{*,k}\mathbf{T} = f(S)\,. \end{aligned}$$

This proves our claim.

Proof of theorem 8.1

The proof is under the assumption that at round t, the training examples can be partitioned as $\mathcal{X}_{\text{examples}} = S_k \cup \mathbf{S}_k$. The set S_k represents examples that are close to $\mathbf{x}_{*,k}$. The set S_k is convex and define $\alpha_k \geqslant 0$ such that $\mathbf{x}\mathbf{T}\mathbf{y} \geqslant \alpha_k$ for all $\mathbf{x}, \mathbf{y} \in S_k$. The set $\mathbf{S}_k$ represents examples that are not close to $\mathbf{x}_{*,k}$. It is defined $\beta_k \geqslant 0$ such that $\mathbf{x}\mathbf{T}\mathbf{y} \leqslant \beta_k$ for all $\mathbf{x} \in S_k$ and $\mathbf{y} \in \mathbf{S}_k$. Define $\alpha_{\min} = \min_k \alpha_k$, and $\beta_{\max} = \beta_{\max}$.

Define the set $S = \cap_{k=1}^K S_k$ as the set of all examples that are close to all $\{\mathbf{x}_{*,k}\}_{k=1}^K$ and $\overline{S} = \cup_{k=1}^K \mathbf{S}_k$ as the set of all examples that are not close to all $\{\mathbf{x}_{*,k}\}_{k=1}^K$. Assume $S \neq \{\emptyset\}$ and $|S| > T$.

The inverse of the covariance matrix after $t-1$ observations is

$$\widehat{\mathbf{\Sigma}}_t^{-1} = \mathbf{\Lambda}_t = \mathbf{I}_d + \sum_{\ell=1}^{t-1} X_\ell X_\ell \mathbf{T} = \sum_{i=1}^{d} \lambda_{t,i}\mathbf{x}_{t,i}\mathbf{x}_{t,i}\mathbf{T}\,.$$

The latter is the eigendecompositon of $\mathbf{\Lambda}_t$, where $\lambda_{t,i}$ is the i-th largest eigenvalue and $\mathbf{x}_{t,i}$ is the corresponding eigenvector. Note that $\mathbf{\Lambda}_t^{-1} = \sum_{i=1}^d \lambda_{t,i}^{-1}\mathbf{x}_{t,i}\mathbf{x}_{t,i}\mathbf{T}$. To simplify exposition, we assume that all examples have unit length. We analyze the eigenvalues of $\mathbf{\Lambda}_t$ first.

Lemma G.4. *For all* $i \in [d]$, $1 \leqslant \lambda_{t,i} \leqslant t$. *Moreover, let* $X_\ell \in S$ *hold for all* $\ell \in [t-1]$. *Then* $\lambda_{t,1} \geqslant \alpha_{\min}^2(t-1)+1$.

Proof. The first claim follows directly from the definition of $\mathbf{\Lambda}_t$ and that $\|X_\ell\|_2 \leqslant 1$. The second claim is proved using the definition of the maximum eigenvalue,

$$\lambda_{t,1} = \mathbf{x}_{t,1}\mathbf{T}\mathbf{\Lambda}_t\mathbf{x}_{t,1} \geqslant \mathbf{x}_{*,k}\mathbf{T}\left(\mathbf{I}_d + \sum_{\ell=1}^{t-1} X_\ell X_\ell \mathbf{T}\right)\mathbf{x}_{*,k} \geqslant \alpha_{\min}^2(t-1)+1\,.$$

The last inequality follows from $\mathbf{x}_{*,k}\mathbf{T}X_\ell \geqslant \alpha_{\min}$. This completes the proof. □

We continue with claims about the eigenvectors of $\mathbf{\Lambda}_t$.

Lemma G.5. *Let* $X_\ell \in S$ *hold for all* $\ell \in [t-1]$. *Then* $\mathbf{x}_{t,1} \in S$. *Moreover, let* $\beta_{\max} \geqslant 1-\alpha_{\min}^2$. *Then* $\mathbf{x}_{t,i} \in \overline{S}$ *for all* $i \geqslant 2$.

Proof. Since all $X_\ell \in S$ and S is a convex set, $\mathbf{x}_{t,1} \in S$. Now take any $\mathbf{x} \in S$ and $i \geqslant 2$, and note that

$$(\mathbf{x}\mathbf{T}\mathbf{x}_{t,i})^2 \leqslant \sum_{i=2}^{d}(\mathbf{x}\mathbf{T}\mathbf{x}_{t,i})^2 = 1-(\mathbf{x}\mathbf{T}\mathbf{x}_{t,1})^2 \leqslant 1-\alpha_{\min}^2\,.$$

We use that $\sum_{i=1}^{d}(\mathbf{x}\mathbf{T}\mathbf{x}_{t,i})^2 = 1$ and $\mathbf{x}\mathbf{T}\mathbf{x}_{t,1} \geqslant \alpha_{\min}$. Therefore, when $\beta_{\max} \geqslant 1-\alpha_{\min}^2$, we have $\mathbf{x}_{t,i} \in \overline{S}$ for all $i \geqslant 2$. □

Our analysis has two parts. First, we bound the approximation error under the assumption that $X_\ell \in S$ holds for all $\ell \in [T]$. Second, we show how to choose $\alpha_{\min}$ and $\beta_{\max}$ to guarantee this. We start with the approximation error.

Lemma G.6. *Let $X_\ell \in S$ hold for all $\ell \in [T]$. Then for any $\mathbf{x}_{*,k}$*

$$\mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_{T+1}^{-1}\mathbf{x}_{*,k} \leqslant \frac{1}{\alpha_{\min}^2 T + 1} + (1 - \alpha_{\min}^2)\,.$$

Proof. We start with

$$\begin{aligned}\mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_{T+1}^{-1}\mathbf{x}_{*,k} &= \sum_{i=1}^{d} \lambda_{T+1,i}^{-1}\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{T+1,i}\mathbf{x}_{T+1,i}\mathbf{T}\mathbf{x}_{*,k} \\ &\leqslant \frac{(\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{T+1,1})^2}{\alpha_{\min}^2 T + 1} + \sum_{i=2}^{d} (\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{T+1,i})^2\,.\end{aligned}$$

The inequality uses lower bounds in theorem G.4. Then we apply $\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{T+1,1} \leqslant 1$ and $\sum_{i=2}^{d}(\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{T+1,i})^2 \leqslant 1 - \alpha_{\min}^2$. □

Now we prove by induction that $X_\ell \in S$ holds for all $\ell \in [T]$.

Lemma G.7. *Let $X_\ell \in S$ hold for all $\ell \in [t-1]$. Suppose that $t \leqslant \dfrac{\alpha_{\min}^2}{(\beta + \sqrt{2})\beta d}$. Then $\mathbf{x}_t \in S$.*

Proof. Our algorithm chooses a example $\mathbf{x}_t \in S$ when for all $\mathbf{x}_{*,k}$

$$\frac{\mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}\mathbf{x}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}_{*,k}}{1 + \mathbf{x}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}} \geqslant \frac{\mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{y}\mathbf{y}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}_{*,k}}{1 + \mathbf{y}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{y}}$$

holds for any $\mathbf{x} \in S$ and $\mathbf{y} \in \overline{S}$. Since $0 \leqslant \mathbf{x}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x} \leqslant 1$ for any $\|\mathbf{x}\|_2 \leqslant 1$, the above event occurs when

$$\begin{aligned}\min_k (\mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x})^2 &= \min_k \mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}\mathbf{x}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}_{*,k} \\ &\geqslant 2\max_{k\in[K]} \mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{y}\mathbf{y}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{x}_{*,k} = 2\max_{k\in[K]} (\mathbf{x}_{*,k}\mathbf{T}\mathbf{\Lambda}_t^{-1}\mathbf{y})^2\,.\end{aligned}$$

We start with an upper bound on the right-hand side,

$$\max_{k\in[K]} |\mathbf{x}_{*,k}\mathbf{T}\boldsymbol{\Lambda}_t^{-1}\mathbf{y}| \leqslant \max_{k\in[K]} \sum_{i=1}^{d} \lambda_{t,i}^{-1} |\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{t,i}\mathbf{x}_{t,i}\mathbf{T}\mathbf{y}| \leqslant \beta_{max} d\,.$$

Here we use $\lambda_{t,i} \geqslant 1$ (theorem G.4), and that $\mathbf{x}_{t,1}\mathbf{T}\mathbf{y} \leqslant \beta_{max}$ and $\mathbf{x}_*\mathbf{T}\mathbf{x}_{t,i} \leqslant \beta_{max}$ when $i \geqslant 2$.

Now we bound the left-hand side as

$$\begin{aligned}\min_k |\mathbf{x}_{*,k}\mathbf{T}\boldsymbol{\Lambda}_t^{-1}\mathbf{x}| &\geqslant \min_k \lambda_{t,1}^{-1} |\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{t,1}\mathbf{x}_{t,1}\mathbf{T}\mathbf{x}| - \sum_{i=2}^{d} \lambda_{t,i}^{-1} |\mathbf{x}_{*,k}\mathbf{T}\mathbf{x}_{t,i}\mathbf{x}_{t,i}\mathbf{T}\mathbf{x}| \\ &\geqslant \frac{\alpha_{min}^2}{t} - \beta_{max}^2 d\,.\end{aligned}$$

To bound the first term, we use $\lambda_{t,1} \leqslant t$, and that $\mathbf{x}_*\mathbf{T}\mathbf{x}_{t,1} \geqslant \alpha_{min}$ and $\mathbf{x}_{t,1}\mathbf{T}\mathbf{x} \geqslant \alpha_{min}$. To bound the second term, we use $\lambda_{t,i} \geqslant 1$, and that $\mathbf{x}_*\mathbf{T}\mathbf{x}_{t,i} \leqslant \beta_{max}$ and $\mathbf{x}_{t,i}\mathbf{T}\mathbf{x} \leqslant \beta_{max}$.

Now we chain all inequalities and get that our algorithm chooses a example $\mathbf{x}_t \in S$ when

$$t \leqslant \frac{\alpha_{min}^2}{(\beta_{max} + \sqrt{2})\beta_{max} d}\,.$$

This completes the proof. □

Proof of theorem 8.2

Consider the test examples $\{\mathbf{x}_*\}_{k=1}^K$. Recall that $H_t = (X_\ell, Y_\ell)_{\ell\in[t-1]}$ is the history till round t. Then the posterior variance is

$$\widehat{\boldsymbol{\Sigma}}_t = \left(\boldsymbol{\Sigma}_0^{-1} + \sigma^{-2}\sum_{\ell=1}^{t-1} X_\ell X_\ell \mathbf{T}\right)^{-1}$$

and $\widehat{\theta}_t = \widehat{\Sigma}_t \left(\Sigma_0^{-1} \theta_0 + \sigma^{-2} \sum_{\ell=1}^{t-1} X_\ell Y_\ell \right)$. Now fix a X_t at round t. Then $Y_t = X_t^\top \theta_* + \epsilon_t$ where $\theta_* \mid H_t \sim \mathcal{N}\left(\widehat{\theta}_t, \widehat{\Sigma}_t\right)$ and $\epsilon_t \sim \mathcal{N}\left(0, \sigma^2\right)$. Then $\theta_* | H_{t+1} \sim \mathcal{N}\left(\widehat{\theta}_{t+1}, \widehat{\Sigma}_{t+1}\right)$ holds for any Y_t.

Now fix an $\mathbf{x}_i \in \mathcal{X}_{\text{examples}}$ and add it to $\widehat{\Sigma}_t$ such that

$$\widehat{\Sigma}_{t,i} = \left(\Sigma_0^{-1} + \sigma^{-2} \left(\sum_{\ell=1}^{t-1} X_\ell X_\ell \mathbf{T} + \mathbf{x}_i \mathbf{x}_i^\top \right) \right)^{-1}.$$

Define $\hat{\sigma}^2_{t,i,k} = \frac{1}{m} \sum_{j=1}^{m} (\tilde{Y}^{(j,1)}_{t,i,k} - \tilde{Y}^{(j,2)}_{t,i,k})^2$ and the $\mathbb{E}[\hat{\sigma}^2_{t,i,k}] = \sigma^2_{t,i,k}$. Then define $\sigma^2_{t,i,\max} = \max_{k \in [K]} \mathbb{E}[\hat{\sigma}^2_{t,i,k}]$. Then using Theorem G.8 we can show that with probability $(1-\delta)$

$$\sigma^2_{t,i,\max} \left[1 - 2\sqrt{\frac{\log(1/\delta)}{m}} \right] \leqslant \max_{k \in [K]} \hat{\sigma}^2_{t,i,k} \leqslant \sigma^2_{t,i,\max} \left[1 + 2\sqrt{\frac{\log(1/\delta)}{m}} + \frac{2\log(1/\delta)}{m} \right] \tag{G.2}$$

Set $m \geqslant 8 \log(1/\delta)$ in (G.2). It follows then that

$$\frac{1}{2} \sigma^2_{t,i,\max} \leqslant \max_{k \in [K]} \frac{1}{m} \sum_{j=1}^{m} (\widetilde{Y}^{(j,1)}_{t,i,k} - \widetilde{Y}^{(j,2)}_{t,i,k})^2 \leqslant \frac{5}{2} \sigma^2_{t,i,\max}.$$

Hence, to minimize the quantity $\max_{k \in [K]} \widetilde{Y}^{(j,1)}_{t,i,k} - \widetilde{Y}^{(j,2)}_{t,i,k}$ we should be minimizing the variance $2 \max_{k \in [K]} \mathbf{x}_{*,k}^\top \widehat{\Sigma}_{t,i} \mathbf{x}_{*,k} + \sigma^2$. Observe that minimizing the variance in SAL leads to minimizing the quantity $2 \max_{k \in [K]} \mathbf{x}_{*,k}^\top \widehat{\Sigma}_{t,i} \mathbf{x}_{*,k} + \sigma^2$ which is same as minimizing the score $\max_{k \in [K]} \mathbf{x}_{*,k}^\top \widehat{\Sigma}_{t,i} \mathbf{x}_{*,k}$ for GO. The claim of the theorem follows.

Lemma G.8. *Fix round $t \in [T]$, a sample example $\mathbf{x}_i$, and a test example $\mathbf{x}_{*,k}$, and failure probability $\delta \in (0,1)$. Suppose that $m > 4 \log(1/\delta)$. Define $\hat{\sigma}^2_{t,i,k} =$*

$\frac{1}{m}\sum_{j=1}^{m}(\tilde{Y}_{t,i,k}^{(j,1)} - \tilde{Y}_{t,i,k}^{(j,2)})^2$ *and the* $\mathbb{E}[\hat{\sigma}_{t,i,k}^2] = \sigma_{i,k}^2$. *Then*

$$\mathbb{P}\left(\hat{\sigma}_{t,i,k}^2 \leqslant \sigma_{i,k}^2\left[1 - 2\sqrt{\frac{\log(1/\delta)}{m}}\right]\right) \leqslant \delta$$

holds with probability at least $1-\delta$. *Analogously,*

$$\mathbb{P}\left(\hat{\sigma}_{t,i,k}^2 \geqslant \sigma_{i,k}^2\left[1 + 2\sqrt{\frac{\log(1/\delta)}{m}} + \frac{2\log(1/\delta)}{m}\right]\right) \leqslant \delta$$

holds with probability at least $1-\delta$.

Proof. Fix an $\mathbf{x}_i \in \mathcal{X}_{\text{examples}}$ and add it to $\widehat{\boldsymbol{\Sigma}}_t$. Denote this new co-variance matrix as $\widehat{\boldsymbol{\Sigma}}_{t,i}$ such that

$$\widehat{\boldsymbol{\Sigma}}_{t,i} = \left(\boldsymbol{\Sigma}_0^{-1} + \sigma^{-2}\left(\sum_{\ell=1}^{t-1} X_\ell X_\ell \mathbf{T} + \mathbf{x}_i \mathbf{x}_i^\top\right)\right)^{-1}.$$

Let $\widetilde{Y}_{t,i,j}^{(1)} = \mathbf{x}_{*,k}^\top \theta_* + \epsilon_{t,i,j,1}$, where $\theta_* \mid H_t \sim \mathcal{N}\left(\widehat{\theta}_t, \widehat{\boldsymbol{\Sigma}}_{t+1}\right)$ and $\epsilon_{t,i,j,1} \sim \mathcal{N}\left(0, \sigma^2\right)$. This yields that $\widetilde{Y}_{t,i,j}^{(1)} \sim \mathcal{N}\left(\mathbf{x}_{*,k}^\top \widehat{\theta}_{t+1}, \mathbf{x}_{*,k}^\top \widehat{\boldsymbol{\Sigma}}_{t,i} \mathbf{x}_{*,k} + \sigma^2\right)$. Similarly $\widetilde{Y}_{t,i,j}^{(2)} = \mathbf{x}_{*,k}^\top \theta_* + \epsilon_{t,i,j,2}$, where $\theta_* \mid H_t \sim \mathcal{N}\left(\widehat{\theta}_{t+1}, \widehat{\boldsymbol{\Sigma}}_{t,i}\right)$ and $\epsilon_{t,i,j,2} \sim \mathcal{N}\left(0, \sigma^2\right)$. Therefore, we get that

$$\widetilde{Y}_{t,i,k}^{(j,1)} - \widetilde{Y}_{t,i,k}^{(j,2)} \sim \mathcal{N}(0, 2\mathbf{x}_{*,k}^\top \widehat{\boldsymbol{\Sigma}}_{t,i} \mathbf{x}_{*,k} + \sigma^2).$$

Now we proceed the same way as in Lemma 2 of Lalitha et al. (2023). Using Cochran's theorem, we have that $\hat{\sigma}_{t,i,k}^2 m/\sigma_{i,k}^2$ is a χ^2 random variable with m degrees of freedom. Then using (4.4) and Lemma 1 of Laurent and Massart (2000) we can show that

$$\mathbb{P}\left(m - \frac{\hat{\sigma}_{t,i,k}^2 m}{\sigma_{i,k}^2} \geqslant 2\sqrt{m\log(1/\delta)}\right) \leqslant \delta \tag{G.3}$$

Dividing both sides of (G.3) in the probability by m, and multiplying by $\sigma^2_{i,k}$, we can get the following

$$\mathbb{P}\left(\sigma^2_{i,k}(1-2\sqrt{\log(1/\delta)/m}) \geqslant \hat{\sigma}^2_{t,i,k}\right) \leqslant \delta.$$

Observe that $1-2\sqrt{\log(1/\delta)/m} > 0$, we can divide both sides by it and get the first claim of the lemma. The second claim is proved by (4.3) in Laurent and Massart (2000), an immediate corollary of their Lemma 1, we have

$$\mathbb{P}\left(\frac{\hat{\sigma}^2_{t,i,k}m}{\sigma^2_{i,k}} - m \geqslant 2\sqrt{m\log(1/\delta)} + 2\log(1/\delta)\right) \leqslant \delta$$

The claim of the lemma follows. □

G.2 Additional Experiments and Results

We use NVIDIA GeForce RTX 3090 GPU with 24GB RAM to load the Large Language Models for inference. The Mistral-7B model requires less than 16GB RAM, and Vinuna-13B model requires less than 22 GB RAM during execution. To run Falcon-40B model we use AWS ml-g5.12xlarge machine. To run the full set of experiments it takes 24-27 hours of compute job. We now briefly discuss the various datasets used in this work.

Datasets

We now briefly describe the datasets used for our experiments. All the real-life datasets are from UCI (Markelle Kelly, 1988) and OpenML (Vanschoren et al., 2013) repositories. We use 4 classification and 3 regression datasets from UCI and OpenML. Additionally, we use 2 custom datasets for movie names and entity names for classification task in our experiments.

These are as follows:
(**1**) *Iris*: We use this UCI dataset for classification task. This dataset consists of four features of flowers and three classes of flowers. We use all four features in the prompts as well as estimating the score for selecting the next action. The dataset consists of 150 instances. We randomly choose $K = 20$ as test examples and the remaining instances as training examples.
(**2**) *Banknote-authentication*: We use this OpenML dataset for classification task. This dataset consists of five features of banknotes and two classes for identifying fake or original banknote. Out of these five features, we use four features in the prompts as well as estimating the score for selecting the next action. The dataset consists of 150 instances. We randomly choose $K = 20$ as test examples and the remaining instances as training examples.
(**3**) *Balance-scale*: We use this OpenML dataset for classification task. This dataset consists of five features of a scale and three classes of whether the scale tips left/right or is balanced. Out of these five features, we use four features in the prompts as well as estimating the score for selecting the next action. The dataset consists of 625 instances. We randomly choose $K = 20$ as test examples and the remaining instances as training examples.
(**4**) *Thyroid-new*: We use this OpenML dataset for classification task. This dataset consists of six features for thyroids and three classes. Out of these six features, we use five features in the prompts as well as estimating the score for selecting the next action. The dataset consists of 215 instances. We randomly choose $K = 20$ as test examples and the remaining instances as training examples.
(**5**) *Movie-name*: We use this custom dataset for classification task. This dataset consists of movie names across five genres (classes) romance, horror, thriller, sport, and action. We convert the movie names into 768 dimensional feature embeddings using Instructor embeddings. Note that in the prompt to the LLM we only pass the movie names and the goal is to identify the common genre. The dataset consists of 100 instances. We

randomly choose $K = 20$ as test examples and the remaining instances as training examples.

(**6**) *Movie-theme*: We use this custom dataset for classification tasks in identifying a common theme between pairs of movies. This dataset consists of movie names across five themes (classes) good-vs-evil, man-vs-nature, redemption, Love conquers all, and coming-of-age. We convert the movie names into 768 dimensional feature embeddings using Instructor embeddings. Note that in the prompt to the LLM we only pass the pair of movie names and the goal is to identify the common theme. The dataset consists of 100 instances. We randomly choose $K = 20$ as test examples and the remaining instances as training examples.

(**7**) *Entity-name*: We use this custom dataset for classification task. This dataset consists of entity names across five entity types (classes) like mountains, seas, rivers, vehicles, and celebrities. Again, we convert the entity names into 768 dimensional feature embeddings using Instructor embeddings. Note that in the prompt to the LLM we only pass the entity names and the goal is to identify the entity type. The dataset consists of 100 pairs of instances. We randomly choose $K = 20$ pairs as test examples and the remaining instances as training examples.

(**8**) *Fifa*: We use this OpenML dataset for the regression task. This dataset consists of six features of players and the clubs they joined as targets. Out of these six features, we use five features in the prompts as well as estimating the score for selecting the next action. The dataset consists of 18063 instances. We randomly choose $K = 20$ test examples and another 200 examples as training examples.

(**9**) *Machine-cpu*: We use this OpenML dataset for regression tasks. This dataset consists of seven features of machine cpu and the target variable is the performance of the cpu. Out of these seven features, we use five features in the prompts as well as estimating the score for selecting the next action. The dataset consists of 209 instances. We randomly choose

$K = 20$ test examples and the remaining examples as training examples.

ARC Experiment

In the recent works of Mirchandani et al. (2023); Srivastava et al. (2022) they showed that LLMs behave as general pattern-matching machine. In fact they showed that LLMs can be used to solve tasks from Abstract Reasoning Corpus (ARC) tasks. In the following experiments, we choose two such tasks: (1) ARC expansion and contraction experiment and (2) ARC rotation experiment.

(1) ARC expansion and contraction experiment: In the expansion and contraction experiment, there are two sets of matrices of dimension 4×4 which constitute half the examples of the feature space $\mathcal{X}$. The first set of input matrices have integer values in their center 2×2 cells while all the other cells are 0. The label space $\mathcal{Y}$ of this 4×4 matrix is also a 4×4 matrix where the 4 inner cells have moved to the 4 corners. These matrices are termed as expansion matrices.

Similarly, the other set of 4×4 matrices have the 4 non-zeros values in their corners. These constitute the remaining examples in $\mathcal{X}$. Then the label space $\mathcal{Y}$ is given by 4×4 matrix where the four non-zeros cells come to the center and all the other cell values are 0. These matrices are termed as contraction matrices. This is shown in Figure G.1a.

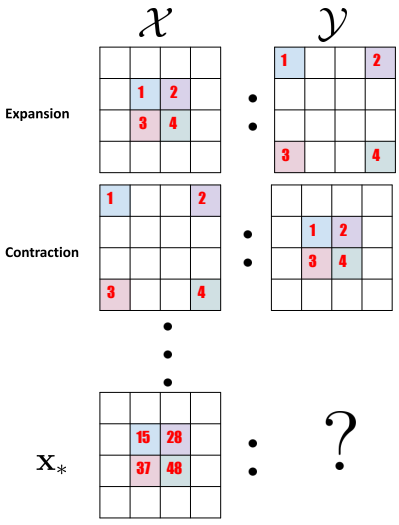
Therefore, the feature space $\mathcal{X}$ consists of both the expansion and contraction matrices. At every trial, and n training examples and K test examples are chosen randomly from $\mathcal{X}$. Then we run all baselines for T iterations where the classification accuracy is calculated if the LLM is able to predict the exact matching. This experiment is shown in Table 8.3.

(2) ARC rotation experiment: In the rotation experiment, there are again two sets of matrices of dimension 4×4 which constitute half the examples of the feature space $\mathcal{X}$. The first set of matrices are have integer values in their four corner cells while all the other cells are 0. The label

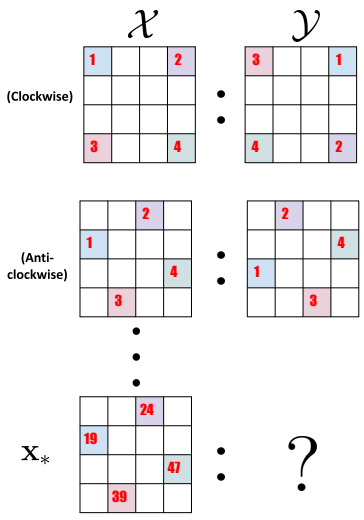
space $\mathcal{Y}$ of this 4×4 matrix is also a 4×4 matrix where the 4 corner cell values have moved $90°$ in the clockwise direction. These matrices are termed as clockwise matrices.

Similarly, the other set of 4×4 matrices have the 4 non-zeros values in their corners. These constitute the remaining examples in the feature space $\mathcal{X}$. Then the label space $\mathcal{Y}$ is given by 4×4 matrix where the four non-zeros cells have moved $15°$ in the anti-clockwise direction and all the other cell values are 0. These matrices are termed as anti-clockwise matrices. This is shown in Figure G.1b.

Therefore, the feature space $\mathcal{X}$ consists of both the clockwise and anti-clockwise matrices. At every trial, n training examples and K test examples are chosen randomly. Then we run all the baselines for T iterations where the classification accuracy is calculated if the LLM is able to predict the exact matching. This experiment is shown in Table 8.3.



(a) Classification on ARC Expansion and Contraction Experiment



(b) Classification on ARC Rotation Experiment

Figure G.1: Explanation of ARC tasks

PCFG Experiment

In this experiment the goal is to predict the next output of a sequence. In the following experiments we choose two such tasks: (1) PCFG add-subtract experiment and (2) PCFG repeat experiment.

(1) PCFG add-subtract experiment: In the add-subtract experiment, there are two sets of sequence of 4 integers. The first set of sequence of 4 integers consists of odd integer values which constitute half the examples in the feature space $\mathcal{X}$. The label space $\mathcal{Y}$ of this sequence of 4 odd examples is sequences of 5 integers where the last integer is padded to the original sequence by adding one to the last odd integer. These sequences are termed as add examples.

Similarly, the other set of examples consists of a sequence of 4 even integer values which constitute the remaining examples in the feature space $\mathcal{X}$. The label space $\mathcal{Y}$ of this sequence of 4 even integer examples is a sequence of 5 integers where the last integer is padded to the original sequence by subtracting one from the last even integer. These examples are termed as even examples. This is shown in Figure G.2a.

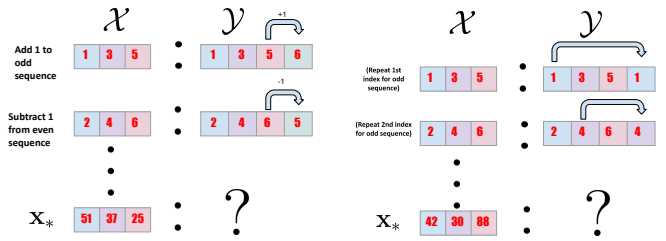
Therefore, the feature space $\mathcal{X}$ consists of both the odd and even sequence of 4 integer value examples. At every trial, n training and K test examples are chosen randomly. Then we run all the baselines for T iterations where the classification accuracy is calculated if the LLM is able to predict the exact matching. This experiment is shown in Table 8.3.

(2) PCFG repeat experiment: In the repeat experiment, there are two sets of sequence of 4 integers. The first set of sequence of 4 odd integer values constitute half the examples of the feature space $\mathcal{X}$. The label space $\mathcal{Y}$ of this sequence of 4 integers examples is a sequence of 5 integers where the last integer is padded to the original sequence by repeating the first odd integer. These sequences are termed as odd-repeat examples.

Similarly, the other set of examples consists of sequence of 4 even integer values which constitute the remaining examples in the feature

space $\mathcal{X}$. The label space $\mathcal{Y}$ of these sequence of 4 integer value examples is a sequence of 5 integers where the last integer is padded to the original sequence by repeating the second even integer. These examples are termed as even-repeat examples. This is shown in Figure G.2b.

Therefore, the feature space $\mathcal{X}$ consist of both the odd-repeat and even-repeat examples. At every trial, n training examples and K test examples are chosen randomly. Then we run all the baselines for T iterations where the classification accuracy is calculated if the LLM is able to predict the exact matching. This experiment is shown in Table 8.3.



(a) Classification on PCFG add-subtract Experiment

(b) Classification on PCFG repeat Experiment

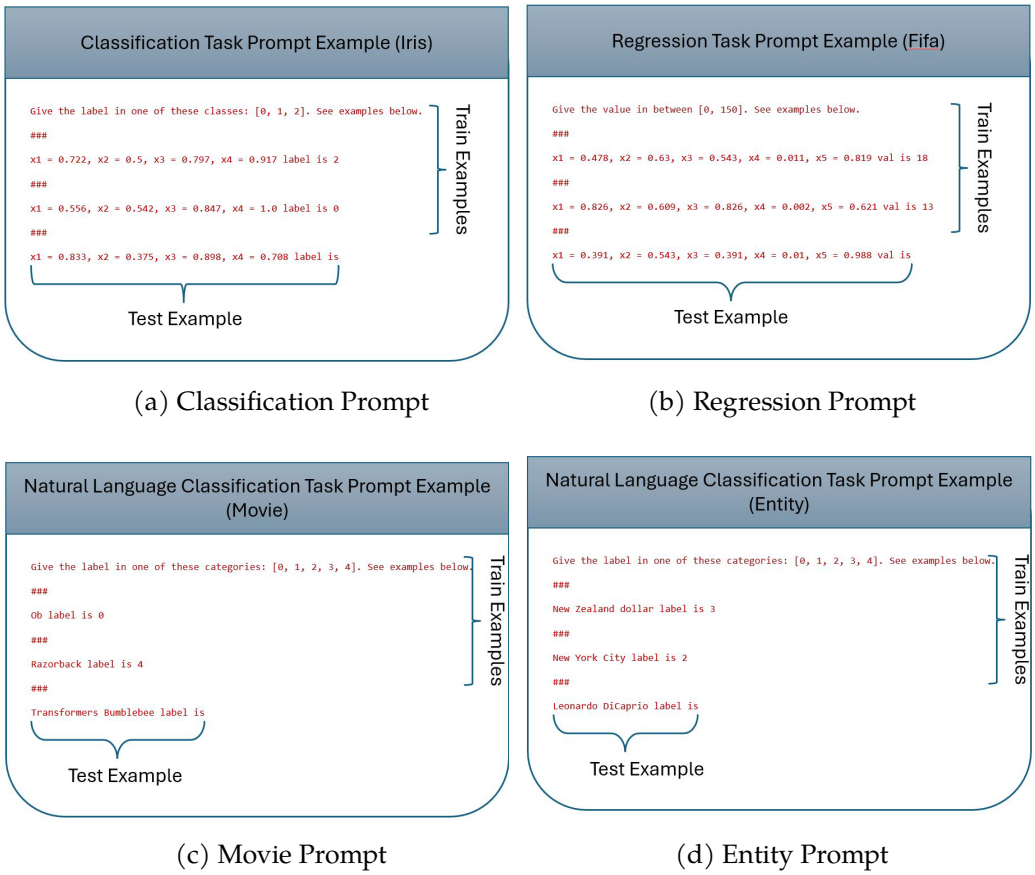
Figure G.2: Explanation of PCFG task.

G.3 Prompt Examples

Classification Dataset Prompts : Below we give an example of how we use the prompts to be used in the LLM for the Iris misclassification task. Similar types of prompts can be found in Dinh et al. (2022); Suzgun et al. (2022). This is shown in Figure G.3a. Note that since we have the feature representation of the training and test examples from the dataset, we directly use them as $\mathbf{x}_i$ and $\mathbf{x}_{*,k}$.

Regression Dataset Prompts : In Figure G.3b we give an example of a prompt for regression task in Fifa dataset. Note that since we have the

feature representation of the training and test examples from the dataset, we directly use them as $\mathbf{x}_i$ and $\mathbf{x}_{*,k}$.



(a) Classification Prompt

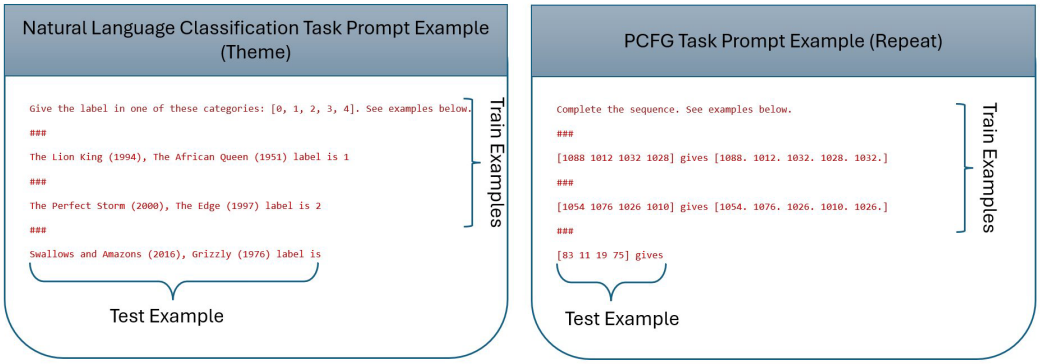
(b) Regression Prompt

(c) Movie Prompt

(d) Entity Prompt

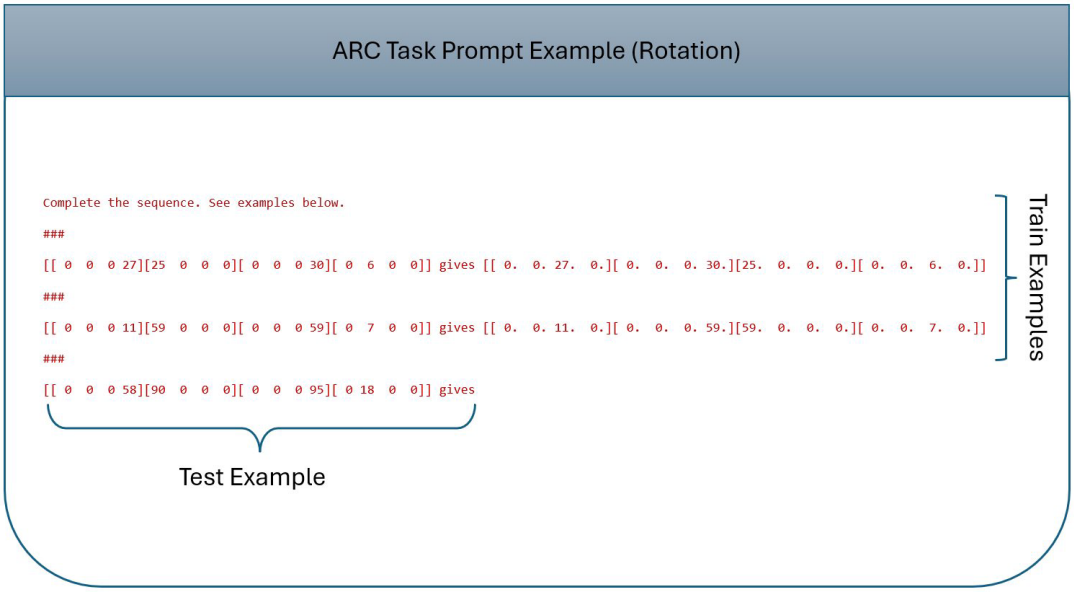
Figure G.3: Prompt examples for Classification, Regression, Movie, and Prompt

Movie Theme Experiment: We use a similar technique as in Iris dataset for this setting. The labels of the pairs of movies belong to 5 classes as follows: good-vs-evil, man-vs-nature, redemption, Love conquers all, and coming-of-age. At every iteration, we pass K pairs of movie test examples where each $\mathbf{x}_{*,k}$ is a pair of movies. In the example below we have $\mathbf{x}_{*,1} =$ ['Swallows and Amazon (2016), Grizzly (1976)']. Note that we feed the



(a) Theme Prompt (b) PCFG Prompt

Figure G.4: Prompt examples for Theme and PCFG tasks



(a) ARC Prompt

Figure G.5: Prompt examples for ARC task

natural language text to the LLM as prompts as shown in Figure G.4a. However, to run GO, SAL, and other baselines we require a featurization of

these natural language prompts. We obtain a 768 dimensional featurized representation of the pairs of movies 'Monsters Inc, Frozen (2013)' using Instructor embedding (Su et al., 2022). This constitutes $\mathbf{x}_i \in \mathbb{R}^{768}$ and $\mathbf{x}_{*,k} \in \mathbb{R}^{768}$.

Movie Name Experiment: We use a similar technique as in Iris dataset for this setting. The labels of the movie genres belong to 5 classes as follows: romance, horror, thriller, sport, and action. At every iteration we pass a set of test movie name examples where each $\mathbf{x}_{*,k}$ is now movie name. Note that we feed the natural language text to the LLM as prompts as shown in Figure G.3c. However, to run GO, SAL, and other baselines we require a featurization of these natural language prompts. We obtain a 768 dimensional featurized representation of the movie names using Instructor embedding (Su et al., 2022). This constitutes $\mathbf{x}_i \in \mathbb{R}^{768}$ and $\mathbf{x}_{*,k} \in \mathbb{R}^{768}$.

Entity Name Experiment: The labels of the entity genres belong to 5 classes as follows: mountains, seas, rivers, vehicles, and celebrities. At every iteration, we pass a set of test entity name examples where each $\mathbf{x}_{*,k}$ is now an entity name. Note that we feed the natural language text to the LLM as prompts as shown in Figure G.3d. However, to run GO, SAL, and other baselines we require a featurization of these natural language prompts. Again, we obtain a 768 dimensional featurized representation of the entity names using Instructor embedding (Su et al., 2022). This constitutes $\mathbf{x}_i \in \mathbb{R}^{768}$ and $\mathbf{x}_{*,k} \in \mathbb{R}^{768}$.

PCFG Experiment: We show an example of this prompt in Figure G.4b. Here we concatenate the sequence to obtain training examples $\mathbf{x}_i$ and test examples $\mathbf{x}_{*,k}$. So a sequence of 4 integers of length 4 will be represented by $\mathbf{x}_i, \mathbf{x}_{*,k} \in \mathbb{R}^{16}$. Similarly the label Y_i and $Y_{*,k}$ consist of sequence of 5 integers of length 4 which we concatentate to get a vector of length $\mathbb{R}^{20}$.

ARC Experiment: We show an example of this prompt in Figure G.5a. Here we vectorized the 4×4 matrix to obtain training examples $\mathbf{x}_i \in \mathbb{R}^{16}$

and test examples $\mathbf{x}_{*,k} \in \mathbb{R}^{16}$. Similarly the label Y_i and $Y_{*,k}$ consist of vectorized matrices of length $\mathbb{R}^{16}$.

G.4 Table of Notations

Notations	Definition
n	Total unlabeled examples
d	Dimension of the feature
$\mathcal{X}$	Feature set
$\mathcal{Y}$	Label space
θ_*	Unknown model parameter
$\mathbf{x}_i$	Feature of sample example i
$\mathbf{x}_{*,k}$	k-th test example
$f(\mathbf{x}, \theta_*)$	Model
Y_*	Label
$H_t = (X_\ell, Y_\ell)_{\ell \in [t-1]}$	History of $t-1$ previously labeled examples
$p(\cdot \mid \mathbf{x}, H_t)$	Distribution of the label of example $\mathbf{x}$ conditioned on H_t
θ_0	Prior mean of the unknown model parameter θ_*
$\mathbf{\Sigma}_0$	Prior mean of the unknown model parameter θ_*
$\widehat{\mathbf{\Sigma}}_t = \left(\mathbf{\Sigma}_0^{-1} + \sigma^{-2} \sum_{\ell=1}^{t-1} X_\ell X_\ell \mathbf{T}\right)^{-1}$	Posterior covariance
$\mathcal{L}_t$	Set of labeled examples
$\mathcal{U}_t$	Set of unlabeled examples
$\widehat{\theta}_{t,i,j}$	Posterior mean

Table G.1: Table of Notations for GO