Title

Uncertainty-Aware Unsupervised and Robust Reinforcement Learning

Author

Zhang, Weitong

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Uncertainty-Aware Unsupervised and Robust Reinforcement Learning

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Computer Science

by

Weitong Zhang

2024

ABSTRACT OF THE DISSERTATION

Uncertainty-Aware Unsupervised and Robust Reinforcement Learning

by

Weitong Zhang
Doctor of Philosophy in Computer Science
University of California, Los Angeles, 2024
Professor Quanquan Gu, Chair

This dissertation is centered around addressing several key concerns in reinforcement learning (RL). RL has been a popular topic in the design of autonomous intelligent agents that make decisions and learn optimal actions through interaction with the environment. Over the past decades, RL has achieved significant success in various domains. However, RL has consistently been criticized for its inefficiency in exploration and vulnerability to model errors or noise. This dissertation aims to tackle these challenges through uncertainty-aware methods.

In the first part of this dissertation, we explore how an RL agent can efficiently explore the environment without human supervision. We begin with a theoretical framework on reward-free exploration and establish a connection between reward-free exploration and unsupervised reinforcement learning. We provide both theoretical analyses and practical algorithms that exhibit competitive empirical performance. In the second part of this dissertation, we aim to develop robust RL algorithm in a misspecified setting, where the function class (e.g., Neural Networks) cannot adequately approximate the underlying ground truth function. We show how significant the approximation error needs to be in order to prevent the agent from

efficiently learning the environment and making good decisions. We also present several algorithms that ensure the agent will only make a finite number of mistakes over infinite runs when this approximation error is small.

The methods and techniques discussed in this dissertation advance the theoretical understanding of key concerns and limitations in RL, particularly in scenarios that require performance guarantees. Additionally, these findings not only suggest further research directions but also pose several open questions that would help better design more robust and efficient decision making processes in the future.

The dissertation of Weitong Zhang is approved.

Richard E. Korf

Lihong Li

Baharan Mirzasoleiman

Stanley J. Osher

Quanquan Gu, Committee Chair

University of California, Los Angeles

2024

To my beloved ones.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my advisor, Quanquan Gu, for his consistent support, encouragement, and guidance throughout my Ph.D. career. Prior to my Ph.D., my research experience was limited and my mathematical background was modest. Quanquan generously provided help and constructive suggestions that lead me into the research community. I vividly remember that during the initial months of my Ph.D., he devoted considerable time to teaching me statistical tools and methodologies to write rigorous proofs in my papers. This not only solidified my understanding of machine learning but also showed me how to mentor and collaborate with students. Furthermore, I treasure the opportunity to work with the talented individuals in Quanquan's lab. He has also encouraged me to build more collaborative relationships, particularly with interdisciplinary researchers, to broaden the application of my research across various topics. Through these collaborations, I have learned how to engage with researchers from different fields and mentor junior Ph.D. students. These will be invaluable in my future academic career. I also appreciate his pioneering vision in research, which not only focuses on insightful theoretical analysis but also on developing practical algorithms. This vision has greatly influenced my own research interests and objectives for the future. Finally, I appreciate his consistent belief in my potential and his encouragement to develop as an independent researcher. I would like to extend my deepest gratitude for his advice and wholehearted support regarding my career plans and job search. This guidance has deepened my understanding of academic communities and the path to becoming a respected researcher and successful professor in the future.

I would like to express my sincere appreciation to the members of my doctoral committee: Richard E. Korf, Lihong Li, Baharan Mirzasoleiman, and Stanley J. Osher for their invaluable feedback and suggestions on my research ideas and thesis writing. Special thanks to Prof. Korf for his suggestion to present my research to a broad computer science audience,

which has been beneficial not only in thesis writing but also in preparing for job talks during my academic job search.

I extend heartfelt gratitude to Lihong Li, Chong Liu, Hongning Wang, Wei Wang and Amy Zhang for their support and guidance in various collaborative projects. I am particularly fortunate to have published my first paper with Lihong, a pioneering researcher in this field. I also treasure the opportunity to work with Chong on interdisciplinary tasks, which have broadened my vision and shown me the vast possibilities of applying machine learning across multiple areas. I am immensely thankful to Joe Eaton, Bradley Rees and Xiaoyun Wang for their mentorship and support during my internship at NVIDIA. The industry experience has greatly enhanced my understanding of how my research applies to real-world applications.

Throughout my PhD journey, I have had the fortune of collaborating with many talented individuals in Quanquan's research team. My deepest gratitude goes to Yuan Cao, Jinghui Chen, Zixiang Chen, Qiwei Di, Jiafan He, Kaixuan Ji, Xuheng Li, Lingxiao Wang, Yue Wu, Pan Xu, Heyang Zhao, Dongruo Zhou and Difan Zou. I also appreciate the extraordinary efforts of Jinghui Chen, Yuanzhou Chen, Yihe Deng, Zhiyuan Fan, Jiafan He, Benjamin Hoar, Zijie Huang, Jeehyun Hwang, Kaixuan Ji, Yiling Jia, Hongyuan Sheng, Jingwen Sun, Lingxiao Wang, Yue Wu, Pan Xu, Junkai Zhang, Linxi Zhao, Qingyue Zhao, Dongruo Zhou and Difan Zou in our collaborative projects. I'd also like to thank Guorui Chen and Huaxiu Yao for their advice in my job-hunting. I'd like to extend special thanks to Dongruo and Difan for their support throughout my PhD. I treasure the memories with Dongruo, hiking through the mountains, beaches and streets of LA.

Furthermore, I would like to extend my appreciation to a group of friends not previously mentioned. I am grateful to Peilin Chai, Fan Jin, Zifeng Kang, Yiwen Lu, Wenhao Qi, Pengwei Wang, Yang Wang, Xinyu Yao, Qiuyang Yin, Tao Zhang and Haiyuan Zou for their consistent support throughout my five-year PhD journey. I treasure the times we spent hiking, playing badminton and tennis, and engaging in late-night conversations about history, the future, and life. I cherish the memory of visiting UCSD in 2020, the first time

I ventured out from home during COVID. You all warmly welcomed me despite the risks associated with the pandemic. A special thank you to Fan for accompanying me on a journey across the US continent; I cannot imagine having done it without you.

Finally, I'd like to extend my deepest gratitude to my parents, Kai and Longhua and my grandparents, Shifang and Yixin, for their understanding and unconditional support throughout my life. I am also profoundly thankful to Yijun for filling my life with love and happiness. Your encouragement and love are the greatest support on this wonderful journey.

VITA

2019	Bachelor of Engineering in Automation, Tsinghua University
2019–2021	Research Assistant, Computer Science Department, University of California, Los Angeles
2021–2022	Teaching Assistant, Computer Science Department, University of California, Los Angeles
2022	Master of Science in Computer Science, University of California, Los Angeles
2023–2024	Research Assistant, Computer Science Department, University of California, Los Angeles

PUBLICATIONS

We select the publications that are most related to this dissertation

* indicates equal contribution.

1. **Weitong Zhang**, Dongruo Zhou, and Quanquan Gu. 2021. Reward-free model-based reinforcement learning with linear function approximation. Advances in Neural Information Processing Systems 34, (2021), 1582–1593. **Presented in Chapter 2.**

2. Junkai Zhang, **Weitong Zhang**, and Quanquan Gu. 2023. Optimal horizon-free reward-free exploration for linear mixture mdps. In International Conference on Machine Learning, PMLR, 41902–41930. **Presented in Chapter 2.**

3. Junkai Zhang*, **Weitong Zhang***, and Quanquan Gu. 2024. Uncertainty-Aware Reward-Free Exploration with General Function Approximation. *To appear in* International Conference on Machine Learning, PMLR,. **Presented in Chapter 3.**

4. **Weitong Zhang**, Jiafan He, Zhiyuan Fan, and Quanquan Gu. 2023. On the interplay between misspecification and sub-optimality gap in linear contextual bandits. In International Conference on Machine Learning, PMLR, 41111–41132. **Presented in Chapter 4.**

5. **Weitong Zhang***, Zhiyuan Fan*, Jiafan He, and Quanquan Gu. 2024. Settling Constant Regrets in Linear Markov Decision Processes. arXiv preprint arXiv:2404.10745 (2024). **Presented in Chapter 5.**

CHAPTER 1

Introduction

Recent years have witnessed great success of reinforcement learning (RL) in excelling at a wide spectrum of games, such as Atari (Mnih et al., 2013), Go (Silver et al., 2016) and even more complex games (Berner et al., 2019; Vinyals et al., 2019). In order to achieve these objectives, reinforcement learning agents usually need to explore and interact with the environment. By receiving the rewards which encode information about the goal of the task, the RL agents can learn through trial and error. Taking the Breakout game as an example, as presented in Figure 1.1[1], the RL agent observes visual input from the screen, which encodes information about the positions of the ball and paddle, as well as the brick structure. It needs to control a paddle to hit the moving ball, receiving positive rewards when the ball hits the bricks and negative rewards when the ball falls off the screen. Through exploring by randomly moving the paddle, the RL agent will learn that moving the paddle to the right in Figure 1.1's situation will lead to a positive reward, whereas moving the paddle to the left will cause the agent to lose the game, thus yield a negative reward. Therefore, RL agents can leverage this information and learn to conquer the Breakout game or eventually beat human experts (Mnih et al., 2013). Besides being

Figure 1.1: A screenshot of the Atari Breakout game.

[1]Image credit: https://en.wikipedia.org/w/index.php?title=Breakout_(video_game)&oldid=1224017580

applied in games, RL has also emerged as a new paradigm for automatically solving more practical tasks such as recommendation systems (Li et al., 2011), robotic systems (Kober et al., 2013), and autonomous driving (Sallab et al., 2017), which all rely on interacting with the environment and dynamically making decisions based on the observations from the environment, such as the user feedback or the system response.

Despite these advances, since massive interaction with the environment is a must in reinforcement learning, there are a series of crucial concerns that prevent RL from being applied to more serious tasks, such as drug design, scientific discovery, and clinical treatment design. These concerns usually consist of the efficiency and robustness of exploration for RL agents, especially in the face of uncertainty. This dissertation focuses on studying these concerns through theoretical analysis. Inspired by these insights, we also develop a series of algorithms that not only offer theoretical guarantees but also demonstrate competitive empirical performance.

The first concern we would like to address is the case when the reinforcement learning agent is facing the *"extrinsic"* uncertainty when exploring the *"unknown"* environment. In particular, we would like to address the efficiency of exploration in reinforcement learning, especially exploration without human supervision or human-crafted rewards. Usually, reward functions are human-crafted to encode the expected behavior of the agent. For example, in the Atari game Montezuma's Revenge, as presented in Figure 1.2[2], the agent is designed to receive a reward of 1 when obtaining the key and 0 otherwise. However, it has been demonstrated that RL agents cannot perform well in environments with these sparse rewards (Kang et al., 2022) because the reward in most of the collected data

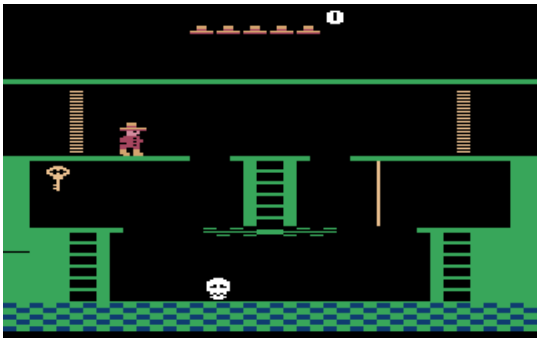


Figure 1.2: A screenshot of the Atari Montezuma's Revenge.

[2]Image credit: `https://www.retrogames.cz/play_124-Atari2600.php`

is zero. Therefore, more complicated and nontrivial rewards (Dilokthanakul et al., 2019) are crafted to guide the behavior of the agent. However, these reward-design processes are inefficient because they require a lot of trial and error to train the agent and adjust the reward. This difficulty also arises when applying RL to environments with limited human knowledge. For instance, when applying RL to drug discovery tasks (Popova et al., 2018), current human knowledge may not adequately describe the detailed mechanisms of some new proteins, making reward design challenging and requiring additional efforts. Moreover, this inefficiency also appears in multi-task robotics (Kalashnikov et al., 2022), where RL agents are expected to excel in multiple objectives instead of a single task. In such cases, using the reward for a single task would lead to repeated exploration of the environment and be inefficient. Therefore, since exploration using a single human-crafted reward raises efficiency issues, the question *"How to explore the environment without human supervision or human-crafted rewards"* becomes a natural concern for these tasks.

To answer the previous question, Jin et al. (2020a) provided a theoretical framework called *reward-free exploration* (RFE) to force the agent to explore without reward signals. Over the past few years, there has been a body of work (Ménard et al., 2020; Wang et al., 2020b; Zhang et al., 2020) theoretically improving the efficiency of RFE in the regime of "tabular RL," where the state and action spaces are finite. On the other hand, Laskin et al. (2020) proposed an empirical framework called *unsupervised reinforcement learning* (URL) to pre-train the RL agent to explore the environment without the reward signals for any specific tasks. Then these pretrained models are expected to behave well in a spectrum of downstream tasks with different reward functions by simply fine-tuning on these tasks. In parallel with the development of theoretical analysis of RFE, there has also been a series of works (Pathak et al., 2017, 2019; Burda et al., 2018b) on empirically designing exploration heuristics for URL.

Both RFE and URL aim to improve the efficiency of multi-task decision-making systems, such as multi-task robotics. In particular, both methods explore the environment by

either collecting data (RFE) or learning good representations of the environment through a pretraining process (URL). Then, with different reward functions representing various downstream tasks, both methods can efficiently output the optimal policy without extensive interaction with the environment. Additionally, these methods both encourage exploration in environments that inherently lack rich reward information, like the aforementioned Montezuma's Revenge. In **Chapter** 2, we make the first step in connecting RFE and URL by studying RFE in a more general case where the state and action spaces are too large to apply the "tabular RL" method. In this case, we study RL with function approximations so that the action space and the state space can be represented compactly (Sutton et al., 1998). We start from linear mixture MDP (Ayoub et al., 2020) which assumes that the transition kernel can be approximated by a linear function. Through uncertainty measurement under linear function approximation, we are able to deliver an RFE strategy that is guaranteed to efficiently explore the environment. In the latter part of **Chapter** 2, we seek to further improve the analysis and the design of the algorithm to make it optimal in various settings, such as the sparse reward setting. In **Chapter** 3, we further push the analysis to general function approximation. We design a practical RFE algorithm that not only enjoys the theoretical RFE guarantee but also has competitive performance on a set of URL benchmarks. The result builds a connection between RFE from a theoretical perspective and URL from an empirical perspective.

When function approximation is used to compress the state and action space, yet another crucial concern is whether the function approximation is expressive enough for making good decisions. Therefore, the second concern we would like to address is the case when the reinforcement learning is facing the *"intrinsic"* uncertainty from the expressiveness of the function approximation. For example, when using linear functions and linear regression to approximate the data generated by some quadratic function, one will suffer from *model misspecification*. Intuitively, a larger model misspecification will potentially have a more negative impact on decision-making systems. Existing theoretical RL literature (Jin et al.,

2020b; Zanette et al., 2020a) usually assumes that the function can *perfectly* approximate the ground truth function. When model misspecification exists, their analysis will leave an "approximation error" term indicating that the model will always make mistakes, regardless of the number of interactions (Takemura et al., 2021; Vial et al., 2022). In **Chapter** 4, we improve this analysis by connecting the "required precision" with the "model misspecification" in misspecified contextual bandits, where the agent is only required to make a one-step decision. The "required precision" can generally be viewed as the difference between the best action and the second best action (a.k.a., suboptimality gap (Lattimore and Szepesvári, 2020)). Obviously, when the "required precision" is larger than the "model misspecification", it would be easy to distinguish the optimal action from the rest of the actions; thus, the agent will easily make the correct decision. Based on that observation, we propose an algorithm that actively learns from the data with higher uncertainty while filtering out the data about which the agent is certain. Intuitively, through this process we can guarantee that the agent will not be significantly affected by the model misspecification. We also reveal the interplay between the "misspecification level" and the "suboptimality gap", indicating *how large misspecification will prevent us from making good decisions*. This positive result matches the negative result proposed in Lattimore et al. (2020). In **Chapter** 5, we extend this result to a general RL setting, i.e., sequential decision processes. We show that through an active data selection regime, the agent suffers only a *finite* suboptimality (a.k.a., *constant regret*) when making decisions over an infinite run, even when the model misspecification exists. This constant regret result requires no prior assumption as made in Papini et al. (2021a); Zhang et al. (2021a), and the interaction between the "required precision" and "model misspecification" provide an insightful vision on the development of empirical robustness algorithms.

1.1 Organization of the Dissertation

The rest of this dissertation is organized as follows. In **Chapter** 2, we discuss reward-free exploration under linear function approximation, which improves the previous results in the tabular setting with finite state and action spaces. An improved algorithm leveraging variance information and an algorithm working in the *bound total reward* setting are presented in the latter part of **Chapter** 2. In **Chapter** 3, we extend RFE with linear function approximation to RFE with general function approximation. In addition to theoretical analysis, numerical experiments demonstrate that our reward-free exploration algorithm has competitive performance on a set of unsupervised reinforcement learning benchmarks. In **Chapter** 4, we move on to the second topic regarding the robustness of misspecified linear bandits (Li et al., 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011). By proposing an active data selection algorithm and revisiting an improved version of Chu et al. (2011), we show the interplay between model misspecification and the suboptimality gap. We are also able to deliver a high-probability constant regret for misspecified linear bandits without prior assumptions on contextual vectors. In **Chapter** 5, we extend this result to linear MDP (Jin et al., 2020b). By introducing the "certified estimator", we are able to provide robust estimation for sequential decision processes in linear MDP and deliver a similar constant regret bound. The conclusions are drawn in **Chapter** 6, which also includes the future directions and open questions in the unsupervised, reward-free reinforcement learning and the misspecified, robust reinforcement learning algorithms.

1.2 Notation System in this Dissertation

In this dissertation, scalars are denoted by lowercase letters. Vectors are denoted by lowercase boldface letters $\mathbf{x}$, and matrices by uppercase boldface letters $\mathbf{A}$. We denote by $[k]$ the set $\{1, 2, \cdots, k\}$ for positive integers k. We use $\log x$ to denote the logarithm of x to the base 2. For two nonnegative sequences $\{a_n\}, \{b_n\}$, $a_n = \mathcal{O}(b_n)$ means that there exists a positive

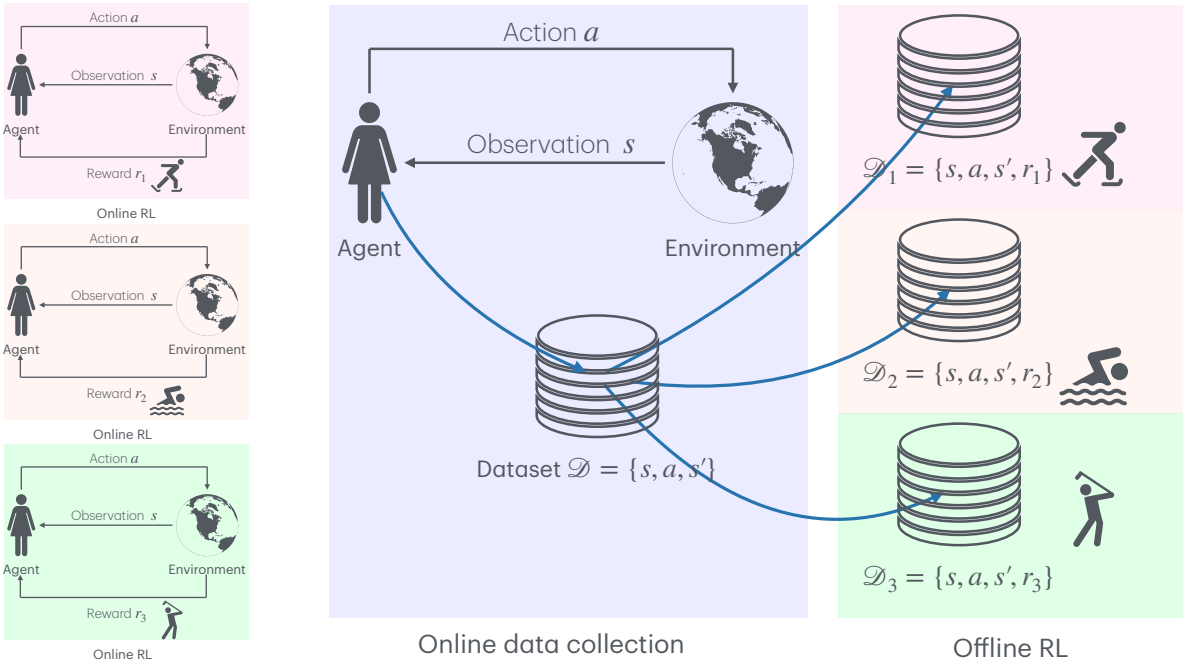
constant C such that $a_n \leqslant Cb_n$. Notation $a_n = \widetilde{\mathcal{O}}(b_n)$ means that there exists a positive constant k such that $a_n = \mathcal{O}(b_n \log^k b_n)$. Notation $a_n = \Omega(b_n)$ means that there exists a positive constant C such that $a_n \geqslant Cb_n$. Notation $a_n = \widetilde{\Omega}(b_n)$ means there exists a positive constant k such that $a_n = \Omega(b_n \log^{-k} b_n)$. Notation $a_n = \omega(b_n)$ means that $\lim_{n\to\infty} b_n/a_n = 0$. For a vector $\mathbf{x} \in \mathbb{R}^d$ and a positive semidefinite matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, we define $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^\top \mathbf{A}\mathbf{x}$. For any set $\mathcal{C}$, we use $|\mathcal{C}|$ to denote its cardinality. We denote the identity matrix by $\mathbf{I}$ and the empty set by $\varnothing$. The total variation distance of two distribution measures $\mathbb{P}(\cdot)$ and $\mathbb{Q}(\cdot)$ is denoted by $\|\mathbb{P}(\cdot) - \mathbb{Q}(\cdot)\|_{\mathrm{TV}}$. Remaining notations are defined before they are used in each chapter.

CHAPTER 2

Uncertainty-Aware Reward-Free Exploration with Linear Function Approximation

2.1 Introduction

In this chapter, we study a theoretical framework for unsupervised exploration in reinforcement learning, which is called *reward-free exploration*. In reinforcement learning (RL), an agent sequentially interacts with an environment and receives rewards from it. In many real-world RL problems, the reward function is designed manually to encourage the desired behavior of the agent. Thus, engineers have to change the reward function time by time and train the agent to check whether it has achieved the desired behavior. In this case, RL algorithms need to be repeatedly executed with different reward functions and are sample inefficient or even intractable. To tackle this challenge, Jin et al. (2020a) proposed a new reinforcement learning paradigm called *Reward-Free Exploration* (RFE), which explores the environment without using any reward function. In detail, the reward-free RL algorithm consists of two phases. The first phase is called the *Exploration Phase*, where the algorithm explores the environment without receiving reward signals. The second phase is called the *Planning Phase*, where the algorithm is given a specific reward function and uses the data collected in the first phase to learn the policy. In Figure 2.1 we provide a comparison between the classical "reward-aware" exploration and the proposed "reward-free" exploration when the RL agent is asked to learn three tasks (e.g., ice skating, swimming and playing golf). Reward-aware exploration (Figure 2.1a) explores the environment with a single, spe-



(a) Reward-aware exploration (b) Reward-free exploration

Figure 2.1: Comparison between (a) the "reward-aware" exploration and (b) the "reward-free" exploration.

cific reward so when the reward changes, the agent needs to repeat the exploration to adapt the new reward function. Reward-free exploration, as presented in Figure 2.1b, aim to learn a dataset without the reward function, which can potentially transferred to any single reward without explore the environment again.

As a first step for reward-free exploration, Jin et al. (2020a) has shown that this exploration paradigm can learn a near-optimal policy in the planning phase given *any* reward function after collecting a polynomial number of episodes in the exploration phase. The subsequent work (Kaufmann et al., 2021a; Ménard et al., 2020; Zhang et al., 2020) proposed improved algorithms to achieve a better or nearly optimal sample complexity.

All of the aforementioned works are focused on the tabular Markov decision process (MDP), where the number of states and the number of actions are finite. In practice, the

number of states and actions can be large or even infinite, for example, in a Go game (Silver et al., 2016), the number of states is typically as large as 10^{360}, making it impossible to apply tabular methods that store data as a table of states. In the Atari games (Mnih et al., 2013), the input is usually a 210×160 image representing the visual input from the video game. In both of these cases, *function approximations* (usually neural networks) are required for the sake of computational tractability and generalization. However, the understanding of function approximation for reward-free exploration, even under the simplest linear function approximation, remains underexplored. To mention a few, Wang et al. (2020b) studied *linear MDPs* (Yang and Wang, 2019; Jin et al., 2020b), where both the transition probability and the reward function admit linear representations, and proposed a reward-free RL algorithm with a $\widetilde{\mathcal{O}}(d^3H^6\epsilon^{-2})$ sample complexity, where d is the dimension of the linear representation, H is the planning horizon, and ϵ is the required accuracy. They also proved that if the optimal state-action function is linear, then the reward-free exploration needs an exponential number of episodes in the planning horizon H to learn an ϵ-optimal policy. Zanette et al. (2020d) considered a slightly larger class of MDPs with *low inherent Bellman error* (Zanette et al., 2020b), and proposed an algorithm with $\widetilde{\mathcal{O}}(d^3H^5\epsilon^{-2})$ sample complexity. However, both works assume that the reward function is a linear function over some feature mapping. Moreover, the lower bound proved in (Wang et al., 2020b) is for a very large class of MDPs where the optimal state-action function is linear, thus it is too conservative and cannot determine the information-theoretic limits of reward-free exploration for linear MDPs or related models.

2.1.1 Organization of this Chapter

In this chapter, we seek a theoretical understanding of the statistical efficiency for reward-free RL with linear function approximation. This chapter is organized as follows. In Section 2.2, we review the related literature. In Section 2.3, we present the basic assumption of RL with linear function approximation and the rigorous definition of reward-free exploration.

The preliminary algorithm and analysis are presented in Section 2.4. In Section 2.5, we seek to improve the sample complexity by incorporating the variance information into our algorithm. In Section 2.6, we further extend the algorithm to a long-horizon setting and present a nearly-minimax-optimal algorithm which does not suffer from *curse of horizon* in RL. The conclusion is drawn in Section 2.7 and we defer the detailed proof of the theorems to Section 2.8.

2.2 Related Works

2.2.1 Reinforcement Learning with Linear Function Approximation

In recent years, a series of works have been devoted to the study of RL with linear function approximation (Jiang et al., 2017; Dann et al., 2018; Yang and Wang, 2019; Wang et al., 2019; Du et al., 2019; Sun et al., 2019; Jin et al., 2020b; Zanette et al., 2020a,b; Yang and Wang, 2020a; Modi et al., 2020; Ayoub et al., 2020; Jia et al., 2020; Cai et al., 2020; Weisz et al., 2021; Zhou et al., 2021c,a; He et al., 2022a; Agarwal et al., 2022). Our work belongs to the linear mixture MDP setting (Yang and Wang, 2019; Modi et al., 2020; Ayoub et al., 2020; Jia et al., 2020; Zhou et al., 2021a,c), where the transition kernel can be parameterized as a linear combination of some basic transition probability functions. Zhou et al. (2021a) firstly achieved minimax regret $\widetilde{O}\left(dH\sqrt{T}\right)$ in linear mixture MDPs by proposing a Bernstein-type concentration inequality for self-normalized martingales. Another kind of popular linearly parameterized MDP is linear MDP (Wang et al., 2019; Du et al., 2019; Yang and Wang, 2020a; Jin et al., 2020b; Zanette et al., 2020a; Wang et al., 2020c; He et al., 2021a), which assumes that both transition probability and reward function are linear functions of known feature mappings in state-action pairs. In this setting, Jin et al. (2020b) first proposed the statistically and computationally efficient algorithm LSVI-UCB and achieved a $\widetilde{O}\left(\sqrt{d^3H^3T}\right)$ regret bound. Recent works (He et al., 2022a) further achieved nearly minimax optimal regret $\widetilde{O}(d\sqrt{H^3K})$ by proposing the computationally efficient algo-

rithm LSVI-UCB++. Its concurrent work (Agarwal et al., 2022) achieves a similar result under assumption $\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant 1$ with regret upper bound of $\widetilde{O}(d\sqrt{HT} + d^6H^5)$.

2.2.2 Reward-free Exploration

Exploration efficiency has always been a popular topic in RL. sophisticated exploration strategies like E^3 (Kearns and Singh, 2002) have been proposed to guide the exploration and these algorithms are proved to require only polynominal time to explore the environment. Unlike standard RL settings in which the agent interacts with the environment with reward signals, *reward-free exploration* (Jin et al., 2020a) in RL introduced a two-phase paradigm. In this approach, the agent initially explores the environment without any reward signals. Then, upon receiving the reward functions, it outputs a policy that maximizes the cumulative reward, without any further interaction with the environment. Jin et al. (2020a) first achieved $\widetilde{O}(H^5S^2A/\epsilon^2)$ sample complexity in tabular MDPs by executing exploratory policy visiting states with probability proportional to its maximum visitation probability under any possible policy. Subsequent works (Kaufmann et al., 2021b; Ménard et al., 2021) proposed algorithms RF-UCRL and RF-Express to gradually improve the result to $\widetilde{O}\left(H^3S^2A\epsilon^{-2}\right)$. The optimal sample complexity bound $\widetilde{O}(H^2S^2A\epsilon^{-2})$ was achieved by the algorithm SSTP proposed in Zhang et al. (2020), which matched the lower bound provided in Jin et al. (2020a) up to logarithmic factors. Recent years have witnessed a trend of reward-free exploration in RL with function approximations, while most of these works are considering linear function approximation: In the linear MDP setting, Wang et al. (2020b) proposes an exploration-driven reward function, and the minimax optimal bound was achieved by Hu et al. (2022) by introducing weighted regression into the algorithm. In linear mixture MDPs, Zhang et al. (2021e) proposed the 'pseudo reward' to encourage exploration, Chen et al. (2021); Wagenmaker et al. (2022) improved the sample complexity by introducing a more complicated, recursively defined pseudo reward. The minimax optimal sample complexity, $\widetilde{\mathcal{O}}(d^2/\epsilon^2)$ was

[1] **Time** means if the algorithm is time-homogeneous (✓) or not (×).

Table 2.1: Comparison of episodic reward-free algorithms.

Setting	Algorithm	Rewards Scale	Time[1]	Sample Complexity
Tabular MDP	Jin et al. (2020a)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}(H^5 S^2 A \varepsilon^{-2})$
	Kaufmann et al. (2021a)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}(H^4 S^2 A \varepsilon^{-2})$
	Ménard et al. (2021)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}(H^3 S^2 A \varepsilon^{-2})$
	Zhang et al. (2020)	$\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant 1$	✓	$\widetilde{O}(S^2 A \varepsilon^{-2})$
	Lower bound (Jin et al., 2020a)	$r_h(s_h, a_h) \in [0, 1]$	×	$\Omega(H^2 S^2 A \varepsilon^{-2})$
	Lower bound (Zhang et al., 2020)	$\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant 1$	✓	$\Omega(S^2 A \varepsilon^{-2})$
Linear MDP	Wang et al. (2020b)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}\left(H^6 d^3 \varepsilon^{-2}\right)$
	Zanette et al. (2020d)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}\left(H^5 d^3 \varepsilon^{-2}\right)$
	Wagenmaker et al. (2022)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}\left(H^5 d^2 \varepsilon^{-2}\right)$
Linear Mixture MDP	Theorem 2.5.1	$r_h(s_h, a_h) \in [0, 1]$	✓	$\widetilde{O}\left(H^4 d(H+d) \varepsilon^{-2}\right)$
	Chen et al. (2021)	$r_h(s_h, a_h) \in [0, 1]$	×	$\widetilde{O}\left(H^3 d(H+d) \varepsilon^{-2}\right)$
	Corollary 2.6.4	$\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant 1$	✓	$\widetilde{O}(d^2 \varepsilon^{-2})$
	Corollary 2.6.6	$\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant H$	✓	$\widetilde{O}(H^2 d^2 \varepsilon^{-2})$
	Lower bound (Thm. 2.6.8)	$\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant 1$	✓	$\Omega\left(d^2 \varepsilon^{-2}\right)$
	Lower bound (Cor. 2.6.10)	$r_h(s_h, a_h) \in [0, 1]$	✓	$\Omega(H^2 d^2 \varepsilon^{-2})$

achieved by Zhang et al. (2023a) in the horizon-free setting. Moving forward, in the general function approximation setting, Kong et al. (2021) used 'online sensitivity score' to estimate the information gain. As a result, they were able to provide a sample complexity of $\widetilde{\mathcal{O}}(d^4 H^6 \epsilon^{-2})$. Here, d represents the dimension of contexts when the problem is reduced to linear function approximations. Yet another line of works (Chen et al., 2022a,b) aimed to follow the Decision-Estimation Coefficient (DEC, Foster et al. 2021) and provided a unified framework for reward-free exploration with general function approximations, achieving a $\widetilde{\mathcal{O}}(\text{poly}(H) d^2 \epsilon^{-2})$, nevertheless, all existing works with general function approximations

leave a huge gap between their proposed upper bound and lower bound, even when reduced to linear settings. We record existing results in Table 2.1.

2.2.3 The Curse of Horizon in Reinforcement Learning

The long planning horizon has long been viewed as RL's main challenge. However, a series of works have shown that RL is no more difficult than contextual bandits by removing the influence of the total reward scale. In tabular MDPs, the algorithm proposed in Wang et al. (2020a) first achieved the polylogarithmic H dependency sample complexity bound $\widetilde{O}(S^5A^4\varepsilon^{-2})$ by carefully reusing samples and avoiding unnecessary sampling. Zhang et al. (2021c) further proposed an improved algorithm `MVP` to achieve the near-optimal regret bound $\widetilde{O}(\sqrt{SAK} + S^2A)$ based on a new Bernstein-type bonus. Similar polylogarithmic dependency bounds H had been established by Ren et al. (2021) for linear MDP with anchor points, Tarbouriech et al. (2021) for the stochastic shortest path. Li et al. (2022) achieved the surprising H independent sample complexity bound $O((SA)^{O(S)}\varepsilon^{-5})$ by building a connection between discounted MDPs and episodic MDPs and a novel perturbation analysis in MDPs. The algorithm proposed by Zhang et al. (2022) further improved the sample complexity to $O(S^9A^3\varepsilon^{-2}\text{polylog}(S, A, \varepsilon^{-1}))$ only polynomially depending on the size of the state and the action spaces by exploiting the power of stationary policy. Thanks to the linear function approximation, Zhou and Gu (2022a) first achieves the horizon-free regret bound $\widetilde{O}(d\sqrt{K}+d^2)$ independently of the size of the state and action spaces. However, all the above works are limited to standard RL settings. In the paradigm of reward-free exploration, the only horizon-free result was achieved by Zhang et al. (2021b) with sample complexity bound of $\widetilde{O}(S^2A\varepsilon^{-2})$, where the polynomial dependency on S and A is still unacceptable when the state space and action space are large. Our algorithm HF-UCRL-RFE++ establishes the first horizon-free sample complexity bound independent of the size of the state space and action space in reward-free exploration.

2.3 Preliminaries

2.3.1 Episodic Markov Decision Processes

We consider episodic finite-horizon Markov Decision Processes (MDPs), which are denoted by a tuple $M(\mathcal{S}, \mathcal{A}, H, \{r_h\}_{h=1}^H, \mathbb{P})$. Here $\mathcal{S}$ is the countable state space (may be infinite), $\mathcal{A}$ is the action space, H is the length of the episode, and $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function. Without loss of generality, we assume that the reward function r_h is *deterministic*. $\mathbb{P}(s'|s, a)$ is the transition probability function that denotes the probability for state s to transit to state s' given the action a at step h. A policy $\pi_h : \mathcal{S} \to \mathcal{A}$ is a function that maps a state s to an action a. We define the action-value function (i.e., Q-function) $Q_h^\pi(s, a)$ as follows:

$$Q_h^\pi(s, a; \{r_h\}_h) = \mathbb{E}\bigg[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'})\bigg| s_h = s, a_h = a\bigg], V_h^\pi(s; \{r_h\}_h) = Q_h^\pi(s, \pi_h(s); \{r_h\}_h).$$

For simplicity, we denote $Q_h^\pi(s, a; r) = Q_h^\pi(s, a; \{r_h\}_h)$ and $V_h^\pi(s; r) = V_h^\pi(s; \{r_h\}_h)$. We define the optimal value function $\{V_h^*\}_{h=1}^H$ and the optimal state-action value function $\{Q_h^*\}_{h=1}^H$ as $V_h^*(s; r) = \sup_\pi V_h^\pi(s; r)$ and $Q_h^*(s, a; r) = \sup_\pi Q_h^\pi(s, a; r)$ respectively. For any function $V : \mathcal{S} \to \mathbb{R}$, we denote $[\mathbb{P}V](s, a; r) = \mathbb{E}_{s' \sim \mathbb{P}(\cdot|s,a)} V(s'; r)$, and denote the variance of V as

$$[\mathbb{V}f](s, a) = [\mathbb{P}f^2](s, a) - \big([\mathbb{P}f](s, a)^2\big). \tag{2.3.1}$$

In particular, we have the following Bellman equation, as well as the Bellman optimality equation:

$$Q_h^\pi(s, a; r) = r_h(s, a) + [\mathbb{P}V_{h+1}^\pi](s, a; r), Q_h^*(s, a; r) = r_h(s, a) + [\mathbb{P}V_{h+1}^*](s, a; r).$$

In this paper, we focus on *model-based* algorithms and consider the following *linear mixture/kernel MDP* (Modi et al., 2020; Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021d), which assumes that the transition probability $\mathbb{P}$ is a linear mixture of d signed basis measures. Meanwhile, for any function V, we assume that we can do the summation $\sum_{s' \in \mathcal{S}} \boldsymbol{\phi}(s'|s, a)V(s)$ efficiently, e.g., using the Monte Carlo method (Yang and Wang, 2020b).

Definition 2.3.1 (Linear Mixture MDPs (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2021d))**.** The unknown transition probability $\mathbb{P}$ is a linear combination of d signed basis measures $\phi_i(s'|s,a)$, that is, $\mathbb{P}(s'|s,a) = \sum_{i=1}^{d} \phi_i(s'|s,a)\theta_i^*$. Meanwhile, for any $V : \mathcal{S} \to [0,1]$, $i \in [d]$, $(s,a) \in \mathcal{S} \times \mathcal{A}$, the calculation of the summation $\sum_{s' \in \mathcal{S}} \phi_i(s'|s,a)V(s')$ is feasible. For simplicity, let $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_d]^\top$, $\boldsymbol{\theta}^* = [\theta_1^*, \ldots, \theta_d^*]^\top$ and $\boldsymbol{\psi}_V(s,a) = \sum_{s' \in \mathcal{S}} \boldsymbol{\phi}(s'|s,a)V(s')$. Without loss of generality, we assume $\|\boldsymbol{\theta}^*\|_2 \leqslant B$, $\|\boldsymbol{\psi}_V(s,a)\|_2 \leqslant 1$ for all $V : \mathcal{S} \to [0,1]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$.

Remark 2.3.2. A similar but notably different definition (i.e., linear MDPs (Yang and Wang, 2019; Jin et al., 2020b)) has been used in Wang et al. (2020b), which assumes that $\mathbb{P}(s'|s,a) = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}(s') \rangle$ and $r_h = \langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h \rangle$, $\boldsymbol{\mu}_h(\cdot)$ is a measure and $\boldsymbol{\theta}_h$ is an unknown vector. Comparing with linear MDPs, linear mixture MDPs do not need the reward function r to be linear, which makes our algorithms more general.

With Definition 2.3.1, it is easy to verify that the expectation of any bounded function V is a linear function of $\boldsymbol{\psi}$:

$$[\mathbb{P}V](s,a) = \langle \boldsymbol{\psi}_V(s,a), \boldsymbol{\theta}^* \rangle. \tag{2.3.2}$$

2.3.2 Formal Definition of Reward-Free Exploration

For reward-free exploration, the algorithm can be divided into two phases: *exploration phase* and *planning phase*. In the exploration phase, the algorithm cannot access the reward function but collect K episodes by doing exploration. In the planning phase, the algorithm is given a series of reward functions and find the optimal policy based on these reward functions, using the K episodes collected in the exploration phase. We formally define (ϵ, δ)-learn and sample complexity of the algorithm as follows (Jin et al., 2020a).

Definition 2.3.3 ((ϵ, δ)-learnability)**.** Given an MDP transition kernel set $\mathcal{P}$, reward function set $\mathcal{R}$ and a initial state distribution μ, we say a reward-free algorithm can (ϵ, δ)-learn

the problem $(\mathcal{P}, \mathcal{R})$ with sample complexity $K(\epsilon, \delta)$, if for any transition kernel $P \in \mathcal{P}$, after receiving $K(\epsilon, \delta)$ episodes in the exploration phase, for any reward function $r \in \mathcal{R}$, the algorithm returns a policy π in planning phase, such that with probability at least $1 - \delta$, $\mathbb{E}_{s_1 \sim \mu}[V_1^*(s_1; r) - V_1^\pi(s_1; r)] \leqslant \epsilon$.

2.4 Theoretical Guaranteed Reward-Free Exploration

In this section, we propose a reward-free algorithm. This algorithm works as follows: Firstly, during the *exploration phase*, it samples the MDP episodes, build an estimator $\boldsymbol{\theta}$ for the MDP parameter $\boldsymbol{\theta}^*$, and compute the covariance matrix $\boldsymbol{\Sigma}$ of the feature mappings, which characterizes the uncertainty of the estimator $\boldsymbol{\theta}$. Secondly, during the *planning phase*, the algorithm uses the collected $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ in the exploration phase to find the optimal policy π based on the given reward functions.

2.4.1 Proposed Algorithms

2.4.1.1 Planning Phase Algorithm

We first introduce the `PLAN` function (Algorithm 1), which is a common module in both planning phase and exploration phase. Given a series of reward functions $\{r_h\}_h$, the goal of `PLAN` function is to output the optimal policies $\{\pi_h\}_h$ and Q-functions $\{Q_h\}_h$ corresponding to $\{r_h\}_h$. Suppose the parameter $\boldsymbol{\theta}^*$ is known, we can compute $\{Q_h\}_h$ recursively by the following Bellman equation:

$$Q_h(s, a; r) = r_h(s, a) + [\mathbb{P}V_{h+1}](s, a; r) = r_h(s, a) + \langle \boldsymbol{\psi}_{V_{h+1}}(s, a), \boldsymbol{\theta}^* \rangle, \tag{2.4.1}$$

$Q_h(s, a; r)$ can be viewed as the summation of the reward function $r_h(s, a)$ and a linear function $\langle \boldsymbol{\psi}_{V_{h+1}}(s, a), \boldsymbol{\theta}^* \rangle$. However, since $\boldsymbol{\theta}^*$ is unknown, we cannot compute Q_h as in (2.4.1). Instead, `PLAN` takes the estimated parameter $\boldsymbol{\theta}$ and the "covariance matrix" $\boldsymbol{\Sigma}$ as input. To calculate Q_h, `PLAN` replaces $\boldsymbol{\theta}^*$ with the estimated $\boldsymbol{\theta}$ and plus an additional

Algorithm 1 UCRL-RFE Planning Module (`PLAN`)

Input: Estimated parameter and covariance $\boldsymbol{\theta}, \boldsymbol{\Sigma}$, reward $\{r_h\}_{h=1}^H$, parameter β.

1: For consistency, set $Q_{H+1}(\cdot,\cdot) \leftarrow V_{H+1}(\cdot) \leftarrow 0$

2: **for** $h = H, H-1, \cdots, 1$ **do**

3: Compute Q function as $Q_h(\cdot,\cdot) \leftarrow \big[r_h(\cdot,\cdot) + \big\langle \boldsymbol{\psi}_{V_{h+1}}(\cdot,\cdot), \boldsymbol{\theta} \big\rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(\cdot,\cdot)\|_{\boldsymbol{\Sigma}^{-1}}\big]_{(0,H)}$

4: Compute value function $V_h(\cdot) \leftarrow \max_{a\in\mathcal{A}} Q_h(\cdot, a)$

5: Compute policy as $\pi_h(\cdot) \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} Q_h(\cdot, a)$.

6: **end for**

Output: Policy $\pi \leftarrow \{\pi_h\}_{h=1}^H$ and $\{V_h\}_{h=1}^H$

exploration bonus term $\beta\|\boldsymbol{\psi}_{V_{h+1}}(\cdot,\cdot)\|_{\boldsymbol{\Sigma}^{-1}}$ to (2.4.1), as in Line 3 of Algorithm 1. Then `PLAN` takes the greedy policy of the calculated optimistic Q_h and proceeds to the previous step. Finally, the algorithm returns policy π in Line 5 as well as the estimated value functions $\{V_h\}_h$.

2.4.1.2 Exploration Phase Algorithm

Based on the introduced `PLAN` function, we propose the UCRL-RFE algorithm in Algorithm 2. In general, UCRL-RFE guides the agent to explore the unknown state space without the information of the reward functions. In detail, for the k-th episode, UCRL-RFE first defines the *exploration driven reward function* as follows:

$$r_h^k(s,a) = \min\left\{1, \frac{2\beta}{H}\sqrt{\max_{f\in\mathcal{S}\mapsto[0,H-h]} \|\boldsymbol{\psi}_f(s,a)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}}\right\}, \tag{2.4.2}$$

where $\boldsymbol{\Sigma}_{1,k}$ is the "covariance matrix" of the feature mapping. Intuitively speaking, $r_h^k(s,a)$ represents the maximum possible uncertainty level of the state-action pair (s,a) caused by the randomness of the MDP transition function, which is *independent* of the true reward functions. Therefore, in order to obtain a good estimation of the optimal policy for any *given* reward functions, it suffices to obtain the optimal policy for $r_h^k(s,a)$. Thus, after obtaining $\{r_h^k\}_h$, UCRL-RFE finds the corresponding near-optimal policies $\{\pi_h^k\}_h$ using `PLAN` function,

with the estimated parameter $\boldsymbol{\theta}_k$ and the "covariance matrix" $\boldsymbol{\Sigma}_{1,k}$ as input. UCRL-RFE uses $\{\pi_h^k\}_h$ as its exploration policy and observes the new episode $s_1^k, a_1^k, \ldots, s_H^k, a_H^k$ induced by $\{\pi_h^k\}_h$.

Next, UCRL-RFE needs to compute the parameters $\boldsymbol{\theta}_{k+1}$ and $\boldsymbol{\Sigma}_{1,k+1}$ for planning in the next episode. Similar to UCRL-VTR proposed by (Jia et al., 2020; Ayoub et al., 2020), UCRL-RFE also uses a "value-targeted regression (VTR)" estimator, which computes $\boldsymbol{\theta}_{k+1}$ as the minimizer to a ridge regression problem with the target being the past value functions. The main difference between UCRL-RFE and UCRL-VTR is that, due to the lack of true reward functions, UCRL-RFE can not use the estimated value functions as its regression targets. Instead, UCRL-RFE defines the following *pseudo value function* u_h^k:

$$u_h^k = \operatorname*{argmax}_{f \in \mathcal{S} \mapsto [0, H-h]} \boldsymbol{\psi}_f^\top(s_h^k, a_h^k) \boldsymbol{\Sigma}_{1,k}^{-1} \boldsymbol{\psi}_f(s_h^k, a_h^k). \tag{2.4.3}$$

Here, u_h^k maximizes the "uncertainty" caused by the transition kernel, which will help the agent to explore the state space. Now given the pseudo value functions, Algorithm 2 computes the estimated $\boldsymbol{\theta}_{k+1}$ as the minimizer to the following ridge regression problem:

$$\boldsymbol{\theta}_{k+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta}} \lambda \|\boldsymbol{\theta}\|_2^2 + \sum_{k'=1}^{k} \sum_{h=1}^{H} \Big(\big\langle \boldsymbol{\theta}, \boldsymbol{\psi}_{u_h^{k'}}(s_h^{k'}, a_h^{k'}) \big\rangle - u_h^{k'}(s_{h+1}^{k'}) \Big)^2, \tag{2.4.4}$$

which has a closed-form solution as in Line 12. It also updates the covariance matrix $\boldsymbol{\Sigma}_{1,k+1}$ as in Line 12, by the observed feature mapping $\{\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\}_h$ in the current episode. In the end, after collecting HK state-action samples, UCRL-RFE calculates the policy $\{\pi_h\}$ as output based on $\boldsymbol{\theta}_{K+1}$ and $\boldsymbol{\Sigma}_{1,K+1}$.

Remark 2.4.1. Here we do a comparison between our UCRL-RFE and the reward-free RL algorithm in Wang et al. (2020b). The main difference is that Wang et al. (2020b) estimates $\boldsymbol{\theta}_k$ by regression with the value function V_h^k being the target, while our UCRL-RFE does regression with the pseudo-value function u_h^k being the target. That is mainly due to the different problem settings (linear MDP vs. linear mixture MDP).

Algorithm 2 UCRL-RFE (Hoeffding Bonus)

Input: Confident parameter β, regularization parameter λ

Phase I: Exploration Phase

Initialize $\mathbf{\Sigma}_{1,1} \leftarrow \lambda \mathbf{I}, \mathbf{b}_1 \leftarrow \boldsymbol{\theta}_1 \leftarrow \mathbf{0}$

for $k = 1, 2, \cdots, K$ **do**

  Compute the exploration driven reward function $\{r_h^k(\cdot,\cdot)\}_{h=1}^H$ according to (2.4.2)

  Compute exploration policy and value function as $(\{\pi_h^k\}_{h=1}^H, \{V_h^k\}_{h=1}^H) \leftarrow \texttt{PLAN}(\boldsymbol{\theta}_k, \mathbf{\Sigma}_{1,k}, \{r_h^k\}_{h=1}^H, \beta)$

  Receive the initial state $s_1^k \sim \mu$

  for $h = 1, 2, \cdots, H$ **do**

    Take action $a_h^k \leftarrow \pi_h^k(s_h^k)$ and receive s_{h+1}^k

    Calculate u_h^k for s_h^k, a_h^k according to (2.4.3)

    Set $\mathbf{\Sigma}_{h+1,k} \leftarrow \mathbf{\Sigma}_{h,k} + \boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)^\top, \mathbf{b}_{h+1,k} \leftarrow \mathbf{b}_{h,k} + \boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)u_h^k(s_{h+1}^k)$

  end for

  Set $\mathbf{\Sigma}_{1,k+1} \leftarrow \mathbf{\Sigma}_{H+1,k}$, $\mathbf{b}_{1,k+1} \leftarrow \mathbf{b}_{H+1,k}, \boldsymbol{\theta}_{k+1} \leftarrow \mathbf{\Sigma}_{1,k+1}^{-1}\mathbf{b}_{1,k+1}$

end for

Phase II: Planning Phase

Receive target reward function $\{r_h\}_{h=1}^H$

Compute policy as $(\{\pi_h\}_{h=1}^H, \{V_h\}_{h=1}^H) \leftarrow \texttt{PLAN}(\boldsymbol{\theta}_{K+1}, \mathbf{\Sigma}_{1,K+1}, \{r_h\}_{h=1}^H, \beta)$

Output: Policy $\{\pi_h\}_{h=1}^H$

Remark 2.4.2 (Implementation Details)**.** In general, solving the maximization problem (2.4.3) is hard. Here, we provide a simple approximate solution to the problem (2.4.2) and (2.4.3) for the finite state space case ($|\mathcal{S}| < \infty$). Instead of maximizing the ℓ_2 norm-based objective $\left\|\mathbf{\Sigma}_{1,k}^{-1/2}\boldsymbol{\psi}_f(s_h^k, a_h^k)\right\|_2$, we write $\boldsymbol{\psi}_f(s,a) = \mathbf{\Phi}(s,a)\mathbf{f}$ with

$$\mathbf{\Phi}(s,a) = (\boldsymbol{\phi}(s,a,S_1), \cdots, \boldsymbol{\phi}(s,a,S_{|\mathcal{S}|})), \mathbf{f} = (f(S_1), \cdots, f(S_{|\mathcal{S}|}))^\top.$$

By relaxing the ℓ_2 norm into ℓ_1 norm due to $\|\mathbf{x}\|_2 \geqslant \|\mathbf{x}_1\|_1/\sqrt{d}$ for any $\mathbf{x} \in \mathbb{R}^d$, we reach a

surrogate objective:

$$\max_{\mathbf{f}} \left\| \mathbf{\Sigma}_{1,k}^{-1/2} \mathbf{\Phi}(s,a)\mathbf{f} \right\|_1 \text{ subject to } \|\mathbf{f}\|_\infty \leqslant H - h, \tag{2.4.5}$$

which can be further formulated as a linear programming problem, and solved by interior method (Karmarkar, 1984) or simplex method (Dantzig, 1965) efficiently. Since $\|\mathbf{x}\|_1/\sqrt{d} \leqslant \|\mathbf{x}\|_2 \leqslant \|\mathbf{x}\|_1$, the performance of this approximate solution is guaranteed. For the case where the state space is infinite, we can use state aggregation methods such as soft state aggregation (Michael and Jordan, 1995) to reduce the infinite state space to a finite state space and then apply the above approximate solution to solve it.

2.4.2 Sample Complexity Analysis

Now we provide the sample complexity for Algorithm 2.

Theorem 2.4.3 (Sample complexity of UCRL-RFE)**.** For Algorithm 2, setting parameter $\beta = H\sqrt{d\log(3(1+KH^3B^2)/\delta)}+1, \quad \lambda = B^{-2}$, then for any $0 < \epsilon < 1$, if $K = \widetilde{\mathcal{O}}(H^5d^2\epsilon^{-2})$, we have with probability at least $1-\delta$ that, for any reward function r, Algorithm 2 produces a policy π with $\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] \leqslant \epsilon$.

Remark 2.4.4. Theorem 2.4.3 shows that UCRL-RFE only needs $\text{poly}(d, H, \epsilon^{-1})$ sample complexity to find an ϵ-optimal policy, which suggests that model-based reward-free algorithm is sample-efficient. Thanks to the linear function approximation, the sample complexity depends only on the dimension of the feature mapping d and the length of the episode and does not depend on the cardinality of the state and action spaces.

Corollary 2.4.5. Under the same conditions as in Theorem 2.4.3, if solving the relaxed optimization problem in (2.4.5), Algorithm 2 has $K = \widetilde{\mathcal{O}}(H^5d^3\epsilon^{-2})$ sample complexity.

2.5 Improved Algorithm and Analysis with Variance Information

Theorem 2.4.3 suggests that UCRL-RFE in Algorithm 2 enjoys an $\widetilde{\mathcal{O}}(H^5d^2\epsilon^{-2})$ sample complexity to find an ϵ-optimal policy. In this section, we seek to further improve the sample complexity.

A key observation is that for any given reward function $\{r_h\}_h$, the error between the exploration policy $\{\pi_h\}_h$ and the optimal policy can be decomposed into two parts: *the exploration error* which is the difference between $\{r_h\}_h$ and the exploration-driven reward function $\{r_h^k\}_h$, and *the approximation error* which is the difference between the optimal value function $V_1^*(\cdot; r_h^k)$ and our estimated value function $V_1^{\pi_h^k}(\cdot; r_h^k)$ with respect to $\{r_h^k\}_h$. For the latter, our exploration strategy adapted from VTR is often too conservative since it does not distinguish different value functions and state-action pairs from different episodes and steps. Therefore, inspired by (Zhou et al., 2021a), we propose a variant of UCRL-RFE called UCRL-RFE+, which adopts a Bernstein-type bonus for exploration and achieves a better sample complexity.

2.5.1 Exploration Phase Algorithm with Variance Information

UCRL-RFE+ is presented in Algorithm 3. The structure of the algorithm is similar to that of UCRL-RFE, which can be decomposed into the exploration phase and the planning phase. There are two main differences. First, in contrast to UCRL-RFE which uses $\boldsymbol{\theta}_k$ for the `PLAN` function in both exploration and planning phases, UCRL-RFE+ only uses $\boldsymbol{\theta}_{K+1}$ for the `PLAN` function in the planning phase. For the exploration phase, UCRL-RFE+ constructs a new estimator $\widehat{\boldsymbol{\theta}}_k$ based on $\{V_{h+1}^{k'}\}_{k' \leqslant k-1,h}$, which are the value functions of the exploration-driven rewards. Second, to build $\widehat{\boldsymbol{\theta}}_k$, one way is to choose it as a solution to the ridge regression problem with contexts $\boldsymbol{\psi}_{V_{h+1}^{k'}}(s_h^{k'}, a_h^{k'})$ and targets $V_{h+1}^{k'}(s_{h+1}^{k'})$, similar to (2.4.4). However, since the targets $V_{h+1}^{k'}(s_{h+1}^{k'})$ have different variances at different steps and episodes, we are actually facing a *heteroscedastic linear regression* problem. Therefore, inspired by a recent

Algorithm 3 UCRL-RFE+ (Bernstein Bonus)

Input: Parameter $\beta, \widehat{\beta}, \widetilde{\beta}, \check{\beta}$, regularization parameter λ

Stage I: Exploration Phase

Initialize $\mathbf{\Sigma}_{1,1} = \widehat{\mathbf{\Sigma}}_{1,1} = \widetilde{\mathbf{\Sigma}}_{1,1} = \lambda \mathbf{I}, \mathbf{b}_1 = \widehat{\mathbf{b}}_1 = \widetilde{\mathbf{b}}_1 = \boldsymbol{\theta}_1 = \widehat{\boldsymbol{\theta}}_1 = \widetilde{\boldsymbol{\theta}}_1 = \mathbf{0}$

for $k = 1, 2, \cdots, K$ **do**

  Set $\{r_h^k(\cdot,\cdot)\}_{h=1}^H$ to (2.4.2).

  Set $(\{\pi_h^k\}_{h=1}^H, \{V_h^k\}_{h=1}^H) \leftarrow \texttt{PLAN}(\widehat{\boldsymbol{\theta}}_k, \widehat{\mathbf{\Sigma}}_{1,k}, \{r_h^k\}_{h=1}^H, \widehat{\beta})$

  Receive the initial state $s_1^k \sim \mu$.

  for $h = 1, 2, \cdots, H$ **do**

    Take action $a_h^k = \pi_h^k(s_h^k)$ and receive s_{h+1}^k

    Calculate u_h^k, ν_h^k for s_h^k, a_h^k according to (2.4.3) and (2.5.2) separately

    Set $\mathbf{\Sigma}_{h+1,k} \leftarrow \mathbf{\Sigma}_{h,k} + \boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)^\top$

    Set $\widehat{\mathbf{\Sigma}}_{h+1,k}$, $\widetilde{\mathbf{\Sigma}}_{h+1,k}$, $\widehat{\mathbf{b}}_{h+1,k}$, $\widetilde{\mathbf{b}}_{h+1,k}$ using (2.5.4)

  end for

  Set $\mathbf{\Sigma}_{1,k+1} \leftarrow \mathbf{\Sigma}_{H+1,k}$

  Set $\widehat{\mathbf{\Sigma}}_{1,k+1} \leftarrow \widehat{\mathbf{\Sigma}}_{H+1,k}$, $\widehat{\mathbf{b}}_{1,k+1} \leftarrow \widehat{\mathbf{b}}_{H+1,k}, \widehat{\boldsymbol{\theta}}_{k+1} \leftarrow \widehat{\mathbf{\Sigma}}_{1,k+1}^{-1}\widehat{\mathbf{b}}_{1,k+1}$

  Set $\widetilde{\mathbf{\Sigma}}_{1,k+1} \leftarrow \widetilde{\mathbf{\Sigma}}_{H+1,k}$, $\widetilde{\mathbf{b}}_{1,k+1} \leftarrow \widetilde{\mathbf{b}}_{H+1,k}, \widetilde{\boldsymbol{\theta}}_{k+1} \leftarrow \widetilde{\mathbf{\Sigma}}_{1,k+1}^{-1}\widetilde{\mathbf{b}}_{1,k+1}$

end for

Set $\boldsymbol{\theta}_{K+1} \leftarrow \mathbf{\Sigma}_{1,K+1}^{-1} \sum_{k=1}^K \sum_{h=1}^H \boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k) u_h^k(s_{h+1}^k)$

Stage II: Planning Phase

Receive target reward function $\{r_h\}_{h=1}^H$

Compute the exploration policy as $(\{\pi_h\}_{h=1}^H, \{V_h\}_{h=1}^H) \leftarrow \texttt{PLAN}(\boldsymbol{\theta}_{K+1}, \mathbf{\Sigma}_{1,K+1}, \{r_h\}_{h=1}^H, \beta)$

Output: Policy $\{\pi_h\}_{h=1}^H$

line of work Zhou et al. (2021a); Wu et al. (2021) using Bernstein inequality for vector-valued self-normalized martingale to construct a tighter confidence ball for exploration, we also incorporate the variance to build choose $\widehat{\boldsymbol{\theta}}_k$ as the solution to the following *weighted*

ridge regression problem, which is an enhanced estimator for the heteroscedastic case:

$$\widehat{\boldsymbol{\theta}}_k \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta}} \lambda\|\boldsymbol{\theta}\|_2^2 + \sum_{k'=1}^{k-1}\sum_{h=1}^{H}\Big(\big\langle \boldsymbol{\theta}, \boldsymbol{\psi}_{V_{h+1}^{k'}}(s_h^{k'}, a_h^{k'})\big\rangle - V_{h+1}^{k'}(s_{h+1}^{k'})\Big)^2/[\sigma_h^{k'}]^2, \tag{2.5.1}$$

where $[\sigma_h^{k'}]^2$ is the variance of $V_{h+1}^{k'}(s_{h+1}^{k'})$. The idea of using variances to improve the sample complexity is closely related to the use of "Bernstein bonus" in reward-free RL for the tabular MDPs (Kaufmann et al., 2021a; Zhang et al., 2020; Ménard et al., 2020). Since $\sigma_h^{k'}$ is unknown, we will use $\nu_h^{k'} = [\bar{\sigma}_h^{k'}]^2$ as a plug-in estimator to replace $[\sigma_h^{k'}]^2$ in (2.5.1). After obtaining $\widehat{\boldsymbol{\theta}}_k$, UCRL-RFE+ sets $\widehat{\boldsymbol{\Sigma}}_{1,k}$ as the covariance matrix of the features $\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)/\bar{\sigma}_h^k$, and feeds it into the `PLAN` function with the exploration-driven reward functions and the confidence radius $\widehat{\beta}$. UCRL-RFE+ takes the output $\{\pi_h^k\}_h$ as the exploration policy and $\{V_h^k\}_h$ as the value functions to construct the estimator $\widehat{\boldsymbol{\theta}}_{k+1}$ for the next episode. In the end, when it comes to the planning phase, after receiving the reward functions $\{r_h\}_h$, UCRL-RFE+ takes $\boldsymbol{\theta}_{K+1}$ as the solution to the ridge regression problem with contexts $\{\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\}_{k,h}$ and targets $\{u_h^k(s_{h+1}^k)\}_{k,h}$, and the covariance matrix $\boldsymbol{\Sigma}_{1,K+1}$ as input, and uses `PLAN` to find the near-optimal policy $\{\pi_h\}_h$ with confidence radius β. It remains to specify ν_h^k in the weighted ridge regression. On the one hand, we need ν_h^k to be an upper bound of $[\sigma_h^k]^2$. On the other hand, we require ν_h^k to have a strictly positive lower bound to let (2.5.1) be valid. Therefore, we construct ν_h^k as follows:

$$\nu_h^k = \max\{\alpha, \bar{\mathbb{V}}_h^k(s_h^k, a_h^k) + E_k^h(s_h^k, a_h^k)\}, \tag{2.5.2}$$

where $\bar{\mathbb{V}}_h^k$ is the estimated variance of value function V_h^k and E_h^k is a correction term to calibrate the estimated variance, and $\alpha > 0$ is a positive constant. To compute $\bar{\mathbb{V}}_h^k(s_h^k, a_h^k)$, consider the following fact:

$$[\mathbb{V}V_{h+1}^k](s,a) = [\mathbb{P}[V_{h+1}^k]^2](s,a) - [\mathbb{P}V_{h+1}^k](s,a)^2 = \langle \boldsymbol{\theta}^*, \boldsymbol{\psi}_{[V_{h+1}^k]^2}(s,a)\rangle - \langle \boldsymbol{\theta}^*, \boldsymbol{\psi}_{V_{h+1}^k}(s,a)\rangle^2,$$

it suffices to estimate $\langle \boldsymbol{\theta}^*, \boldsymbol{\psi}_{[V_{h+1}^k]^2}(s,a)\rangle$ and $\langle \boldsymbol{\theta}^*, \boldsymbol{\psi}_{V_{h+1}^k}(s,a)\rangle$ separately. For the first term, $\boldsymbol{\theta}^*$ can be regarded as the unknown parameter of a regression problem between contexts

$\boldsymbol{\psi}_{[V_{h+1}^{k'}]^2}(s_h^{k'}, a_h^{k'})$ and targets $\boldsymbol{\psi}_{[V_{h+1}^{k'}]^2}(s_h^{k'}, a_h^{k'})$. Therefore, the first term can be estimated by $\big\langle \boldsymbol{\psi}_{[V_{h+1}^{k}]^2}(s,a), \widetilde{\boldsymbol{\theta}}_k \big\rangle$, where

$$\widetilde{\boldsymbol{\theta}}_k \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta}} \lambda \|\boldsymbol{\theta}\|_2^2 + \sum_{k'=1}^{k-1} \sum_{h=1}^{H} \Big(\big\langle \boldsymbol{\theta}, \boldsymbol{\psi}_{[V_{h+1}^{k'}]^2}(s_h^{k'}, a_h^{k'}) \big\rangle - [V_{h+1}^{k'}(s_{h+1}^{k'})]^2 \Big)^2.$$

In addition, the second term $\langle \boldsymbol{\theta}^*, \boldsymbol{\psi}_{V_{h+1}^k}(s,a) \rangle$ can be approximated by $\langle \boldsymbol{\psi}_{V_{h+1}^k}(s,a), \widehat{\boldsymbol{\theta}}_k \rangle$. Therefore, the final estimator $[\bar{\mathbb{V}} V_{h+1}^k](s,a)$ is defined as

$$\bar{\mathbb{V}}_h^k(s,a) = \Big[\big\langle \boldsymbol{\psi}_{[V_{h+1}^{k}]^2}(s,a), \widetilde{\boldsymbol{\theta}}_k \big\rangle \Big]_{(0,H^2)} - \Big[\big\langle \boldsymbol{\psi}_{V_{h+1}^{k}}(s,a), \widehat{\boldsymbol{\theta}}_k \big\rangle \Big]_{(0,H)}^2. \tag{2.5.3}$$

For the correction terms E_h^k, we define it as follows:

$$E_h^k(s,a) = \min \Big\{ H^2, \widetilde{\beta} \big\| \boldsymbol{\psi}_{[V_{h+1}^{k}]^2}(s,a) \big\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}} \Big\} + \min \Big\{ H^2, 2H\check{\beta} \big\| \boldsymbol{\psi}_{V_{h+1}^{k}}(s,a) \big\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}} \Big\},$$

where $\widetilde{\boldsymbol{\Sigma}}_{1,k}$ is the covariance matrix of the features $\boldsymbol{\psi}_{[V_{h+1}^{k'}]^2}(s_h^{k'}, a_h^{k'})$, $\widetilde{\beta}$, $\check{\beta}$ are two confidence radius. It can be shown that, with these definitions, $\bar{\mathbb{V}}_h^k(s,a) + E_h^k(s,a)$ is an upper bound of $[\sigma_h^k]^2$.

Finally, to enable online update, UCRL-RFE+ updates its covariance matrices recursively as follows, along with sequences $\widehat{\mathbf{b}}_h^k, \widetilde{\mathbf{b}}_h^k$:

$$\begin{aligned}
\widehat{\boldsymbol{\Sigma}}_{h+1,k} &\leftarrow \widehat{\boldsymbol{\Sigma}}_{h,k} + \boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k) \boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)^\top / \nu_h^k \\
\widetilde{\boldsymbol{\Sigma}}_{h+1,k} &\leftarrow \widetilde{\boldsymbol{\Sigma}}_{h,k} + \boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k) \boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)^\top \\
\widehat{\mathbf{b}}_{h+1,k} &\leftarrow \widehat{\mathbf{b}}_{h,k} + \boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k) V_{h+1}^k(s_{h+1}^k) / \nu_h^k \\
\widetilde{\mathbf{b}}_{h+1,k} &\leftarrow \widetilde{\mathbf{b}}_{h,k} + \boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k) [V_{h+1}^k(s_{h+1}^k)]^2,
\end{aligned} \tag{2.5.4}$$

where u_h^k is the pseudo value function in (2.4.3) and ν_h^k is defined in (2.5.2). Then UCRL-RFE+ computes $\widehat{\boldsymbol{\theta}}_k, \widetilde{\boldsymbol{\theta}}_k$ as in Line 14 to Line 15 of Algorithm 3.

2.5.2 Sample Complexity Analysis

Now we present the sample complexity for Algorithm 3.

Theorem 2.5.1 (Sample complexity of UCRL-RFE+)**.** For Algorithm 3, setting $\lambda = B^{-2}$, $\alpha = H^2/d$ in (2.5.2), and the confidence radius as

$$\begin{aligned}
\widehat{\beta} &= 8\sqrt{d\log(1+KHB^2)\log(48K^2H^2/\delta)} + 4\sqrt{d}\log(48K^2H^2/\delta) + 1 \\
\check{\beta} &= 8d\sqrt{\log(1+KHB^2)\log(48K^2H^2/\delta)} + 4\sqrt{d}\log(48K^2H^2/\delta) + 1 \\
\widetilde{\beta} &= 8H^2\sqrt{d\log(1+KHB^2)\log(48K^2H^2/\delta)} + 4H^2\log(48K^2H^2/\delta) + 1 \\
\beta &= H\sqrt{d\log(12(1+KH^3B^2)/\delta)} + 1,
\end{aligned}$$

then for any $0 < \epsilon < 1$, if $K = \widetilde{\mathcal{O}}(H^4d(H+d)\epsilon^{-2})$, then with probability at least $1-\delta$, for any reward function r, Algorithm 2 outputs a policy π with $\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] \leqslant \epsilon$.

Remark 2.5.2. Theorem 2.5.1 suggests that when $d \geqslant H$, the sample complexity of UCRL-RFE+ is $\widetilde{\mathcal{O}}(H^4d^2\epsilon^{-2})$, which improves the sample complexity of UCRL-RFE by a factor of H. On the other hand, when $H \geqslant d$, the sample complexity of UCRL-RFE+ reduces to $\widetilde{\mathcal{O}}(H^5d\epsilon^{-2})$, which is better than that of UCRL-RFE by a factor of d. At a high level, the sample complexity improvement is attributed to the Bernstein-type bonus.

Corollary 2.5.3. Under the same conditions as in Theorem 2.5.1, if solving the relaxed optimization problem in (2.4.5), Algorithm 3 has $K = \widetilde{\mathcal{O}}(H^5d^3\epsilon^{-2})$ sample complexity.

2.6 Optimal Horizon-Free Reward-Free Exploration Algorithms

In Section 2.4 and 2.5, we have discussed the reward-free exploration when the reward function is bounded by $r_h(s,a) \in [0,1]$. Therefore, the total reward collected over H steps is bounded by $\sum_{h=1}^H r_h(s,a) \leqslant H$ and the sample complexity presented in Theorem 2.4.3 and Theorem 2.5.1 have a high dependence on H. In this section, we extend the reward-free exploration algorithms to the *bounded total reward* setting and aim to remove the dependence of H in this situation. Therefore, we assume that the accumulated reward of an episode for any trajectory is upper bounded by 1, which ensures that the only factors affecting the final

statistical complexity are difficulties brought by exploration and long planning horizon rather than the scale of the total reward.

Assumption 2.6.1. (Bounded total reward) For any trajectory $\{s_h, a_h\}_{h=1}^H$, we have $0 \leqslant \sum_{h=1}^H r_h(s_h, a_h) \leqslant 1$. We denote the set of reward functions that satisfy this by $\mathcal{R}$.

Following Zhou and Gu (2022b), we propose an exploration algorithm called HF-UCRL-RFE++ using high-order estimation to get the horizon-free sample complexity in the regime of reward-free exploration.

2.6.1 Proposed Algorithms

In this section, we propose our reward-free exploration algorithm HF-UCRL-RFE++. This algorithm consists of two phases. In the exploration phase, it builds an estimator $\boldsymbol{\theta}$ for the linear mixture MDP transition kernel parameter $\boldsymbol{\theta}^*$ based on the sampled episodes. At a high level, the estimation follows the *value-targeted regression* (VTR) framework proposed by Jia et al. (2020). The VTR is basically a ridge regression with value functions as responses and feature mappings as predictors. However, value functions do not have estimates since the reward function is not accessible. Therefore, the value functions and reward functions are replaced by well-designed exploration-driven pseudo-value functions and pseudo-reward functions. To achieve a better estimation, we further apply the *high-order moment estimation* (HOME) technique proposed by Zhou and Gu (2022a). Then, during the planning phase, the algorithm uses the estimator $\boldsymbol{\theta}$ acquired in the exploration phase to find the optimal policy π for the given reward functions. Our algorithm is described in Algorithm 4.

2.6.1.1 Exploration-driven Pseudo Value Function

As mentioned above, in the paradigm of reward-free exploration, we have to construct the pseudo-reward function to guide the agent in taking actions in the absence of the real reward function. As we adopt in this work, the most natural idea is to construct the pseudo-reward

Algorithm 4 HF-UCRL-RFE++ (High-order Estimation)

Input: Confidence radius $\{\beta_k\}$, regularization λ, number of the high-order estimator M.

Phase I: Exploration Phase

Initialize $\widehat{\boldsymbol{\Sigma}}_{1,1,m} = \widetilde{\boldsymbol{\Sigma}}_{1,1,m} = \lambda\mathbf{I}$, $\widetilde{\mathbf{b}}_{1,1,m} = \widehat{\mathbf{b}}_{1,1,m} = \mathbf{0}$ for all $m \in \overline{[M]}$, $\mathcal{U}_1 = \big\{\boldsymbol{\theta} | \boldsymbol{\theta} \in \mathbb{R}^d\big\}$.

Set $\widehat{\boldsymbol{\theta}}_{1,m} \leftarrow \widehat{\boldsymbol{\Sigma}}_{1,1,m}^{-1}\widehat{\mathbf{b}}_{1,1,m}$, $\widetilde{\boldsymbol{\theta}}_{1,m} \leftarrow \widetilde{\boldsymbol{\Sigma}}_{1,1,m}^{-1}\widetilde{\mathbf{b}}_{1,1,m}$ for all $m \in \overline{[M]}$.

for $k = 1, 2, \cdots, K$ **do**

  Set $\pi_k, \boldsymbol{\theta}_k, r_k = \operatorname{argmax}_{\pi, \boldsymbol{\theta} \in \mathcal{U}_k, r \in R} \widehat{V}_{k,1}(s_1; \boldsymbol{\theta}, \pi, r)$, $\widehat{V}_{k,1}$ is defined in (2.6.2).

  Denote $\big\{\widetilde{V}_{k,h}(\cdot)\big\}_{h=1}^{H} = \{V_h(\cdot; \boldsymbol{\theta}_k, \pi_k, r_k)\}_{h=1}^{H}$. Receive initial state $s_1^k = s_1$.

  for $h = 1, 2, \cdots, H$ **do**

    Execute $a_h^k = \pi_h^k\left(s_h^k\right)$, receive $s_{h+1}^k \sim \mathbb{P}\left(\cdot | s_h^k, a_h^k\right)$.

    For $m \in \overline{[M]}$, denote $\widehat{\boldsymbol{\phi}}_{k,h,m} = \boldsymbol{\phi}_{\widehat{V}_{k,h+1}^{2^m}}(s_h^k, a_h^k)$, $\widetilde{\boldsymbol{\phi}}_{k,h,m} = \boldsymbol{\phi}_{\widetilde{V}_{k,h+1}^{2^m}}(s_h^k, a_h^k)$.

    Set $\big\{\widehat{\sigma}_{k,h,m}\big\} \leftarrow \mathrm{HOME}_{\text{Alg. 5}}\Big(\big\{\widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}, \widehat{\boldsymbol{\Sigma}}_{k,h,m}, \dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}\big\}, \beta_k, \alpha, \gamma\Big)$.

    Set $\big\{\widetilde{\sigma}_{k,h,m}\big\} \leftarrow \mathrm{HOME}_{\text{Alg. 5}}\Big(\big\{\widetilde{\boldsymbol{\phi}}_{k,h,m}, \widetilde{\boldsymbol{\theta}}_{k,m}, \widetilde{\boldsymbol{\Sigma}}_{k,h,m}, \dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m}\big\}, \beta_k, \alpha, \gamma\Big)$.

    Set $\widetilde{\boldsymbol{\Sigma}}_{k,h+1,m} \leftarrow \widetilde{\boldsymbol{\Sigma}}_{k,h,m} + \widetilde{\boldsymbol{\phi}}_{k,h,m}\widetilde{\boldsymbol{\phi}}_{k,h,m}^{\top}\widetilde{\sigma}_{k,h,m}^{-2}$ for $m \in \overline{[M]}$.

    Set $\widehat{\boldsymbol{\Sigma}}_{k,h+1,m} \leftarrow \widehat{\boldsymbol{\Sigma}}_{k,h,m} + \widehat{\boldsymbol{\phi}}_{k,h,m}\widehat{\boldsymbol{\phi}}_{k,h,m}^{\top}\widehat{\sigma}_{k,h,m}^{-2}$ for $m \in \overline{[M]}$.

    Set $\widetilde{\mathbf{b}}_{k,h+1,m} \leftarrow \widetilde{\mathbf{b}}_{k,h,m} + \widetilde{\boldsymbol{\phi}}_{k,h,m}\widetilde{V}_{k,h+1}^{2^m}(s_{h+1}^k)\widetilde{\sigma}_{k,h,m}^{-2}$ for $m \in \overline{[M]}$.

    Set $\widehat{\mathbf{b}}_{k,h+1,m} \leftarrow \widehat{\mathbf{b}}_{k,h,m} + \widehat{\boldsymbol{\phi}}_{k,h,m}\widehat{V}_{k,h+1}^{2^m}(s_{h+1}^k)\widehat{\sigma}_{k,h,m}^{-2}$ for $m \in \overline{[M]}$.

  end for

  $\dot{\widetilde{\boldsymbol{\Sigma}}}_{k+1,m} \leftarrow \widetilde{\boldsymbol{\Sigma}}_{k,H+1,m}$, $\dot{\widehat{\boldsymbol{\Sigma}}}_{k+1,m} \leftarrow \widehat{\boldsymbol{\Sigma}}_{k,H+1,m}$.

  Set $\widetilde{\boldsymbol{\Sigma}}_{k+1,1,m} \leftarrow \widetilde{\boldsymbol{\Sigma}}_{k,H+1,m}, \widetilde{\mathbf{b}}_{k+1,1,m} \leftarrow \widetilde{\mathbf{b}}_{k,H+1,m}$, $\widetilde{\boldsymbol{\theta}}_{k+1,m} = \widetilde{\boldsymbol{\Sigma}}_{k+1,1,m}^{-1}\widetilde{\mathbf{b}}_{k+1,1,m}$.

  Set $\widehat{\boldsymbol{\Sigma}}_{k+1,1,m} \leftarrow \widehat{\boldsymbol{\Sigma}}_{k,H+1,m}, \widehat{\mathbf{b}}_{k+1,1,m} \leftarrow \widehat{\mathbf{b}}_{k,H+1,m}$, $\widehat{\boldsymbol{\theta}}_{k+1,m} = \widehat{\boldsymbol{\Sigma}}_{k+1,1,m}^{-1}\widehat{\mathbf{b}}_{k+1,1,m}$.

  Update the confidence set $\mathcal{U}_k$ to $\mathcal{U}_{k+1}$ by adding constraints (2.6.5), (2.6.6).

end for

Phase II: Planning Phase

Receive reward function r and return policy $\widehat{\pi}_r = \arg\max_{\pi} V_1(\cdot; \boldsymbol{\theta}_K, \pi, r)$.

function related to uncertainty, which urges the agent to collect information about the most uncertain states and actions. Two approaches follow this idea: one is constructing the pseudo

reward function directly measuring and maximizing the uncertainty of each stage, and the other is constructing the pseudo reward function maximizing the overall uncertainty along trajectories. Zhang et al. (2021e) took the first approach, constructing the pseudo-reward function in the form of

$$r_h^k(s,a) = \min\Big\{1, \frac{2\beta}{H}\sqrt{\max_{V\in\mathcal{S}\to[0,H-h]} \|\boldsymbol{\phi}_V(s,a)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}}\Big\},$$

and the pseudo-value function to be the argument of the maxima for the above uncertainty measure. Under this construction, the suboptimality in the planning phase can be bounded by the accumulation of uncertainty. This approach is straightforward but has the following two drawbacks. Firstly, without the truncation for accumulation of uncertainty, the upper bound of overall suboptimality in the planning phase will be in the scale of $O(H)$, which is meaningless since the value function lies in the interval of $[0,1]$ under our assumption. Second, since VTR utilizes value functions' variance information for $\boldsymbol{\theta}$ estimation, it requires a Bellman-equation-type equality between two consecutive stages h and $h+1$. However, the first approach does not satisfy this requirement, preventing us from acquiring a more accurate estimate.

To address the above issues, we follow the design of pseudo value function proposed in Chen et al. (2021). In particular, we are constructing the pseudo-reward function aiming to maximize the overall uncertainty along trajectories. We view the uncertainty of states and actions as a function of (pseudo) reward function r, policy π, and transition kernel parameter $\boldsymbol{\theta}$ defined as follows

$$u_{k,h}(s,a;\boldsymbol{\theta},\pi,r) = \min\Big\{1, \beta\big\|\boldsymbol{\phi}_{V_h(\cdot;\boldsymbol{\theta},\pi,r)}(s,a)\big\|_{\dot{\tilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\Big\}, \tag{2.6.1}$$

where $V_h(\cdot;\boldsymbol{\theta},\pi,r)$ is the the value function of policy π for linear mixture MDP with transition kernel parameter $\boldsymbol{\theta}$ and the reward function r, and the overall uncertainty along the trajectory is the truncated sum of each step uncertainty defined as

$$\bar{V}_{k,h}(s;\boldsymbol{\theta},\pi,r) = \min\Big\{1, u_{k,h}(s,\pi(s);\boldsymbol{\theta},\pi,r) + \boldsymbol{\phi}_{\bar{V}_{k,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}^{\top}(s,\pi(s))\boldsymbol{\theta}^*\Big\}.$$

However, the definition of $\bar{V}_{k,h}(s;\boldsymbol{\theta},\pi,r)$ involves $\boldsymbol{\theta}^*$, which is unknown to the agent. Hence, we construct the optimistic estimation of $\bar{V}_{k,h}(s;\boldsymbol{\theta},\pi,r)$ as $\widehat{V}_{k,h}(s;\boldsymbol{\theta},\pi,r)$ defined as

$$\widehat{V}_{k,h}(s;\boldsymbol{\theta},\pi,r) = \min\Big\{1, u_{k,h}(s,\pi(s);\boldsymbol{\theta},\pi,r) + 2\beta\big\|\boldsymbol{\phi}_{\widehat{V}_{k,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\big\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} + \boldsymbol{\phi}^{\top}_{\widehat{V}_{k,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\boldsymbol{\theta}\Big\}. \tag{2.6.2}$$

Notable, the definitions of $u_{k,h}$ and $\widehat{V}_{k,h}$ involve the covariance matrices $\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}$ and $\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}$, which are computed at the end of the preceding episode at Line 17 of Algorithm 4. In the following content, when there is no confusion, we may write $\widehat{V}_{k,h}(\cdot) = \widehat{V}_{k,h}(\cdot;\boldsymbol{\theta}_k,\pi_k,r_k)$, $u_{k,h}(\cdot,\cdot) = u_{k,h}(\cdot,\cdot;\boldsymbol{\theta}_k,\pi_k,r_k)$. In order to collect more information, the agent is expected to transit through the trajectory with the largest uncertainty $\widehat{V}_{k,h}$. It is notable that $\widehat{V}_{k,h}$ is a function of (pseudo) reward function r_k, policy π_k, and transition kernel parameter $\boldsymbol{\theta}_k$. Thus, at the beginning of each episode, we set r_k, π_k, and $\boldsymbol{\theta}_k$ to be arguments of the maxima, as presented in Line 5 in Algorithm 4. Through this process, we acquire the pseudo value function r_k, which is essential for reward-free exploration. Afterward, the algorithm collects samples along trajectories induced by policy π_k and improves the estimation of $\boldsymbol{\theta}_k$ in Line 6 to Line 21. In this stage, Algorithm 4 encounters two series of functions in the form of Bellman equations; one is the sum of pseudo rewards r, $\widetilde{V}_{k,h}(\cdot) = V_h(\cdot;\boldsymbol{\theta}_k,\pi_k,r_k)$, which we refer as pseudo value function, and one is the uncertainty along the trajectory, $\widehat{V}_{k,h}$. These two series of functions are both eligible for refined VTR and thus help estimate $\boldsymbol{\theta}$, as we will explain in the following.

Algorithm 5 High-order moment estimator (HOME)

Input: Features $\{\boldsymbol{\phi}_{k,h,m}\}_{m\in\overline{[M]}}$, vector estimators $\{\boldsymbol{\theta}_{k,m}\}_{m\in\overline{[M]}}$, covariance matrix $\left\{\boldsymbol{\Sigma}_{k,h,m}, \dot{\boldsymbol{\Sigma}}_{k,m}\right\}_{m\in\overline{[M]}}$, confidence radius β_k, α, γ.

1: **for** $m = 0, \ldots, M-2$ **do**

2: Set $\left[\overline{\mathbb{V}}_{k,m} V_{k,h+1}^{2^m}\right](s_h^k, a_h^k) \leftarrow \left[\boldsymbol{\phi}_{k,h,m+1}^\top \boldsymbol{\theta}_{k,m+1}\right]_{[0,1]} - \left[\boldsymbol{\phi}_{k,h,m}^\top \boldsymbol{\theta}_{k,m}\right]_{[0,1]}^2$.

3: Set $E_{k,h,m} \leftarrow \left[2\beta_k \left\|\boldsymbol{\phi}_{k,h,m}\right\|_{\dot{\boldsymbol{\Sigma}}_{k,m}^{-1}}\right]_{[0,1]} + \left[\beta_k \left\|\boldsymbol{\phi}_{k,h,m+1}\right\|_{\dot{\boldsymbol{\Sigma}}_{k,m+1}^{-1}}\right]_{[0,1]}$.

4: Set $\overline{\sigma}_{k,h,m}^2 \leftarrow \max\left\{\gamma^2 \left\|\boldsymbol{\phi}_{k,h,m}\right\|_{\boldsymbol{\Sigma}_{k,h,m}^{-1}}, \left[\overline{\mathbb{V}}_{k,m} V_{k,h+1}^{2^m}\right](s_h^k, a_h^k) + E_{k,h,m}, \alpha^2\right\}$.

5: **end for**

6: Set $\overline{\sigma}_{k,h,M-1}^2 \leftarrow \max\left\{\gamma^2 \left\|\boldsymbol{\phi}_{k,h,M-1}\right\|_{\boldsymbol{\Sigma}_{k,h,M-1}^{-1}}, 1, \alpha^2\right\}$.

Output: $\{\overline{\sigma}_{k,h,m}\}_{m\in\overline{[M]}}$.

2.6.1.2 High-order Moment Estimation

The key technique used in our algorithm consists of two series of high-order estimations for the transition kernel parameter $\boldsymbol{\theta}$. The algorithm for high-order moment estimation is stated in Algorithm 5. In the exploration phase, the agent learns the environment with the help of two series of value functions $\widetilde{V}_{k,h}$ and $\widehat{V}_{k,h}$. They serve to characterize different aspects of the model, one for pseudo values and one for trajectory uncertainty. And thus, they rely on different estimations of transition kernel parameter $\boldsymbol{\theta}$. Two independent series of higher-order moment estimations are necessary for achieving accurate estimation. In the Algorithm 4, both estimations of $\boldsymbol{\theta}$ are the solutions to the weighted regression problem in the following form:

$$\underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left(\lambda\|\boldsymbol{\theta}\|_2^2 + \sum_{j=1}^{k-1}\sum_{h=1}^{H} \left(\boldsymbol{\phi}_{j,h,0}^\top \boldsymbol{\theta} - V_{j,h}(s_{h+1}^j)\right)^2 / \bar{\sigma}_{j,h,0}^2\right), \tag{2.6.3}$$

where the regression weight $\bar{\sigma}_{j,h,0}$ is set as Equation (2.6.4).

$$\overline{\sigma}_{k,h,0}^2 \leftarrow \max\left\{\gamma^2 \left\|\boldsymbol{\phi}_{k,h,0}\right\|_{\boldsymbol{\Sigma}_{k,h,0}^{-1}}, \left[\overline{\mathbb{V}}_{k,0} V_{k,h+1}\right](s_h^k, a_h^k) + E_{k,h,0}, \alpha^2\right\}. \tag{2.6.4}$$

$\bar{\sigma}_{j,h,0}$ can be considered as an combination of *aleatoric uncertainty* and *epistemic uncertainty* (Kendall and Gal, 2017; Mai et al., 2022). The first term $\gamma^2 \left\|\boldsymbol{\phi}_{k,h,m}\right\|_{\boldsymbol{\Sigma}_{k,h,0}^{-1}}$ in (2.6.4) is the *epistemic uncertainty* caused by limited available data. And the second term in Equation (2.6.4) is supposed to be the *aleatoric uncertainty* $\mathbb{V}_{k,0}V_{k,h+1}$ characterizing the inherent non-determinism of the transition kernel, which is irreducible. Here the $\mathbb{V}_{k,m}V_{k,h+1}$ is the variance of $V_{k,h+1}$ to 2^m defined as $[\mathbb{P}V_{k,h+1}^{2^{m+1}}](s_h^k, a_h^k) - [\mathbb{P}V_{k,h+1}^{2^m}](s_h^k, a_h^k)^2$. Then, $\left[\mathbb{V}_{k,0}V_{k,h+1}\right](s_h^k, a_h^k)$ is further replaced with its estimate $\left[\overline{\mathbb{V}}_{k,0}V_{k,h+1}\right](s_h^k, a_h^k)$ plus its error bound $E_{k,h,0}$ since real variance $\left[\mathbb{V}_{k,0}V_{k,h+1}\right](s_h^k, a_h^k)$ is unknown to the agent. Because $\left[\mathbb{V}_{k,0}V_{k,h+1}\right](s_h^k, a_h^k)$ is a quadratic function of the real transition kernel parameter $\boldsymbol{\theta}^*$, its estimate can be achieved as

$$\left[\overline{\mathbb{V}}_{k,0}V_{k,h+1}\right]\left(s_h^k, a_h^k\right) = \left[\left\langle \boldsymbol{\phi}_{k,h,1}, \boldsymbol{\theta}_{k,1} \right\rangle\right]_{[0,1]} - \left[\left\langle \widehat{\boldsymbol{\phi}}_{k,h,0}, \boldsymbol{\theta}_{k,0} \right\rangle\right]_{[0,1]}^2,$$

where $\boldsymbol{\theta}_{k,1}$ is again the solution to the weighted regression problem similar to (2.6.4) with predictors $\boldsymbol{\phi}_{k,h,1} = \boldsymbol{\phi}_{V_{k,h+1}^2}(s_h^k, a_h^k)$, responses $V_{k,h+1}^2(s_{h+1}^k)$ and weight $\overline{\sigma}_{k,h,1}$. Following the above idea, the value of weight $\overline{\sigma}_{k,h,1}$ further relies on $\boldsymbol{\theta}_{k,2}$, which is the solution to a weighted regression problem involving another weight $\overline{\sigma}_{k,h,2}$. The algorithm carried out this process recursively until $\overline{\sigma}_{k,h,M-1}$, where its second term is replaced by the trivial upper bound of aleatoric uncertainty.

Applying HOME to the reward-free setting brings additional difficulties in controlling the error of our estimate for the model, as the error introduced by using the pseudo reward function instead of the real reward function and the error introduced by estimating the true transition kernel must be controlled separately. To address this problem, we carefully estimate variables indicating different kinds of error into two series of HOME in Line 10 and Line 11. Since the separation of variables deeply exploits the inner structure of the problem, the two series of HOME can be merged in the end to achieve a unified control for both kinds of error.

Previous work Chen et al. (2021) implemented the weighted value regression in a more crude way. The weights are constructed only on aleatoric uncertainty, totally ignoring epis-

temic uncertainty. In addition, they use the same instead of different transition kernel parameters to calculate different order moments of the value function and stop target value regression at second order moment, which increased avoidable error. As a result, Chen et al. (2021) can only replace factor Hd with factor $H + d$ when trying to improve the dependency on d in the upper bound. In contrast, our work further improves factor $H + d$ to factor H through the well-designed target value regression, as we can see in Corollary 2.6.6.

2.6.1.3 High Confidence Set

At the end of each episode, we add the following constraints into $\mathcal{U}_k$ to update the high confidence set in Line 20 of Algorithm 4.

$$\left\|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{k,m}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}} \leqslant \beta_k,\ m \in \overline{[M]}, \tag{2.6.5}$$

$$\left\|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}_{k,m}\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m}} \leqslant \beta_k,\ m \in \overline{[M]}. \tag{2.6.6}$$

High confidence set $\mathcal{U}_k$ ensures that the estimate $\boldsymbol{\theta}_k$ lies in a neighborhood of real transition kernel parameter $\boldsymbol{\theta}^*$. Here the algorithm adds $2M$ inequalities to constraints in each episode. These inequalities guarantee that estimations of the variance of $\widehat{V}_{k,h}$ and $\widetilde{V}_{k,h}$ up to M-th order are near the real values.

2.6.1.4 Planning Phase

After finishing the exploration, the agent enters the planning phase and receives the real reward function. Depending on optimal Bellman equations, the agent is able to obtain the optimal policy backward from state H to state 1 by dynamic programming based on real reward function r and transition kernel parameter estimate $\boldsymbol{\theta}_K$. And then, the algorithm outputs the optimal policy.

Remark 2.6.2 (Computational Complexity of HF-UCRL-RFE++)**.** Similar with Chen et al. (2021), we assume that the optimization over $\boldsymbol{\theta}$, π, and r in Line 5 of Algorithm 4

can be accomplished with an oracle which is obvious to be called for K times. At each episode k and each stage h, HF-UCRL-RFE++ computes $\{\widehat{\boldsymbol{\phi}}_{k,h,m}\}_{m\in\overline{[M]}}$, $\{\widetilde{\boldsymbol{\phi}}_{k,h,m}\}_{m\in\overline{[M]}}$, $\{\widehat{\sigma}_{k,h,m}\}_{m\in\overline{[M]}}$, $\{\widetilde{\sigma}_{k,h,m}\}_{m\in\overline{[M]}}$, and updates $\{\widehat{\boldsymbol{\Sigma}}_{k,h,m}\}_{m\in\overline{[M]}}$, $\{\widetilde{\boldsymbol{\Sigma}}_{k,h,m}\}_{m\in\overline{[M]}}$. The computation of $\{\widehat{\boldsymbol{\phi}}_{k,h,m}\}_{m\in\overline{[M]}}$ and $\{\widetilde{\boldsymbol{\phi}}_{k,h,m}\}_{m\in\overline{[M]}}$ require $O(\mathcal{O}M)$ times. According to Algorithm 5, calculating $\{\widehat{\sigma}_{k,h,m}\}_{m\in\overline{[M]}}$ and $\{\widetilde{\sigma}_{k,h,m}\}_{m\in\overline{[M]}}$ require $O(Md^2)$ time since the computation of the inner-product an inversion of matrix and a vector needs $O(d^2)$. The updates of $\{\widehat{\boldsymbol{\Sigma}}_{k,h,m}\}_{m\in\overline{[M]}}$ and $\{\widetilde{\boldsymbol{\Sigma}}_{k,h,m}\}_{m\in\overline{[M]}}$ further require $O(Md^2)$ time. Lastly, determining the optimal policy during the planning phase takes $O(H(SAd+\mathcal{O}))$ time. Therefore, the total time complexity of HF-UCRL-RFE++ is $O(KH(\mathcal{O}M+Md^2)+HSAd)$.

2.6.2 Sample Complexity Analysis

We provide the theoretical analysis for HF-UCRL-RFE++ in this section. In order to show the optimality of HF-UCRL-RFE++, we also provide lower bound of sample complexity for all reward-free exploration algorithms.

2.6.2.1 Upper Bound of the Sample Complexity

We first provide the suboptimality upper bound of our algorithm HF-UCRL-RFE++.

Theorem 2.6.3. For Algorithm 4, set $M = \log(7KH)/\log(2)$, $\alpha = H^{-1/2}$, $\gamma = d^{-1/4}$, $\lambda = d/B^2$, $\{\beta_k\}_{k\geqslant 1}$ as $\beta_k = 12\sqrt{d\eta\tau} + 30\tau/\gamma^2 + \sqrt{\lambda}B$, and denote $\beta = \beta_K$, where $\eta = \log(1+kH/(\alpha^2 d\lambda))$ and $\tau = \log(32(\log(\gamma^2/\alpha)+1)k^2H^2/\delta)$. Then, for any $0<\delta<1$, we have with probability at least $1-\delta$, after collecting K episodes of samples, algorithm 4 returns a policy $\widehat{\pi}_r$ satisfying the following sub-optimality bound,

$$V_1^*(s_1;r) - V_1(s_1;\boldsymbol{\theta}^*,\widehat{\pi}_r,r) = \widetilde{O}\left(\frac{d^2}{K} + \frac{d}{\sqrt{K}}\right).$$

The next corollary specifies the sample complexity of our algorithm.

Corollary 2.6.4. Under the same conditions as in Theorem 2.6.3, Algorithm 4 has sample

complexity of

$$m(\varepsilon,\delta') = \frac{16}{\varepsilon^2}\left(64\max\left\{8\beta\sqrt{d\iota},\sqrt{2\zeta}\right\} + 120\beta\sqrt{d\iota H\alpha^2}\right)^2 + \frac{8}{\varepsilon}\left(2752\max\left\{64\beta^2 d\iota, 2\zeta\right\} + 24\zeta + 240d\iota + 240\beta\gamma^2 d\iota + 120\beta d\iota\sqrt{M}\right) \tag{2.6.7}$$

Moreover, setting $\alpha = H^{-1/2}$, $\gamma = d^{-1/4}$, and $\lambda = d/B^2$, we have the reward-free sample complexity bound $m(\varepsilon,\delta') = \widetilde{O}(d^2\varepsilon^{-2})$.

Proof of Corollary 2.6.4. (2.6.7) is derived directly from Theorem 2.6.3 by setting the sub-optimality to ε and solving the K. □

Remark 2.6.5. To the best of our knowledge, Corollary 2.6.4 provides the first horizon-free sample complexity upper bound independent of state space size S and action space size A for reward-free exploration. This result shows that long-horizon planning does not add extra difficulty to reward-free exploration.

Corollary 2.6.6. When re-scaling the assumption $\sum_{h=1}^{H} r_h(s_h,a_h) \leqslant 1$ to $\sum_{h=1}^{H} r_h(s_h,a_h) \leqslant H$, under the same conditions as Theorem 2.6.3, Algorithm 4 has sample complexity of

$$m(\varepsilon,\delta') = \frac{16H^2}{\varepsilon^2}\left(64\max\left\{8\beta\sqrt{d\iota},\sqrt{2\zeta}\right\} + 120\beta\sqrt{d\iota H\alpha^2}\right)^2 + \frac{8H}{\varepsilon}\left(2752\max\left\{64\beta^2 d\iota, 2\zeta\right\} + 24\zeta + 240d\iota + 240\beta\gamma^2 d\iota + 120\beta d\iota\sqrt{M}\right) \tag{2.6.8}$$

Moreover, setting $\alpha = H^{-1/2}$, $\gamma = d^{-1/4}$, and $\lambda = d/B^2$, we have the reward-free sample complexity bound $m(\varepsilon,\delta') = \widetilde{O}(H^2d^2\varepsilon^{-2})$.

Proof of Corollary 2.6.6. (2.6.8) is a direct result of Corollary 2.6.4 by setting $r'_h(s_h,a_h) = r_h(s_h,a_h)/H$. □

Remark 2.6.7. The assumption $\sum_{h=1}^{H} r_h(s_h,a_h) \leqslant H$ covers the standard reward assumption $r_h(s_h,a_h) \in [0,1]$. Therefore, compared with Chen et al. (2021), our analysis does

not require the $d > H$ assumption and achieves the same sample complexity bound up to logarithmic factors except for the trivial $\widetilde{O}(H)$ difference between time-homogeneous and time-inhomogeneous models with a milder assumption. This improvement can be attributed to the refined value target regression technique, high-order moment estimation (HOME), adopted in our approach. We provide a detailed analysis of this improvement in the "High-order Moment Estimation" part in the Section 2.6.1.

2.6.2.2 Lower Bound of the Sample Complexity

The following results provide lower bounds of the sample complexity and suggest that our algorithm is minimax optimal. We will consider the *hard-to-learn linear mixture MDPs* constructed in Zhou and Gu (2022a). The state space is $\mathcal{S} = \{x_1, x_2, x_3\}$ and the action space is $\mathcal{A} = \{\mathbf{a}\} = \{-1, 1\}^{d-1}$. The reward function satisfies $r(x_1, \cdot) = r(x_2, \cdot) = 0$, and $r(x_3, \cdot) = \frac{1}{H}$. The transition probability is defined to be $\mathbb{P}(x_2 \mid x_1, \boldsymbol{a}) = 1 - (\delta + \langle \boldsymbol{\mu}, \boldsymbol{a} \rangle)$ and $\mathbb{P}(x_3 \mid x_1, \boldsymbol{a}) = \delta + \langle \boldsymbol{\mu}, \boldsymbol{a} \rangle$, where $\delta = 1/6$ and $\boldsymbol{\mu} \in \{-\Delta, \Delta\}^{d-1}$ with $\Delta = \sqrt{\delta/K}/(4\sqrt{2})$.

Theorem 2.6.8. Suppose $B > 1$. Then for any algorithm $\texttt{ALG}_{Free}$ solving reward-free linear mixture MDP problems satisfying assumption 2.6.1, there exist a linear mixture MDP $\mathcal{M}$ such that $\texttt{ALG}_{Free}$ needs to collect at least $\Omega\left(d^2\varepsilon^{-2}\right)$ episodes of samples to output an ε-optimal policy with probability at least $1 - \delta$. This lower bound matches the sample com-

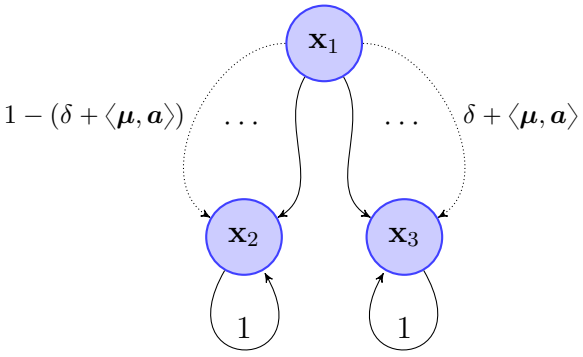


Figure 2.2: The transition kernel of the hard-to-learn linear mixture MDPs.

plexity upper bound provided in Corollary 2.6.4, which shows our upper bound is optimal.

Remark 2.6.9. The lower bound is similar to the lower bound provided in Chen et al. (2021). The first difference is that we rescale the non-zero reward in hard-to-learn cases from 1 to $\frac{1}{H}$ in order to satisfy Assumption 2.6.1. The second difference is that we consider the time-homogeneous model instead of the time-inhomogeneous one in theirs. By these changes, our lower bound for reward-free exploration provided in Theorem 2.6.8 removes the unnecessary polynomial dependency on episode length H introduced by the scale of total reward.

Corollary 2.6.10. Under the same conditions as Theorem 2.6.8 and replacing the bounded total reward $\sum_{h=1}^{H} r_h(s_h, a_h) \leqslant 1$ with $r_h \in [0, 1]$, for any algorithm $\texttt{ALG}_{Free}$ solving reward-free linear mixture MDP problems satisfying assumption 2.6.1, there exist a linear mixture MDP $\mathcal{M}$ such that $\texttt{ALG}_{Free}$ needs to collect at least $\widetilde{\Omega}\left(H^2 d^2 \varepsilon^{-2}\right)$ episodes to output an ε-optimal policy with probability at least $1-\delta$, which suggests that the upper bound presented in Theorem 2.6.3 is optimal.

Proof of Corollary 2.6.10. The result presented in Corollary 2.6.10 is directly obtained by letting $r(x_3, \cdot) = 1$ in the hard case presented in Figure 2.2. □

2.7 Conclusion

We study model-based reward-free exploration for learning the linear mixture MDPs. We propose an algorithm which is guaranteed to efficiently explore the environment with the help of the pseudo reward function. In order to improve the sample complexity of this exploration, we leverage the variance information in reinforcement learning and improve the algorithm using a Bernstein-type concentration inequality.

We also extend the aforementioned algorithm into a bounded total reward setting. In this setting, our algorithm is guaranteed to have horizon-free sample complexity in the exploration

phase to find a near-optimal policy in the planning phase for any given reward function. By providing sample complexity lower bound for reward-free exploration in linear mixture MDPs under our assumptions. We show that the sample complexity of our algorithm matches the lower bound up to logarithmic factors, indicating that our algorithm is optimal.

2.8 Proofs

In this section we present the detailed proof of Theorem 2.4.3, Theorem 2.5.1, Theorem 2.6.3 and Theorem 2.6.8 and corollaries we claimed in this chapter.

2.8.1 Proof of Theorem 2.4.3

We will first introduce a lemma to show that for the planning module Algorithm 1, if it is guaranteed that the estimation $\boldsymbol{\theta}$ is close to the true parameter $\boldsymbol{\theta}^*$, then the estimated value function is optimistic. Also the gap between the optimal value function and the value function of the output policy $\{\pi_h\}_{h=1}^H$ could be controlled by the summation of UCB bonus term.

Lemma 2.8.1. Let $\boldsymbol{\theta}, \boldsymbol{\Sigma}, \beta$ be as defined in Algorithm 1. Suppose there exists some event $\boldsymbol{\xi}$ such that $\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_{\boldsymbol{\Sigma}} \leqslant \beta$ on this event. Then on this event, for all $s \in \mathcal{S}$, $V_1(s) \geqslant V_1^*(s; r)$, where V_1 is the output value function for Algorithm 1. We also have that

$$V_1(s) - V_1^{\pi}(s) \leqslant \mathbb{E}\bigg[\sum_{h=1}^{H} \min\{H, 2\beta \|\boldsymbol{\psi}_{V_{h+1}}(s_h, \pi_h(s_h))\|_{\boldsymbol{\Sigma}^{-1}}\} \Big| s, \pi \bigg],$$

where the policy $\pi = \{\pi_h\}_{h=1}^H$ is generated by the planning module Algorithm 1 and V_h is the value function calculated on Line 5 in Algorithm 1.

Next we will give the lemmas on how to guarantee the condition of Lemma 2.8.1 and how to utilize the result of that lemma to control the final policy error $V_1^*(s_1; r) - V_1^{\pi}(s_1; r)$ where the policy π is output of the planning phase. We start with Algorithm 2, which uses the Hoeffding bonus.

Firstly, the next lemma shows how to guarantee the condition in Lemma 2.8.1.

Lemma 2.8.2 (Confidence interval, Hoeffding)**.** For Algorithm 2, let λ, β be as defined in Theorem 2.4.3, then with probability at least $1-\delta/3$, $\|\boldsymbol{\theta}^* - \boldsymbol{\theta}_k\|_{\boldsymbol{\Sigma}_{1,k}} \leqslant \beta$ for any $k \in [K+1]$.

Secondly, based on the lemma above, we find that the policy error during the planning phase is controlled by a summation of the UCB terms. Since from the intuition, the exploration driven reward function (2.4.2) is the UCB term divided by H, the policy error during the planning phase can be converted to the value function V_1^k in the exploration phase. The next lemma shows that the summation of V_1^k over K iterations is sub-linear to K, thus the policy error during the planning phase should be small.

Lemma 2.8.3 (Summation, Hoeffding)**.** Set the parameters of Algorithm 2 as that of Theorem 2.4.3. If the condition in Lemma 2.8.2 holds, then with probability at least $1-\delta/3$, the summation of the value function $V_1^k(s_1^k)$ during the exploration phase is controlled by

$$\sum_{k=1}^{K} V_1^k(s_1^k) \leqslant 8\beta\sqrt{HKd\log(1+KH^3B^2/d)} + 8\beta Hd\log(1+KH^3B^2) + 2H\sqrt{2HK\log(1/\delta)}.$$

Equipped with these lemmas, we are about to prove Theorem 2.4.3.

Proof of Theorem 2.4.3. In the following proof, we condition on the events in Lemma 2.8.2 and Lemma 2.8.3 which holds with probability at least $1-2\delta/3$ by taking the union bound. Applying Lemma 2.8.1 to the final planning phase, we have

$$V_1^*(s;r) - V_1^{\pi}(s;r) \leqslant V_1(s;r) - V_1^{\pi}(s;r) \leqslant \underbrace{\mathbb{E}\Bigg[\sum_{h=1}^{H} \min\{H, 2\beta\|\boldsymbol{\psi}_{V_{h+1}}(s_h,\pi_h(s_h))\|_{\boldsymbol{\Sigma}_{1,K+1}^{-1}}\}\Bigg]}_{I_1}, \tag{2.8.1}$$

where the expectation is taken condition on initial state s and policy π generated by the planning phase. Since $\boldsymbol{\Sigma}_{1,k} \preceq \boldsymbol{\Sigma}_{1,K+1}$ for all $k \in [K]$, we can guarantee that

$$\|\boldsymbol{\psi}_{V_{h+1}}(s_h,\pi_h(s_h))\|_{\boldsymbol{\Sigma}_{1,K+1}^{-1}} \leqslant \|\boldsymbol{\psi}_{V_{h+1}}(s_h,\pi_h(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}.$$

Recall the exploration driven reward function is defined by

$$r_h^k(s,a) = \min\left\{1, \frac{2\beta}{H}\sqrt{\max_{f\in\mathcal{S}\mapsto[0,H-h]} \|\boldsymbol{\psi}_f(s,a)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}}\right\}, \quad (2.8.3) \tag{2.8.2}$$

one can easily verify that $\min\{H, 2\beta\|\boldsymbol{\psi}_{V_{h+1}}(s_h,\pi_h(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\} \leqslant Hr_h^k(s_h,\pi_h(s_h))$. Therefore for any $k \in [K]$ episode, we can bound the term I_1 using the value function $V_1^\pi(s;\{r_h^k\}_{h=1}^H)$ of the output policy π in the planning phase given the $\{r_h^k\}_{h=1}^H$ as the reward function, i.e.

$$I_1 \leqslant \mathbb{E}\bigg[\sum_{h=1}^H Hr_h^k(s_h,\pi_h(s_h))\bigg] = HV_1^\pi(s;\{r_h^k\}_{h=1}^k). \tag{2.8.3}$$

Plugging the bound of I_1 back into (2.8.1) then taking the expectation over the initial state distribution μ, we have for any $k \in [K]$,

$$\begin{aligned}\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] &\leqslant H\mathbb{E}_{s\sim\mu}[V_1^\pi(s;\{r_h^k\}_{h=1}^k)] \\ &= H\Big(\mathbb{E}_{s\sim\mu}[V_1^\pi(s;\{r_h^k\}_{h=1}^k)] - V_1^\pi(s_1^k;\{r_h^k\}_{h=1}^k)\Big) \\ &\quad + HV_1^\pi(s_1^k;\{r_h^k\}_{h=1}^k).\end{aligned}$$

Hence

$$\begin{aligned}\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] \leqslant \frac{H}{K}\sum_{k=1}^K \Big(&\mathbb{E}_{s\sim\mu}[V_1^\pi(s;\{r_h^k\}_{h=1}^k)] - V_1^\pi(s_1^k;\{r_h^k\}_{h=1}^k) \\ &+ V_1^\pi(s_1^k;\{r_h^k\}_{h=1}^k)\Big).\end{aligned} \tag{2.8.4}$$

Since $V_1^\pi(s;\{r_h^k\}_{h=1}^k) \leqslant H$ for all $k \in [K], s \in \mathcal{S}$, by Azuma-Hoeffding's inequality, with probability at least $1-\delta/3$,

$$\sum_{k=1}^K \Big(\mathbb{E}_{s\sim\mu}[V_1^\pi(s;\{r_h^k\}_{h=1}^k)] - V_1^\pi(s_1^k;\{r_h^k\}_{h=1}^k)\Big) \leqslant H\sqrt{2K\log(3/\delta)}. \tag{2.8.5}$$

By plugging (2.8.5) into (2.8.4), we have

$$\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] \leqslant \frac{H}{K}\sum_{k=1}^K V_1^\pi(s_1^k;\{r_h^k\}_{h=1}^k) + H^2\sqrt{2\log(3/\delta)/K}.$$

Applying Lemma 2.8.1 to the exploration phase, for any k-th episode, $V_1^{\pi}(s_1^k; \{r_h^k\}_{h=1}^k) \leqslant V_1^*(s_1^k; \{r_h^k\}_{h=1}^k) \leqslant V_1^k(s_1^k)$, thus replacing the value function V_1^{π} with the estimated value function V_1^k, we have

$$\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^{\pi}(s;r)] \leqslant \frac{H}{K}\sum_{k=1}^{K} V_1^k(s_1^k) + H^2\sqrt{2\log(3/\delta)/K}. \tag{2.8.6}$$

Finally by Lemma 2.8.3 we can bound the summation over V_1^k, hence

$$\begin{aligned}\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^{\pi}(s;r)] \leqslant H^2\sqrt{2\log(3/\delta)/K} + 8\beta\sqrt{H^3 d\log(1+KH^3B^2/d)/K} \\ + 8\beta d H^2\log(1+KH^3B^2)/K + 2H^2\sqrt{2H\log(1/\delta)/K}\end{aligned}$$

and by taking union bound, the result holds with probability at least $1-\delta$. Recall the setting of $\beta \sim \widetilde{\mathcal{O}}(H\sqrt{d})$ as in Theorem 2.4.3, let $K = \widetilde{\mathcal{O}}(H^5d^2\epsilon^{-2})$, the policy error $\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^{\pi}(s;r)]$ is bounded by ϵ. □

2.8.1.1 Proof of Corollary 2.4.5

Proof of Corollary 2.4.5. Following the proof of Theorem 2.4.3, since for all $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{x}\|_1 \leqslant \|\mathbf{x}\|_2 \leqslant \sqrt{d}\|\mathbf{x}\|_1$ it follows that

$$\begin{aligned}\|\boldsymbol{\psi}_{V_{h+1}}(s_h, \pi_h(s_h))\|_{\boldsymbol{\Sigma}_{1,K+1}^{-1}} &= \|\boldsymbol{\Sigma}_{1,K+1}^{-1/2}\boldsymbol{\psi}_{V_{h+1}}(s_h, \pi_h(s_h))\|_2 \\ &\leqslant \sqrt{d}\|\boldsymbol{\Sigma}_{1,K+1}^{-1/2}\boldsymbol{\psi}_{V_{h+1}}(s_h, \pi_h(s_h))\|_1. \end{aligned}\tag{2.8.7}$$

We denote $\widetilde{u}_h^k$ as the result using the ℓ_1 norm as the surrogate objective function in this optimization problem (2.4.5), i.e.

$$\widetilde{u}_h^k := \operatorname*{argmax}_{f\in\mathcal{S}\mapsto[0,H-h]} \|\boldsymbol{\Sigma}_{1,k}^{-1/2}\boldsymbol{\psi}_f(s_h^k, a_h^k)\|_1,$$

then (2.8.7) yields

$$\begin{aligned}\|\boldsymbol{\psi}_{V_{h+1}}(s_h,\pi_h(s_h))\|_{\boldsymbol{\Sigma}^{-1}_{1,K+1}} &\leqslant \sqrt{d}\|\boldsymbol{\Sigma}^{-1/2}_{1,K+1}\boldsymbol{\psi}_{V_{h+1}}(s_h,\pi_h(s_h))\|_1\\ &\leqslant \sqrt{d}\|\boldsymbol{\Sigma}^{-1/2}_{1,K+1}\boldsymbol{\psi}_{\widetilde{u}^k_h}(s_h,\pi_h(s_h))\|_1\\ &\leqslant \sqrt{d}\|\boldsymbol{\Sigma}^{-1/2}_{1,K+1}\boldsymbol{\psi}_{\widetilde{u}^k_h}(s_h,\pi_h(s_h))\|_2\\ &\leqslant \sqrt{d}\|\boldsymbol{\Sigma}^{-1/2}_{1,K+1}\boldsymbol{\psi}_{u^k_h}(s_h,\pi_h(s_h))\|_2,\end{aligned}$$

where the second inequality comes from $\widetilde{u}^k_h$ is the solution in (2.4.5), the third inequality comes from the fact that $\|\mathbf{x}\|_1 \leqslant \|\mathbf{x}\|_2$ and the forth inequality comes from the definition that u^k_h. Then (2.8.3) is changed to be

$$I_1 \leqslant H\sqrt{d}V^\pi_1(s,\{r^k_h\}^k_{h=1}).$$

Noticing that comparing to the original result, there's an additional $\sqrt{d}$ factor which yields (2.8.7)

$$\mathbb{E}_{s\sim\mu}[V^*_1(s;r) - V^\pi_1(s;r)] \leqslant \frac{H\sqrt{d}}{K}\sum_{k=1}^K V^k_1(s^k_1) + H^2\sqrt{2d\log(3/\delta)/K}.$$

Then it is easy to show that using ℓ_1 as the surrogate objective function, the sample complexity of Algorithm 2 turns out to be $\widetilde{\mathcal{O}}(H^5d^3\epsilon^{-2})$ □

2.8.2 Proof of Theorem 2.5.1

We are going to analyze Algorithm 3 and provide the proof of Theorem 2.5.1. Following the proof of Theorem 2.4.3, we only need to revise Lemmas 2.8.2 and 2.8.3 to continue the proof of Theorem 2.5.1.

Lemma 2.8.4 (Confidence interval, Bernstein)**.** Let $\beta, \widehat{\beta}, \widetilde{\beta}, \check{\beta}$ and λ be defined as Theorem 2.5.1, then with probability at least $1-\delta/3$, for all $k \in [K+1]$,

$$\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widehat{\beta},\ \|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \check{\beta},\ \|\boldsymbol{\theta}^* - \widetilde{\boldsymbol{\theta}}_k\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widetilde{\beta},\ \|\boldsymbol{\theta}^* - \boldsymbol{\theta}_{K+1}\|_{\boldsymbol{\Sigma}_{1,K+1}} \leqslant \beta, \quad (2.8.8)$$

and $|[\mathbb{V}_h V^k_{h+1}](s,a) - \bar{\mathbb{V}}^k_h(s,a)| \leqslant E^k_h(s,a)$.

Lemma 2.8.5 (Summation, Bernstein)**.** For Algorithm 2, setting its parameters as in Lemma 2.8.2, with probability at least $1 - \delta/3$, the summation of the value function during exploration phase is controlled by

$$\sum_{k=1}^{K} V_1^k(s_1^k) \leqslant \widetilde{\mathcal{O}}(\sqrt{H^3Kd} + Hd\sqrt{K}) + o(\sqrt{K}).$$

Proof of Theorem 2.5.1. The proof is almost the same as the proof of Theorem 2.4.3 by replacing Lemma 2.8.2 with Lemma 2.8.4, Lemma 2.8.3 with Lemma 2.8.5. In detail, following the same method, (2.8.6) works for Algorithm 3 under the condition in Lemma 2.8.4 holds. Therefore, by using Lemma 2.8.5 instead of Lemma 2.8.3, with probability at least $1 - \delta$,

$$\begin{aligned}\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] &\leqslant \frac{H}{K}\sum_{k=1}^{K} V_1^k(s_1^k) + H^2\sqrt{2\log(3/\delta)/K} \\ &\leqslant \widetilde{\mathcal{O}}\Big((\sqrt{H^4d^2} + \sqrt{H^5d})/\sqrt{K}\Big).\end{aligned}$$

Letting $K = \widetilde{\mathcal{O}}(H^4d(H+d)\epsilon^{-2})$, the policy error for the planning phase could be controlled by $\mathbb{E}_{s\sim\mu}[V_1^*(s;r) - V_1^\pi(s;r)] \leqslant \epsilon$. □

2.8.2.1 Proof of Corollary 2.5.3

Proof of Corollary 2.5.3. The proof is almost the same as proof of Corollary 2.4.5, by adding the additional dependency d into the regret bound achieved by Theorem 2.5.1, it's easy to verify that the sample complexity using the ℓ_1 norm as the surrogate function (2.4.5) is $\widetilde{\mathcal{O}}(H^4d^2(H+d)\epsilon^{-2})$. □

2.8.3 Proof of Theorem 2.6.3

We first define the good event such that the high-order estimator is well-bounded.

Lemma 2.8.6. For all $0 < \delta < 1$, suppose β_k is set as in Theorem 2.6.3, the following event

happens with probability at least $1 - 2M\delta$

$$\left\|\widehat{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}} \leqslant \beta_k \tag{2.8.9}$$

$$\left\|\widetilde{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m}} \leqslant \beta_k \tag{2.8.10}$$

$$\left\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}} \leqslant 2\beta_k \tag{2.8.11}$$

$$\left\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}} \leqslant 2\beta_k. \tag{2.8.12}$$

We define the event that Lemma 2.8.6 holds to be $\mathcal{E}_{2.8.6}$. Then the following lemma controls the suboptimality gap between optimal value functions and our estimated value function in the planning phase with the uncertainty along trajectories.

Lemma 2.8.7. Under event $\mathcal{E}_{2.8.6}$, for any reward function r in the planning phase, the suboptimality gap of the outputted policy $\widehat{\pi}_r$ can be bounded as

$$V_1^*\left(s_1; r\right) - V_1\left(s_1; \boldsymbol{\theta}^*, \widehat{\pi}_r, r\right) \leqslant 4\widehat{V}_{K,1}\left(s_1\right). \tag{2.8.13}$$

The next lemma shows that the uncertainty along trajectories decreases with respect to episodes. This lemma is intuitively right since the uncertainty should decrease with more information collected.

Lemma 2.8.8. Under event $\mathcal{E}_{2.8.6}$, for uncertainty along trajectories, we have

$$\widehat{V}_{K,1}(s_1; \boldsymbol{\theta}_K, \pi_K, r_K) \leqslant \frac{1}{K}\bigg(\sum_{k=1}^{K} \widehat{V}_{k,1}(s_1; \boldsymbol{\theta}_k, \pi_k, r_k)\bigg).$$

The last lemma upper bounds the sum of the uncertainty along trajectories.

Lemma 2.8.9. For any $0 < \delta < 1$, with probability at least $1 - 4M\delta$, we have

$$\sum_{k=1}^{K} \widehat{V}_{k,1}(s_1; \boldsymbol{\theta}_k, \widehat{\pi}_k, r_k) = \widetilde{O}(d\sqrt{K} + d^2). \tag{2.8.14}$$

Equipped with the above lemmas, we are ready to prove Theorem 2.6.3.

Proof of Theorem 2.6.3. The following proof is conditioned on $\mathcal{E}_{2.8.6} \cap \mathcal{E}_{2.8.27}$, which holds with probability at least $1 - 4M\delta = 1 - \delta'$. We have

$$
\begin{aligned}
&V_1^*(s_1; r) - V_1(s_1; \boldsymbol{\theta}^*, \widehat{\pi}_r, r) \\
&\leqslant 4\widehat{V}_{K,1}\left(s_1; \boldsymbol{\theta}_K, \widehat{\pi}_K, r_K\right) \\
&\leqslant \frac{4}{K} \sum_{k=1}^{K} V_{k,1}(s_1; \boldsymbol{\theta}_k, \widehat{\pi}_k, r_k) \\
&\leqslant \frac{4}{K}\left(896 \max\left\{64\beta^2 d\iota, 2\zeta\right\} + 24\zeta + 240 d\iota + 240\beta\gamma^2 d\iota + 120\beta d\iota\sqrt{M} + 24\sqrt{\zeta M d\iota} + M d\iota\right) \\
&\quad + \frac{4}{\sqrt{K}}\left(64 \max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\} + 120\beta\sqrt{d\iota H\alpha^2}\right),
\end{aligned}
$$

where the first inequality holds due to Lemma 2.8.7, the second inequality holds due to Lemma 2.8.8, and the third equality holds due to Lemma 2.8.9. □

2.8.4 Proof of Theorem 2.6.8

Reward-free exploration is more difficult than non-reward-free MDP by definitions since we can easily solve non-reward-free MDP by ignoring its reward and executing reward-free exploration. Thus, we will start with acquiring lower bounds under non-reward-free MDP settings and then obtain sample complexity lower bounds of reward-free exploration. The proof follows ideas of Zhou and Gu (2022a) and Chen et al. (2021).

As noted in Section 2.6.2.2, we will consider the *hard-to-learn linear mixture MDPs* constructed in Zhou and Gu (2022a). The state space $\mathcal{S} = \{x_1, x_2, x_3\}$ and the action space $\mathcal{A} = \{\mathbf{a}\} = \{-1, 1\}^{d-1}$. The reward function satisfies $r(x_1, \cdot) = r(x_2, \cdot) = 0$, and $r(x_3, \cdot) = \frac{1}{H}$. The transition probability satisfies $\mathbb{P}(x_2 \mid x_1, \boldsymbol{a}) = 1 - (\delta + \langle \boldsymbol{\mu}, \boldsymbol{a} \rangle)$ and $\mathbb{P}(x_3 \mid x_1, \boldsymbol{a}) = \delta + \langle \boldsymbol{\mu}, \boldsymbol{a} \rangle$, where $\delta = 1/6$ and $\boldsymbol{\mu} \in \{-\Delta, \Delta\}^{d-1}$ with $\Delta = \sqrt{\delta/K'}/(4\sqrt{2})$. The

transition kernel is formulated as

$$\boldsymbol{\phi}\left(s' \mid s, \boldsymbol{a}\right) = \begin{cases} \left(\alpha(1-\delta), -\beta \boldsymbol{a}^{\top}\right)^{\top}, & s = x_1, s' = x_2; \\ \left(\alpha\delta, \beta \boldsymbol{a}^{\top}\right)^{\top}, & s = x_1, s' = x_3; \\ \left(\alpha, \mathbf{0}^{\top}\right)^{\top}, & s \in \left\{x_2, x_3\right\}, s' = s; \\ \mathbf{0}, & \text{otherwise.} \end{cases} \qquad \boldsymbol{\theta} = \left(1/\alpha, \boldsymbol{\mu}^{\top}/\beta\right)^{\top}$$

The following lemma from Zhou and Gu (2022a) lower bounds the regret for linear mixture MDP.

Lemma 2.8.10 (Theorem 5.4, Zhou and Gu (2022a))**.** Let $B > 1$. Then for any algorithm, when $K' \geqslant \max\left\{3d^2, (d-1)/(192(B-1))\right\}$, there exists a B-bounded linear mixture MDP satisfying Assumptions 3.2 such that its expected regret $\mathbb{E}[\text{Regret}(K')]$ is lower bounded by $\Omega\left(d\sqrt{K'}/(16\sqrt{3})\right)$.

Given Lemma 2.8.10, we will use the regret lower bound of non-reward-free linear mixture MDPs to derive the sample coomplexity lower bound.

Lemma 2.8.11. Suppose $B > 1$. Then for any algorithm $\texttt{ALG}_{NonFree}$ solving non-reward-free linear mixture MDP problems satisfying assumption 2.6.1, there exist a linear mixture $\mathcal{M}$ such that $\texttt{ALG}_{NonFree}$ needs to collect at least $\frac{Cd^2}{\varepsilon^2}$ episodes to output an ε-policy with probability at least $1-\delta$. Here C is an absolute constant.

Proof of Lemma 2.8.11. For any algorithm $\texttt{ALG}_{NonFree}$, we construct an algorithm $\texttt{ALG}'_{NonFree}$ executing totally $K_1 = cK$ episodes, where c is a constant integer larger than 1. The first K episodes of $\texttt{ALG}'_{NonFree}$ are the same as $\texttt{ALG}_{NonFree}$, and the rest episodes keep executing the policy at the end of episode K. By Lemma 2.8.10, we have

$$\sum_{k=1}^{K_1} \mathbb{E}\left[V(s_1; \boldsymbol{\theta}^*, \pi^*, r) - V(s_1; \boldsymbol{\theta}^*, \pi_k, r)\right] \geqslant \frac{c' d\sqrt{K_1}}{16\sqrt{3}}, \tag{2.8.15}$$

for some constant c'. In addition, based on the construction of *the hard-to-learn MDPs*, where $K^{'} = K_1$, the per-episode regret is upper bounded by

$$\mathbb{E}\left[V(s_1;\boldsymbol{\theta}^*,\pi^*,r) - V(s_1;\boldsymbol{\theta}^*,\pi_k,r)\right] \leqslant \frac{d}{4\sqrt{3K_1}}. \tag{2.8.16}$$

Thus, calculating (2.8.15) - $(K_1 - K)\times$ (2.8.16), and choosing$c = \max\left\{5/c', 2\right\}$, we have

$$\sum_{k=K+1}^{K_1} \mathbb{E}\left[V(s_1;\boldsymbol{\theta}^*,\pi^*,r) - V(s_1;\boldsymbol{\theta}^*,\pi_k,r)\right] \geqslant \frac{d\sqrt{K}}{16\sqrt{3c}}.$$

Since the policies in episode $K+1$ to episode K_1 are same to π_K, we have

$$\mathbb{E}\left[V(s_1;\boldsymbol{\theta}^*,\pi^*,r) - V(s_1;\boldsymbol{\theta}^*,\pi_K,r)\right] \geqslant \frac{d}{16\sqrt{3cK}c}.$$

Suppose the $\texttt{ALG}_{NonFree}$ return return a ε-optimal policy with probability $1-\delta$. Then,

$$(1-\delta)\varepsilon + \delta\frac{d}{4\sqrt{3cK}} \geqslant \frac{d}{16\sqrt{3cK}c}.$$

Setting $\delta < \min\{1, 1/(4c)\}$, by solving the inequality, we have $K \geqslant \frac{Cd^2}{\varepsilon^2}$ for some constant C. □

Since reward-free MDP is more difficult than non-reward-free MDP, Lemma 2.8.11 directly indicates Theorem 2.6.8.

Proof of Theorem 2.6.8. We will prove the theorem by contradiction. Assume all reward-free linear mixture MDPs can be solved with sample complexity of $o(\frac{d^2}{\varepsilon^2})$. Then, for any non-reward-free MDP $\mathcal{M}$, there exists an algorithm $\texttt{ALG}'$ (ε, δ) learning its reward-free counterpart $\mathcal{M}'$ with sample complexity of $o(\frac{d^2}{\varepsilon^2})$. We define $\texttt{ALG}$ solving $\mathcal{M}$ as follows: it collects K episodes of data and outputs the policy in the same way as $\texttt{ALG}'$ by ignoring the rewards. Then $\texttt{ALG}$ can also (ε, δ) learning $\mathcal{M}$ with sample complexity of $o(\frac{d^2}{\varepsilon^2})$, which contradicts Theorem 2.8.11. □

Corollary 2.6.10 can be viewed as an direct result of Theorem 2.6.8.

2.8.5 Proofs in Section 2.8.1 and Section 2.8.2

2.8.5.1 Filtration

For the simplicity of further proof, we define the event filtration here as

$$\mathcal{G}_{h,k} = \big\{\{s_i^\kappa, a_i^\kappa\}_{i=1,\kappa=1}^{H,k-1}, \{s_i^k, a_i^k\}_{i=1}^{h-1}\big\},$$

it is easy to verify that s_h^k is $\mathcal{G}_{h+1,k}$-measurable. Also, since π^k is $\mathcal{G}_{h,k}$-measurable for all $h \in [H]$, $a_h^k = \pi_h^k(s_h^k)$ is also $\mathcal{G}_{h+1,k}$-measurable. Also, for any function $f \leqslant R$ built on $\mathcal{G}_{h+1,k}$, such as $V_{h+1}^k, u_h^k,\ f(s_{h+1}^k) - [\mathbb{P}f](s_h^k, a_h^k)$ is $\mathcal{G}_{h+1,k}$-measurable and it is also a zero-mean R-sub-Gaussian conditioned on $\mathcal{G}_{h+1,k}$.

Since $\mathcal{G}_{H+1,k} = \mathcal{G}_{1,k+1}$, we could arrange the filtration as

$$\mathcal{G} = \{\mathcal{G}_{1,1}, \cdots, \mathcal{G}_{H,1}, \cdots, \mathcal{G}_{1,k}, \cdots, \mathcal{G}_{h,k}, \cdots \mathcal{G}_{H,k}, \cdots, \mathcal{G}_{1,k+1}, \cdots, \mathcal{G}_{H,K}, \mathcal{G}_{1,K+1}\},$$

and we will use $\mathcal{G}$ as the filtration set for all of the proofs in the following section and it is obvious that $\mathcal{G}_{1,K+1}$ contains all information we collect during the exploration phase.

2.8.5.2 Proof of Lemma 2.8.1

Proof of Lemma 2.8.1. We prove this lemma by induction on time step h. Indeed, when $h = H+1$, $V_{H+1}(s) = V_{H+1}^*(s; r) = 0$ by definition. Suppose for $h \in [H]$, $V_{h+1}(s) \geqslant V_{h+1}^*(s; r)$, then following the update rule of Q function in Algorithm 1, we have

$$\begin{aligned}
&Q_h(s,a) - Q_h^*(s,a;r) \\
&\quad = \min\big\{H, r_h(s,a) + \langle \boldsymbol{\psi}_{V_{h+1}}(s,a), \boldsymbol{\theta}\rangle + \beta\|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}}\big\} - r_h(s,a) - [\mathbb{P}V_{h+1}^*](s,a;r) \\
&\quad \geqslant \min\big\{H - Q_h^*(s,a;r), \langle \boldsymbol{\psi}_{V_{h+1}}(s,a), \boldsymbol{\theta}\rangle + \beta\|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}} - [\mathbb{P}V_{h+1}^*](s,a;r)\big\}.
\end{aligned}$$

We need to show that $Q_h(s,a) \geqslant Q_h^*(s,a;r)$. Since it is obvious that the first term $H - Q_h^*(s,a;r)$ in min operator is greater than zero, we only need to verify that the second term

is also positive where

$$\begin{aligned}
&\langle \boldsymbol{\psi}_{V_{h+1}}(s,a), \boldsymbol{\theta} \rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}} - [\mathbb{P}V^*_{h+1}](s,a;r) \\
&\quad \geqslant \langle \boldsymbol{\psi}_{V_{h+1}}(s,a), \boldsymbol{\theta} \rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}} - [\mathbb{P}V_{h+1}](s,a;r) \\
&\quad = \langle \boldsymbol{\psi}_{V_{h+1}}(s,a), \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}} \\
&\quad \geqslant \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}} - \|\boldsymbol{\psi}_{V_{h+1}}(s,a)\|_{\boldsymbol{\Sigma}^{-1}} \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\boldsymbol{\Sigma}},
\end{aligned}$$

where the first inequality is from the induction assumption that $V^*_{h+1}(s;r) \leqslant V_{h+1}(s)$. The second equality is from the expectation of value function is a linear function of $\boldsymbol{\psi}_{V_{h+1}}$ shown in (2.3.2). Then the inequality on the third line is utilizing the fact that $\langle \mathbf{x}, \mathbf{y} \rangle \geqslant -\|\mathbf{x}\|_{\mathbf{A}^{-1}} \|\mathbf{y}\|_{\mathbf{A}}$. Since it is guaranteed that $\beta \geqslant \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\boldsymbol{\Sigma}}$ from the statement of this lemma, $Q_h(s,a) - Q^*_h(s,a;r) \geqslant 0$, which from induction we get our conclusion.

For the second part controlling $V_1(s) - V^\pi_1(s)$, since aforementioned proof has shown that $V^*_h(s;r) \leqslant V_h(s)$ for all $h \in [H]$, we have $V^*_h(s;r) - V^\pi_h(s;r) \leqslant V_h(s) - V^\pi_h(s;r)$ and

$$\begin{aligned}
V_h(s) - V^\pi_h(s;r) &= \min\{H, r_h(s,\pi_h(s)) + \langle \boldsymbol{\psi}_{V_{h+1}}, \boldsymbol{\theta} \rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,\pi_h(s))\|_{\boldsymbol{\Sigma}^{-1}}\} \\
&\quad - r_h(s,\pi_h(s)) - [\mathbb{P}V^\pi_{h+1}](s,\pi_h(s);r) \\
&\leqslant \min\{H, \langle \boldsymbol{\psi}_{V_{h+1}}, \boldsymbol{\theta} \rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,\pi_h(s))\|_{\boldsymbol{\Sigma}^{-1}} - [\mathbb{P}V_{h+1}](s,\pi_h(s))\} \\
&\quad + [\mathbb{P}V_{h+1}](s,\pi_h(s))\} - [\mathbb{P}V^\pi_{h+1}](s,\pi_h;r) \\
&= \min\{H, \langle \boldsymbol{\psi}_{V_{h+1}}, \boldsymbol{\theta} - \boldsymbol{\theta}^* \rangle + \beta \|\boldsymbol{\psi}_{V_{h+1}}(s,\pi_h(s))\|_{\boldsymbol{\Sigma}^{-1}}\} \\
&\quad + [\mathbb{P}V_{h+1}](s,\pi_h(s))\} - [\mathbb{P}V^\pi_{h+1}](s,\pi_h(s);r) \\
&\leqslant \min\{H, 2\beta \|\boldsymbol{\psi}_{V_{h+1}}(s,\pi_h(s))\|_{\boldsymbol{\Sigma}^{-1}}\} \\
&\quad + [\mathbb{P}V_{h+1}](s,\pi_h(s))\} - [\mathbb{P}V^\pi_{h+1}](s,\pi_h(s);r),
\end{aligned}$$

where the first inequality is directly from moving term $-r_h(s,\pi_h(s)) - [\mathbb{P}V_{h+1}](s,\pi_h(s))$ into the min operator, the second inequality uses the condition that $\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_{\boldsymbol{\Sigma}} \leqslant \beta$ and

$\langle \mathbf{x}, \mathbf{y} \rangle \leqslant \|\mathbf{x}\|_{\mathbf{A}^{-1}} \|\mathbf{y}\|_{\mathbf{A}}$. Considering the first step $h = 1$, we have

$$\begin{aligned}
V_1(s_1) - V_1^{\pi}(s_1; r) &\leqslant \min\{H, 2\beta \|\boldsymbol{\psi}_{V_2}(s_1, \pi_1(s_1))\|_{\boldsymbol{\Sigma}^{-1}}\} + \mathbb{E}_{s_2 \sim \mathbb{P}(\cdot|s_1, \pi_1(s_1))}[V_2(s_2) - V_2^{\pi}(s_2)] \\
&\leqslant \min\{H, 2\beta \|\boldsymbol{\psi}_{V_2}(s_1, \pi_1(s_1))\|_{\boldsymbol{\Sigma}^{-1}}\} \\
&\quad + \mathbb{E}_{s_2 \sim \mathbb{P}(\cdot|s_1, \pi_1(s_1))} \Big[\min\{H, 2\beta \|\boldsymbol{\psi}_{V_3}(s_2, \pi_2(s_2))\|_{\boldsymbol{\Sigma}^{-1}}\} \\
&\quad + \mathbb{E}_{s_3 \sim \mathbb{P}(\cdot|s_2, \pi_2(s_2))}[V_3(s_3) - V_3^{\pi}(s_3)] \Big] \\
&\leqslant \cdots \\
&\leqslant \mathbb{E}\Bigg[\sum_{h=1}^{H} \min\{H, 2\beta \|\boldsymbol{\psi}_{V_{h+1}}(s_h, \pi_h(s_h))\|_{\boldsymbol{\Sigma}^{-1}}\} \Bigg| s_1, \pi \Bigg],
\end{aligned}$$

which concludes our proof. □

2.8.5.3 Proof of Lemma 2.8.2

We introduce the classical confidence set lemma from (Abbasi-Yadkori et al., 2011).

Lemma 2.8.12 (Theorem 2, Abbasi-Yadkori et al. (2011))**.** Let $\{\mathcal{F}_t\}_{t=0}^{\infty}$ be a filtration and $\{\eta_t\}$ is a real-valued stochastic process which is F_t-measurable and conditionally R-sub-Gaussian. Set $y_t = \langle \mathbf{x}_t, \boldsymbol{\psi}^* \rangle + \eta_t$, $\mathbf{V}_t = \lambda \mathbf{I} + \sum_{i=1}^{t} \mathbf{x}_i \mathbf{x}_i^{\top}$ where $\mathbf{x} \in \mathbb{R}^d$. Denote the estimation of $\boldsymbol{\psi}^*$ as $\boldsymbol{\psi}_t = \mathbf{V}_t^{-1} \sum_{i=1}^{t} y_i \mathbf{x}_i$. If $\|\boldsymbol{\psi}^*\|_2 \leqslant S, \|\mathbf{x}_t\|_2 \leqslant L$, then with probability at least $1 - \delta$, for all $t \geqslant 0$

$$\|\boldsymbol{\psi}^* - \boldsymbol{\psi}_t\|_{\mathbf{V}_t} \leqslant R \sqrt{d \log\left(\frac{1 + tL^2/\lambda}{\delta} \right)} + S\sqrt{\lambda}.$$

Equipped with this lemma, we begin our proof.

Proof of Lemma 2.8.2. Since $[\mathbb{P}u_h^k](s_h^k, a_h^k) = \langle \boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k), \boldsymbol{\theta}^* \rangle$ due to (2.3.2) and $u_h^k(s) \leqslant H$, $u_h^k(s) - \langle \boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k), \boldsymbol{\theta}^* \rangle$ is $\mathcal{G}_{h,k}$-measurable and it is also a zero mean H-sub-Gaussian random variable conditioned on $\mathcal{G}_{h,k}$. Also from Definition 2.3.1, $\|\boldsymbol{\theta}^*\|_2 \leqslant B, \|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_2 \leqslant H$. Therefore, recall the calculation of $\boldsymbol{\theta}_k$, according to Lemma 2.8.12, let $t = (k-1)H$ we have

$$\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|_{\boldsymbol{\Sigma}_{1,k}} \leqslant H \sqrt{d \log\left(\frac{1 + (k-1)H^3/\lambda}{\delta} \right)} + B\sqrt{\lambda}.$$

Let $\lambda = B^{-2}$, $\delta = \delta/3$ and relax k with $k = K+1$, we can get the β claimed in Theorem 2.4.3.

□

2.8.5.4 Proof of Lemma 2.8.3

We provide the proof to control the summation of the value function during the exploration phase. To start with, since rather than immediately updating the parameter after each time step, we can only update the estimation $\boldsymbol{\theta}$ and its 'covariance matrix' $\boldsymbol{\Sigma}$ once after each episode. As a result, this 'batched update rule' make the UCB bonus term at step (h, k) be $\|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\mathbf{U}_{1,k}^{-1}}$ instead of $\|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\mathbf{U}_{h,k}^{-1}}$ in the vanilla linear bandit setting. Therefore, we need lemmas showing that these two UCB terms are close to each other.

Lemma 2.8.13. For any $\{\mathbf{x}_{h,k}\}_{h=1,k=1}^{H,K} \subset \mathbb{R}^d$ satisfying that $\|\mathbf{x}_{h,k}\|_2 \leqslant L, \forall (h,k) \in [H] \times [K]$, let $\mathbf{U}_{h,k} = \lambda \mathbf{I} + \sum_{\kappa=1}^{k-1} \sum_{i=1}^{H} \mathbf{x}_{i,\kappa} \mathbf{x}_{i,\kappa}^\top + \sum_{i=1}^{h-1} \mathbf{x}_{i,k} \mathbf{x}_{i,k}^\top$, there exists at most $2Hd \log(1 + KHL^2/\lambda)$ pairs of (h, k) tuple such that $\det \mathbf{U}_{h,k} \leqslant 2 \det \mathbf{U}_{1,k}$.

Lemma 2.8.14 (Lemma 12, Abbasi-Yadkori et al. (2011))**.** Suppose $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times d}$ are two positive definite matrices satisfying that $\mathbf{A} \succeq \mathbf{B}$, then for any $\mathbf{x} \in \mathbb{R}^d$, we have $\|\mathbf{x}\|_{\mathbf{A}} \leqslant \|\mathbf{x}\|_{\mathbf{B}} \sqrt{\det(\mathbf{A}) / \det(\mathbf{B})}$.

Following that, we also need to introduce the classical lemma to control the summation of the UCB bonus terms in vanilla linear bandit setting.

Lemma 2.8.15 (Lemma 11, Abbasi-Yadkori et al. (2011))**.** For any $\{\mathbf{x}_t\}_{t=1}^T \subset \mathbb{R}^d$ satisfying that $\|\mathbf{x}_t\|_2 \leqslant L, \forall t \in [T]$, let $\mathbf{U}_t = \lambda \mathbf{I} + \sum_{\tau=1}^{t-1} \mathbf{x}_\tau \mathbf{x}_\tau^\top$, we have

$$\sum_{t=1}^{T} \min\{1, \|\mathbf{x}_t\|_{\mathbf{U}_t^{-1}}\}^2 \leqslant 2d \log\left(\frac{d\lambda + TL^2}{d\lambda}\right).$$

We also need to introduce the Azuma-Hoeffding's inequality to build the concentration bound for martingale difference sequences.

Lemma 2.8.16 (Azuma-Hoeffding's inequality, Azuma (1967))**.** Let $\{x_i\}_{i=1}^n$ be a martingale difference sequence with respect to a filtration $\{\mathcal{G}_i\}_{i=1}^n$ (i.e. $\mathbb{E}[x_i|\mathcal{G}_i] = 0$ a.s. and x_i is $\mathcal{G}_{i+1}$ measurable) such that $|x_i| \leqslant M$ a.s.. Then for any $0 < \delta < 1$, with probability at least $1-\delta$, $\sum_{i=1}^n x_i \leqslant M\sqrt{2n\log(1/\delta)}$.

Proof of Lemma 2.8.3. By Lemma 2.8.1, for the k-th episode, we have

$$\begin{aligned} V_1^k(s_1^k) - V^{\pi^k}(s_1^k) &= \mathbb{E}\bigg[\sum_{h=1}^H \min\{H, 2\beta\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\}\bigg| s_1^k, \pi^k\bigg] \\ &\leqslant \mathbb{E}\bigg[\sum_{h=1}^H \min\{H, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\}\bigg| s_1^k, \pi^k\bigg] \end{aligned} \tag{2.8.17}$$

where the inequality comes from that the pseudo value function u_h^k defined in (2.4.3) is from maximizing the UCB term $\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}$ and we denote $\{\pi_h^k\}_{h=1}^H$ by π^k in short. By the definition of r_h^k, we have

$$\begin{aligned} V^{\pi^k}(s_1^k) &= \mathbb{E}[\sum_{h=1}^H r_h^k(s_h, \pi_h^k(s_h))|s_1^k, \pi^k] \\ &= \mathbb{E}\bigg[\sum_{h=1}^H \min\{1, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}/H\}\bigg| s_1^k, \pi^k\bigg]. \end{aligned} \tag{2.8.18}$$

Adding (2.8.17) and (2.8.18) together and taking summation over k, we have

$$\sum_{k=1}^K V_1^k(s_1^k) \leqslant \frac{H+1}{H} \underbrace{\sum_{k=1}^K \mathbb{E}\bigg[\sum_{h=1}^H \min\{H, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\}\bigg| s_1^k, \pi^k\bigg]}_{I_1} \leqslant 2I_1, \tag{2.8.19}$$

where the last inequality is due to $(H+1)/H \leqslant 2$. Next we are going to control the expectation of summation I_1. Consider the filtration $\{\mathcal{G}_{h,k}\}_{h=1,k=1}^{H,K}$ defined in Section 2.8.5.1, denote $x_{h,k}$ as follows:

$$x_{h,k} = \min\{H, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\} - \mathbb{E}_{s_h}\big[\min\{H, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\}\big],$$

then $x_{h,k}$ is obviously a martingale difference sequence bounded by H w.r.t. $\{\mathcal{G}_{h,k}\}_{h=1,k=1}^{H,K}$. Thus by Azuma-Hoeffding's inequality in Lemma 2.8.16, we have with probability at least

$1-\delta$, $\sum_{k=1}^{K}\sum_{h=1}^{H} x_h \leqslant H\sqrt{2HK\log(1/\delta)}$. Therefore,

$$
\begin{aligned}
I_1 &= \sum_{k=1}^{K}\sum_{h=1}^{H} \min\{H, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\} + \sum_{k=1}^{K}\sum_{h=1}^{H} x_h \\
&\leqslant 2\beta \sum_{k=1}^{K}\sum_{h=1}^{H} \min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\} + H\sqrt{2HK\log(1/\delta)} \\
&\leqslant 2\sqrt{2}\beta \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H} \min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\boldsymbol{\Sigma}_{h,k}^{-1}}\}}_{I_2} + 4\beta Hd\log(1+KH^3/\lambda) + H\sqrt{2HK\log(1/\delta)},
\end{aligned}
$$

where the inequality on the second line is due to $2\beta \geqslant 2H\sqrt{d\log 3} \geqslant H$ and the last inequality uses Lemma 2.8.14 with $\boldsymbol{\Sigma}_{1,k}^{-1} \succeq \boldsymbol{\Sigma}_{h,k}^{-1}$ and $\det \boldsymbol{\Sigma}_{1,k}^{-1} \leqslant 2\det \boldsymbol{\Sigma}_{1,k}^{-1}$ expect for $\widetilde{\mathcal{O}}(Hd)$ cases by Lemma 2.8.13. By $\min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h, \pi_h^k(s_h))\|_{\boldsymbol{\Sigma}_{h,k}^{-1}}\} \leqslant 1$ and $\|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_2 \leqslant H$ since $u_h^k \leqslant H$, we can further bound the $\widetilde{\mathcal{O}}(Hd)$ terms where $\det \boldsymbol{\Sigma}_{1,k}^{-1} > 2\det \boldsymbol{\Sigma}_{1,k}^{-1}$. To bound I_2, by Lemma 2.8.15, using Cauchy-Schwarz inequality we have

$$
I_2 \leqslant \sqrt{KH}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H} \min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h^k, a_h^k)\|_{\boldsymbol{\Sigma}_{h,k}^{-1}}^2\}} \leqslant \sqrt{2KHd\log(1+KH^3/(d\lambda))},
$$

Plugging I_2 into I_1 then plugging I_1 into (2.8.19). Let $\lambda = B^{-2}$, the summation of the value function $V_1^k(s_1^k)$ is bounded by

$$
\begin{aligned}
\sum_{k=1}^{K} V_1^k(s_1^k) \leqslant 8\beta\Big(\sqrt{HKd\log(1+KH^3B^2/d)} + dH\log(1+KH^3B^2)\Big) \\
+ 2H\sqrt{2HK\log(1/\delta)}.
\end{aligned}
$$

Taking $\delta = \delta/3$, we can finalize the proof of Lemma 2.8.3. □

2.8.5.5 Proof of Lemma 2.8.4

The proof of this lemma is similar to the proof of Lemma 5.2 in (Zhou et al., 2021a). We extend their proof to a *time varying* reward and *homogeneous* setting, where the rewards (i.e., the exploration-driven reward function r_h^k) are different in different episode k. To prove this lemma, we need to introduce the Bernstein inequality for vector-valued martingales.

Lemma 2.8.17 (Theorem 4.1, Zhou et al. (2021a)). Let $\{\mathcal{G}_t\}_{t=1}^{\infty}$ be a filtration, $\{\mathbf{x}_t, \eta_t\}_{t\geqslant 1}$ a stochastic process so that $\mathbf{x}_t \in \mathbb{R}^d$ is $\mathcal{G}_t$-measurable and η_t is $\mathcal{G}_{t+1}$-measurable. Fix $R, L, \sigma, \lambda > 0, \boldsymbol{\mu}^* \in \mathbb{R}^d$. For $t \geqslant 1$, let $y_t = \langle \boldsymbol{\mu}^*, \mathbf{x}_t \rangle + \eta_t$. Suppose $\eta_t, \mathbf{x}_t$ satisfy

$$|\eta_t| \leqslant R,\ \mathbb{E}[\eta_t|\mathcal{G}_t] = 0,\ \mathbb{E}[\eta_t^2|\mathcal{G}_t] \leqslant \sigma^2,\ \|\mathbf{x}_t\|_2 \leqslant L.$$

Then for any $0 < \delta < 1$, with probability at least $1-\delta$, we have

$$\forall t > 0,\ \left\| \sum_{\tau=1}^{t} \mathbf{x}_\tau \eta_\tau \right\|_{\mathbf{U}_\tau^{-1}} \leqslant \beta_t,\ \|\boldsymbol{\mu}_t - \boldsymbol{\mu}^*\|_{\mathbf{U}_t} \leqslant \beta_t + \sqrt{\lambda}\|\boldsymbol{\mu}^*\|_2,$$

where $\boldsymbol{\mu}_t = \mathbf{U}_t^{-1}\mathbf{b}_t, \mathbf{U}_t = \lambda\mathbf{I} + \sum_{\tau=1}^{t} \mathbf{x}_\tau \mathbf{x}_\tau^\top, \mathbf{b}_t = \sum_{\tau=i}^{t} y_\tau \mathbf{x}_\tau$, and

$$\beta_t = 8\sigma\sqrt{d\log(1 + tL^2/d\lambda)\log(4t^2/\delta)} + 4R\log(4t^2/\delta).$$

We also introduce the following lemma to analyze the error between the estimated variance $\bar{\mathbb{V}}_h^k$ and the true variance $\mathbb{V}_h^k$.

Lemma 2.8.18 (Lemma C.1, Zhou et al. (2021a)). Let $\mathbb{V}_h^k(s,a)$ be as defined in (2.3.1) and $\bar{\mathbb{V}}_h^k(s,a)$ be as defined in (2.5.3), then

$$\begin{aligned}|\mathbb{V}_h^k(s,a) - \bar{\mathbb{V}}_h^k(s,a)| \leqslant &\min\left\{H^2, \|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s,a)\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}} \|\widetilde{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}}\right\} \\ &+ \min\left\{H^2, 2H\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}} \|\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}}\right\}.\end{aligned}$$

Equipped with these lemmas, we can start the proof of Lemma 2.8.4.

Proof of Lemma 2.8.4. Recall the regression in (2.5.4). For the regression on $\widehat{\boldsymbol{\Sigma}}, \widehat{\boldsymbol{\theta}}$, let $\mathbf{x}_h^k = \boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)/\bar{\sigma}_h^k$, and $\eta_h^k = V_{h+1}^k(s_{h+1}^k)/\bar{\sigma}_h^k - \langle \boldsymbol{\theta}^*, \mathbf{x}_h^k \rangle$. Since $\bar{\sigma}_h^k \geqslant H/\sqrt{d}$ defined in (2.5.2), we get $\|\mathbf{x}_h^k\|_2 \leqslant \sqrt{d}, |\eta_h^k| \leqslant \sqrt{d}$, thus one could verify that $\mathbb{E}[[\eta_h^k]^2|\mathcal{G}_{h,k}] \leqslant d$, $\mathbb{E}[\eta_h^k|\mathcal{G}_{h,k}] = 0$, from Lemma 2.8.17, taking $t = (k-1)H$ we have

$$\begin{aligned}\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant &\, 8d\sqrt{\log(1 + (k-1)H/\lambda)\log(4(k-1)^2H^2/\delta)} \\ &+ 4\sqrt{d}\log(4(k-1)^2H^2/\delta) + \sqrt{\lambda}B.\end{aligned}$$

For the regression of $\widetilde{\boldsymbol{\Sigma}}, \widetilde{\boldsymbol{\theta}}$, $\mathbf{x}_h^k = \boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)$ which directly implies $\|\mathbf{x}_h^k\|_2 \leqslant H^2$. Let $\eta_h^k = V_{h+1}^k(s_{h+1}^k)^2 - \langle \boldsymbol{\theta}^*, \mathbf{x}_h^k \rangle$, one can easily verify that $|\eta_h^k| \leqslant H^2$ and $\mathbb{E}[\eta_h^k | \mathcal{G}_{h,k}] = 0, \mathbb{E}[[\eta_h^k]^2 | \mathcal{G}_{h,k}] \leqslant H^4$, thus using Lemma 2.8.17 again we have

$$\begin{aligned}\|\boldsymbol{\theta}^* - \widetilde{\boldsymbol{\theta}}_k\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}} \leqslant\ & 8H^2\sqrt{d\log(1+(k-1)H/\lambda)\log(4(k-1)^2H^2/\delta)} \\ & + 4H^2\log(4(k-1)^2H^2/\delta) + \sqrt{\lambda}B.\end{aligned}$$

Since $\lambda = B^{-2}$, if we select $\check{\beta}$ and $\widetilde{\beta}$ as

$$\begin{aligned}\check{\beta} &= 8d\sqrt{\log(1+KHB^2/)\log(4K^2H^2/\delta)} + 4\sqrt{d}\log(4(k-1)^2H^2/\delta) + 1 \\ \widetilde{\beta} &= 8H^2\sqrt{d\log(1+KHB^2)\log(4K^2H^2/\delta)} + 4H^2\log(4K^2H^2/\delta) + 1,\end{aligned}$$

then with probability at least $1-2\delta$, for all $k \in [K+1]$, $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \check{\beta}$, $\|\boldsymbol{\theta}^* - \widetilde{\boldsymbol{\theta}}_k\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widetilde{\beta}$.

Next we are going to give the choice of $\widehat{\beta}$ to make sure that $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \check{\beta}$ holds with high probability. The following proof is conditioned on that the aforementioned event $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \check{\beta}$, $\|\boldsymbol{\theta}^* - \widetilde{\boldsymbol{\theta}}_k\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widetilde{\beta}$ holds, then from Lemma 2.8.18 we have

$$\begin{aligned}&|\mathbb{V}_h^k(s,a) - \bar{\mathbb{V}}_h^k(s,a)| \\ &\quad\leqslant \min\left\{H^2, \widetilde{\beta}\|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s,a)\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}}\right\} + \min\left\{H^2, 2\check{\beta}H\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\right\} \\ &\quad= E_h^k(s,a) \qquad (2.8.20)\end{aligned}$$

Again, let $\mathbf{x}_h^k = \boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)/\bar{\sigma}_h^k$ to denote the context vector and $\eta_h^k = V_{h+1}^k(s_{h+1}^k)/\bar{\sigma}_h^k - \langle \boldsymbol{\theta}^*, \mathbf{x}_h^k \rangle$ to denote the noise term, since $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \check{\beta}$, we have

$$\mathbb{E}[[\eta_h^k]^2 | \mathcal{G}_{h,k}] = \mathbb{V}_h^k(s_h^k, a_h^k)/\nu_h^k \leqslant (E_h^k(s_h^k, a_h^k) + \bar{\mathbb{V}}_h^k(s_h^k, a_h^k))/\nu_h^k \leqslant 1,$$

where the first inequality is from (2.8.20), the second inequality holds because the definition of ν_h^k in (2.5.2).

Therefore we have verified that the noise term η_h^k is a zero-mean random variable conditioned on $\mathcal{G}_{h,k}$ and $\mathbb{E}[[\eta_h^k]^2 | \mathcal{G}_{h,k}] \leqslant 1$. In that case, using Lemma 2.8.17 again we could get

with probability at least $1-\delta$,

$$\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant 8\sqrt{d(1+(k-1)H/\lambda)\log(4(k-1)^2H^2/\delta)} \tag{2.8.21}$$

$$+ 4\sqrt{d}\log(4(k-1)^2H^2/\delta) + \sqrt{\lambda}B, \tag{2.8.22}$$

again, since $\lambda = B^{-2}$, if we select $\widehat{\beta}$ as

$$\widehat{\beta} = 8\sqrt{d(1+KHB^2)\log(4K^2H^2/\delta)} + 4\sqrt{d}\log(4K^2H^2/\delta) + 1,$$

then $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widehat{\beta}$ with probability at least $1-\delta$ for all $k \in [K+1]$.

Next, for the regression of $\boldsymbol{\theta}_{K+1}, \boldsymbol{\Sigma}_{1,K+1}$, by Lemma 2.8.2, we obtain the same result with the selection of β as

$$\beta = H\sqrt{d\log\left(\frac{1+KH^3/\lambda}{\delta}\right)} + B\sqrt{\lambda},$$

which suggests that with probability at least $1-\delta$, $\|\boldsymbol{\theta}_{K+1} - \boldsymbol{\theta}^*\|_{\boldsymbol{\Sigma}_{1,K+1}} \leqslant \beta$. Then taking union bound with all aforementioned event $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \check{\beta}$, $\|\boldsymbol{\theta}^* - \widetilde{\boldsymbol{\theta}}_k\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widetilde{\beta}$, $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_k\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}} \leqslant \widehat{\beta}$, we have all these events mentioned in this proof holds with probability at least $1-4\delta$. Replace δ with $\delta/12$, we obtain our final results.

Next, for the regression of $\boldsymbol{\theta}_{K+1}, \boldsymbol{\Sigma}_{1,K+1}$, by Lemma 2.8.2, we obtain the same result with the selection of β as

$$\beta = H\sqrt{d\log\left(\frac{1+KH^3/\lambda}{\delta}\right)} + B\sqrt{\lambda},$$

which suggests that with probability at least $1-\delta$, $\|\boldsymbol{\theta}_{K+1} - \boldsymbol{\theta}^*\|_{\boldsymbol{\Sigma}_{1,K+1}} \leqslant \beta$. Again, taking an additional union bound, with probability at least $1-4\delta$, all events mentioned in this proof hold. Replace δ with $\delta/12$, we obtain our final results. □

2.8.5.6 Proof of Lemma 2.8.5

The proof of this lemma borrows some intuition from the proof of Theorem 5.3 in (Zhou et al., 2021a). Unlike Zhou et al. (2021a) that deals the fixed reward and time-inhomogeneous

setting, we need to extend their proof in order to deal with the time-varying reward and time-homogeneous setting.

The next lemmas shows the relationship between the summation of ν_h^k and the difference between $V_h^k(s)$ calculated in Algorithm 3 and $V_h^{\pi^k}(s;\{r_h^k\}_{h=1,k=1}^{H,K})$

Lemma 2.8.19. Let V_h^k, ν_h^k be defined in Algorithm 3. Then if the condition in Lemma 2.8.4 holds, the following inequality holds with probability at least $1-2\delta$,

$$\begin{aligned}\sum_{k=1}^{K}[V_1^k(s_1^k)-V_1^{\pi^k}(s_1^k)] &\leqslant 4\sqrt{d}\widehat{\beta}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}\sqrt{\log(1+KHB^2)}\\ &\quad+2H^2d\log(1+KHB^2d)+H\sqrt{2KH\log(1/\delta)}\\ \sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}(V_{h+1}^k-V_{h+1}^{\pi^k})](s_h^k,a_h^k) &\leqslant 4\sqrt{d}H\widehat{\beta}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}\sqrt{\log(1+KHB^2)}\\ &\quad+2H^3d\log(1+KHB^2d)+2H^2\sqrt{2KH\log(1/\delta)},\end{aligned}$$

Lemma 2.8.20. Let V_h^k, ν_h^k be defined in Algorithm 3. Then if the condition in Lemma 2.8.4 holds, with probability at least $1-\delta$,

$$\begin{aligned}\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k &\leqslant \frac{H^3K}{d}+3H^2K+3H^3\log(1/\delta)+2H\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}(V_{h+1}^k-V_{h+1}^{\pi^k}](s_h^k,a_h^k)\\ &\quad+2\widetilde{\beta}\sqrt{KHd\log(1+KH^5B^2/d)}+4\widetilde{\beta}Hd\log(1+KH^5B^2/d)\\ &\quad+8H^2\check{\beta}\sqrt{KHd\log(1+KHB^2)}+8H^3d\check{\beta}\log(1+KHdB^2).\end{aligned}$$

Equipped with these two lemmas, we can start to prove Lemma 2.8.5.

Proof of Lemma 2.8.5. In this proof, we use $\widetilde{\mathcal{O}}(\cdot)$ to ignore all constant and log terms to simplify the results. Recall the selection of $\beta, \widehat{\beta}, \check{\beta}, \widetilde{\beta}$, we have $\beta=\widetilde{\mathcal{O}}(H\sqrt{d})$, $\widehat{\beta}=\widetilde{\mathcal{O}}(\sqrt{d})$, $\check{\beta}=\widetilde{\mathcal{O}}(d)$, $\widetilde{\beta}=\widetilde{\mathcal{O}}(H^2\sqrt{d})$. Therefore Lemma 2.8.19 could be simplified as

$$\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}(V_{h+1}^k-V_{h+1}^{\pi^k})](s_h^k,a_h^k)\leqslant\widetilde{\mathcal{O}}\left(Hd\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}+H^3d+\sqrt{KH^5}\right). \tag{2.8.23}$$

Lemma 2.8.20 could also be simplified as

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k \leqslant \widetilde{\mathcal{O}}\left(\frac{H^3K}{d}+H^2K+H\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}(V_{h+1}^k-V_{h+1}^{\pi^k})](s_h^k,a_h^k)+\sqrt{KH^5d^3}+H^3d^2\right). \tag{2.8.24}$$

Let $\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}=x$, plugging (2.8.23) into (2.8.24), we have

$$x^2 \leqslant \widetilde{\mathcal{O}}(H^3Kd^{-1}+H^2K+H^2dx+H^4d+\sqrt{KH^7}+\sqrt{KH^5d^3}+H^3d^2),$$

Since the quadratic inequality $x^2 \leqslant \widetilde{\mathcal{O}}(bx+c)$ indicates that $x \leqslant \mathcal{O}(b+\sqrt{c})$, setting

$$b=\widetilde{\mathcal{O}}(H^2d), c=\widetilde{\mathcal{O}}(H^3Kd^{-1}+H^2K+H^4d+\sqrt{KH^7}+\sqrt{KH^5d^3}+H^3d^2),$$

hence

$$\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k} \leqslant \widetilde{\mathcal{O}}(H^2d+\sqrt{H^3K/d}+H\sqrt{K}+H^2\sqrt{d}+d\sqrt{H^3}+(KH^7)^{1/4}+(KH^5d^3)^{1/4}) \tag{2.8.25}$$

$$=\widetilde{\mathcal{O}}(\sqrt{H^3K/d}+H\sqrt{K})+o(\sqrt{K}). \tag{2.8.26}$$

Plugging (2.8.26) back to Lemma 2.8.19, we have

$$\sum_{k=1}^{K}[V_1^k(s_1^k)-V_1^{\pi^k}(s_1^k)] \leqslant \widetilde{\mathcal{O}}(\sqrt{H^3Kd}+Hd\sqrt{K})+o(\sqrt{K}). \tag{2.8.27}$$

Next we are going to show the bound of the summation over $V_1^{\pi^k}(s_1^k)$, note that this value function is bounded by H and from Bellman equality, we have

$$V_h^{\pi^k}(s_1^k)=r_h^k(s_1^k,a_1^k)+[\mathbb{P}V_{h+1}^{\pi^k}](s_h^k,a_h^k),$$

taking summation over $h\in[H], k\in[K]$ then

$$\begin{aligned}\sum_{k=1}^{K}V_1^{\pi^k}(s_1^k)&=\sum_{k=1}^{K}\sum_{h=1}^{H}r_h^k(s_1^k,a_1^k)+\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}V_{h+1}^{\pi^k}](s_h^k,a_h^k)-V_{h+1}^{\pi^k}(s_{h+1}^k)\\&\leqslant\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1,2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h^k,a_h^k)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}/H\}+H\sqrt{HK\log(1/\delta)},\end{aligned}$$

where the last inequality holds due to Azuma-Hoeffding's inequality i.e. Lemma 2.8.16. For the first term,

$$\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, 2\beta\|\boldsymbol{\psi}_{u_h^k}(s_h^k,a_h^k)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}/H\} \leqslant \frac{2\beta}{H}\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h^k,a_h^k)\|_{\boldsymbol{\Sigma}_{1,k}^{-1}}\}}_{I_1},$$

where the inequality is due to $\beta \geqslant H\sqrt{\log(12)} \geqslant H/2$. Since Lemma 2.8.13 suggests that there are only up to $\widetilde{\mathcal{O}}(Hd)$ steps with $\det \boldsymbol{\Sigma}_{1,k}^{-1} \leqslant 2\det \boldsymbol{\Sigma}_{h,k}^{-1}$, by Lemma 2.8.13 and Lemma 2.8.14 with $\boldsymbol{\Sigma}_{1,k}^{-1} \geq \boldsymbol{\Sigma}_{h,k}^{-1}$ and setting $\lambda = B^{-2}$, we have

$$\begin{aligned}
I_1 &\leqslant 2Hd\log(1+KH^3B^2) + \sqrt{2}\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h^k,a_h^k)\|_{\boldsymbol{\Sigma}_{h,k}^{-1}}\} \\
&\leqslant 2Hd\log(1+KH^3B^2) + \sqrt{2HK}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{u_h^k}(s_h^k,a_h^k)\|^2_{\boldsymbol{\Sigma}_{h,k}^{-1}}\}} \\
&\leqslant 2Hd\log(1+KH^3B^2) + 2\sqrt{HKd\log(1+KH^3B^2/d)}.
\end{aligned}$$

Therefore, since $\beta = \widetilde{\mathcal{O}}(H\sqrt{d})$, then

$$\sum_{k=1}^{K} V_1^{\pi^k}(s_1^k) \leqslant 4\beta d\log(1+KH^3B^2) + 4\beta\sqrt{Kd\log(1+KH^3B^2/d)/H} + \sqrt{H^3K\log(1/\delta)} \tag{2.8.28}$$

$$\leqslant \widetilde{\mathcal{O}}(d\sqrt{KH} + \sqrt{KH^3}) + o(\sqrt{K}). \tag{2.8.29}$$

Adding (2.8.27) and (2.8.29) together, we have the following result,

$$\sum_{k=1}^{K} V_1^k(s_1^k) \leqslant \widetilde{\mathcal{O}}(\sqrt{H^3Kd} + Hd\sqrt{K}) + o(\sqrt{K}).$$

By taking the union bound, this inequality holds with probability at least $1-4\delta$. Since δ only appears in the logarithmic terms, thus changing δ to $\delta/12$ will not affect the result. □

2.8.6 Proof of Auxiliary Lemmas in Section 2.8.5

2.8.6.1 Proof of Lemma 2.8.13

Proof of Lemma 2.8.13. We bound the number of tuples (h, k) with $\det \mathbf{U}_{h,k} \geqslant 2 \det \mathbf{U}_{1,k}$. To begin with, if there exists $k \in [K]$ such that $\det \mathbf{U}_{1,k+1} \leqslant 2 \det \mathbf{U}_{1,k}$, then it is obvious that for all $h \in [H]$, we have $\det \mathbf{U}_{h,k} \leqslant \det \mathbf{U}_{1,k+1} \leqslant 2 \det \mathbf{U}_{1,k}$.

Therefore, suppose there exists a set $\mathcal{K} \subset [K]$ such that for all $k \notin \mathcal{K}$, $\det \mathbf{U}_{1,k+1} \leqslant 2 \det \mathbf{U}_{1,k}$ and for all $k \in \mathcal{K}$, $\det \mathbf{U}_{1,k+1} > 2 \det \mathbf{U}_{1,k}$, then the pair of (h, k) such that $\det \mathbf{U}_{h,k} \geqslant 2 \det \mathbf{U}_{1,k}$ is upper bounded by $H|\mathcal{K}|$.

Notice that for all $k \in \mathcal{K}$, $\det \mathbf{U}_{1,k+1} > 2 \det \mathbf{U}_{1,k}$, it is easy to show that

$$\det \mathbf{U}_{1,K+1} > 2^{|\mathcal{K}|} \det \mathbf{U}_{1,1} = 2^{|\mathcal{K}|} \lambda^d,$$

where the last inequality comes from $\mathbf{U}_{1,1} = \lambda \mathbf{I} \in \mathbb{R}^{d \times d}$. Notice that $\det \mathbf{U} \leqslant \|\mathbf{U}\|_2^d$, taking log we have

$$d \log(\|\mathbf{U}_{1,K+1}\|_2) \geqslant \log \det \mathbf{U}_{1,K+1} > |\mathcal{K}| \log 2 + d \log \lambda. \tag{2.8.30}$$

From the definition of $\mathbf{U}_{1,K+1}$, by triangle inequality,

$$\|\mathbf{U}_{1,K+1}\|_2 \leqslant \lambda + \sum_{k=1} K \sum_{h=1}^{H} \|\mathbf{x}_h^k \mathbf{x}_h^{k\top}\|_2 \leqslant \lambda + KH \|\mathbf{x}_h^k\|_2^2 \leqslant \lambda + KHL^2, \tag{2.8.31}$$

where the last inequality is due to $\|\mathbf{x}\|_2 \leqslant L$ from the statement of the lemma. Therefore we conclude our proof by merging (2.8.30) and (2.8.31) together to get

$$|\mathcal{K}| \log 2 < d \log(1 + HKL^2/\lambda),$$

noticing $\log 2 \geqslant 1/2$ we can get the result claimed in the lemma. □

2.8.6.2 Proof of Lemma 2.8.19

Proof of Lemma 2.8.19. Assume that the condition in Lemma 2.8.4 holds, then

$$\begin{aligned}
&V_h^k(s_h^k) - V_h^{\pi^k}(s_h^k) \\
&\leqslant \langle \widehat{\boldsymbol{\theta}}_k, \boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\rangle - [\mathbb{P}V_{h+1}^{\pi^k}](s_h^k, a_h^k) + \widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}} \\
&\leqslant \|\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}^*\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}} + [\mathbb{P}V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k, a_h^k) + \widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}} \\
&\leqslant 2\widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}} + [\mathbb{P}V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k, a_h^k),
\end{aligned}$$

where the first inequality holds due to the definition of V_h^k, the second inequality holds due to Cauchy-Schwarz inequality and the third one holds due to the condition (2.8.8) in Lemma 2.8.4. Notice that $V_h^k - V_h^{\pi^k} \leqslant H$, we have

$$V_h^k(s_h^k) - V_h^{\pi^k}(s_h^k) \leqslant \min\{H, 2\widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\} + [\mathbb{P}V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k, a_h^k)$$

Taking summation over $k \in [K]$ and $h \in [H]$, we have

$$\begin{aligned}
\sum_{k=1}^K [V_1^k(s_1^k) - V_1^{\pi_k}(s_1^k)] &\leqslant \sum_{k=1}^K \sum_{h=1}^H \min\{H, 2\widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\} \\
&\quad + \sum_{k=1}^K \sum_{h=1}^H \Big[[\mathbb{P}V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k, a_h^k) - [V_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi^k}(s_{h+1}^k)]\Big] \\
&\leqslant \underbrace{\sum_{k=1}^K \sum_{h=1}^H \min\{H, 2\widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k, a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\}}_{I_1} + H\sqrt{2KH\log(1/\delta)},
\end{aligned} \tag{2.8.32}$$

where the second inequality is due to Azuma-Hoeffding's inequality as in Lemma 2.8.16.

Next we bound I_1. Recall the update rule of $\widehat{\boldsymbol{\Sigma}}_{h,k}$, notice that $\bar{\sigma}_h^k \geqslant H/\sqrt{d}$ and the fact that $\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^K, a_h^K)\|_2 \leqslant H$ from $V_{h+1}^k \leqslant H$, it is easy to verify that $\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^K, a_h^K)/\widehat{\sigma}_h^k\|_2 \leqslant$

$\sqrt{d}$. Hence

$$
\begin{aligned}
I_1 &\leqslant \sqrt{2}\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{H, 2\widehat{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k,a_h^k)\|_{\widehat{\boldsymbol{\Sigma}}_{h,k}^{-1}}\} + 2H^2 d\log(1+KHd/\lambda)\\
&\leqslant \sqrt{2}\max\{\sqrt{d}, 2\widehat{\beta}\}\sum_{k=1}^{K}\sum_{h=1}^{H}\bar{\sigma}_h^k\min\{1, \|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k,a_h^k)/\bar{\sigma}_h^k\|_{\widehat{\boldsymbol{\Sigma}}_{h,k}^{-1}}\} + 2H^2 d\log(1+KHd/\lambda)\\
&\leqslant 2\sqrt{2}\widehat{\beta}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{V_{h+1}^k}(s_h^k,a_h^k)/\bar{\sigma}_h^k\|^2_{\widehat{\boldsymbol{\Sigma}}_{h,k}^{-1}}\}} + 2H^2 d\log(1+KHd/\lambda)\\
&\leqslant 4\widehat{\beta}\sqrt{d}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}\sqrt{\log(1+KH/\lambda)} + 2H^2 d\log(1+KHd/\lambda),
\end{aligned}
$$

where the first inequality, similar to the corresponding proof in Lemma 2.8.3, is a direct implication of Lemma 2.8.13 and Lemma 2.8.14 with $\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1} \succeq \widehat{\boldsymbol{\Sigma}}_{h,k}^{-1}$ and $\det\boldsymbol{\Sigma}_{1,k}^{-1} \leqslant 2\det\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}$ except for $\widetilde{\mathcal{O}}(Hd)$ cases mentioned in Lemma 2.8.13, the second inequality moves $\bar{\sigma}_h^k$ outside, the third inequality holds because $\widehat{\beta} \geqslant 4\sqrt{d}\log 12 \geqslant \sqrt{d}$ and Cauchy-Schwarz inequality, and the forth inequality holds due to Lemma 2.8.15. Plugging I_1 into (2.8.32) and let $h'=1, \lambda = B^{-2}$, we have

$$
\begin{aligned}
\sum_{k=1}^{K}[V_1^k(s_1^k) - V_1^{\pi^k}(s_1^k)] \leqslant{}& 4\sqrt{d}\widehat{\beta}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}\sqrt{\log(1+KHB^2)}\\
&+ 2H^2 d\log(1+KHB^2d) + H\sqrt{2KH\log(1/\delta)}.
\end{aligned}
$$

Furthermore, by Azuma-Hoeffding's inequality as in Lemma 2.8.16,

$$
\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{P}[V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k,a_h^k) ={}& \sum_{k=1}^{K}\sum_{h=2}^{H}[V_h^k - V_h^{\pi^k}](s_h^k)\\
&+ \sum_{k=1}^{K}\sum_{h=1}^{H}\Big[\mathbb{P}(V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k,a_h^k) - [V_{h+1}^k - V_{h+1}^{\pi^k})](s_{h+1}^k)\\
\leqslant{}& 4\sqrt{d}H\widehat{\beta}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k}\sqrt{\log(1+KHB^2)}\\
&+ 2H^3 d\log(1+KHB^2d) + (H+1)H\sqrt{2KH\log(1/\delta)},
\end{aligned}
$$

which becomes the second part of the statement in the lemma. Using $H+1 \leqslant 2H$ we can get the result claimed in the lemma. □

2.8.6.3 Proof of Lemma 2.8.20

To begin with, we will first show the total variance lemma originally introduced in (Jin et al., 2018).

Lemma 2.8.21 (Total variance lemma, Lemma C.5, Jin et al. (2018))**.** [2] With probability at least $1-\delta$, we have

$$\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{V}V_h^{\pi^k}(\cdot;\{r_h^k\}_{h=1}^H)](s,a) \leqslant 3H^2K + 3H^3\log(1/\delta).$$

Proof of Lemma 2.8.20. Assume the condition in Lemma 2.8.4 holds, we have with probability at least $1-\delta$,

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k &\leqslant \sum_{k=1}^{K}\sum_{h=1}^{H}\left(\frac{H^2}{d} + \bar{\mathbb{V}}_h^k(s_h^k,a_h^k) + E_h^k(s_h^k,a_h^k)\right) \\
&= \frac{H^3K}{d} + \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\left([\mathbb{V}_hV_{h+1}^k](s_h^k,a_h^k) - [\mathbb{V}_hV_{h+1}^{\pi^k}](s_h^k,a_h^k)\right)}_{I_1} + 2\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}E_h^k(s_h^k,a_h^k)}_{I_2} \\
&\quad + \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{V}_hV_{h+1}^{\pi^k}](s_h^k,a_h^k)}_{I_3} + \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\left[\bar{\mathbb{V}}_h^k(s_h^k,a_h^k) - [\mathbb{V}_hV_{h+1}^k](s_h^k,a_h^k) - E_h^k\right]}_{I_4} \\
&\leqslant \frac{H^3K}{d} + I_1 + I_2 + 3H^2K + 3H^3\log(1/\delta), \qquad (2.8.33)
\end{aligned}$$

where the value function $V_h^{\pi^k}(s)$ is short for $V_h^{\pi^k}(s;\{r_h^k\}_{h=1}^H)$ for simplicity. The first inequality is from the definition of ν_h^k in (2.5.2), while the last inequality is from Lemma 2.8.21 to control I_3. $I_4 \leqslant 0$ is due to Lemma 2.8.4. Next we are about to bound I_1 and I_2 separately.

[2] The original Lemma C.5 in Jin et al. (2018) holds for the identical reward functions, i.e., $r_h^1 = \cdots = r_h^K$. Their lemma also holds for the general case $r_h^1 \neq \cdots \neq r_h^K$ without changing their proof.

Since the estimated value function V_{h+1}^k and the real value function $V_{h+1}^{\pi^k}$ are both bounded by $[0, H]$, we have

$$I_1 \leqslant \sum_{k=1}^{K}\sum_{h=1}^{H}\left[\mathbb{P}([V_{h+1}^k]^2 - [V_{h+1}^{\pi^k}]^2)\right](s_h^k, a_h^k) \leqslant 2H\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}(V_{h+1}^k - V_{h+1}^{\pi^k})](s_h^k, a_h^k).$$

For term I_2, we have

$$\begin{aligned}
I_2 &\leqslant \sum_{k=1}^{K}\sum_{h=1}^{H}\min\{H^2, \widetilde{\beta}\|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}}\} + \sum_{k=1}^{K}\sum_{h=1}^{H}\min\{H^2, 2H\check{\beta}\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\} \\
&\leqslant \max\{H^2, \widetilde{\beta}\}\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}}\} \\
&\quad + \sum_{k=1}^{K}\sum_{h=1}^{H}\max\{H^2, 2H\check{\beta}\bar{\sigma}_h^k\}\min\left\{1, \left\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)/\bar{\sigma}_h^k\right\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\right\}.
\end{aligned}$$

Noticing that from the definition of ν_h^k,

$$\nu_k^h = \max\{H^2/d, \bar{\mathbb{V}}_h^k(s_h^k, a_h^k) + E_h^k(s_h^k, a_h^k)\} \leqslant \max\{H^2/d, H^2 + 2H^2\} = 3H^2,$$

thus $\bar{\sigma}_h^k = \sqrt{\nu_h^k} \leqslant 2H$. Recall that $\widetilde{\beta} \geqslant 4H^2\log(12) \geqslant H^2$ and $\check{\beta} \geqslant 1$, we have

$$I_2 \leqslant \underbrace{\widetilde{\beta}\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}}\}}_{I_5} + 4H^2\check{\beta}\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{1, \left\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)/\bar{\sigma}_h^k\right\|_{\widehat{\boldsymbol{\Sigma}}_{1,k}^{-1}}\right\}}_{I_6}.$$

For I_5, using Lemmas 2.8.13 and 2.8.14 with $\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1} \succeq \widetilde{\boldsymbol{\Sigma}}_{h,k}^{-1}$ and $\det\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1} \leqslant 2\det\widetilde{\boldsymbol{\Sigma}}_{1,k}^{-1}$ except for $\widetilde{\mathcal{O}}(Hd)$ cases mentioned in Lemma 2.8.13, we have

$$\begin{aligned}
I_5 &\leqslant \sqrt{2}\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Sigma}}_{h,k}^{-1}}\} + 2Hd\log(1 + KH^5/d\lambda) \\
&\leqslant \sqrt{2KH}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{1, \|\boldsymbol{\psi}_{[V_{h+1}^k]^2}(s_h^k, a_h^k)\|_{\widetilde{\boldsymbol{\Sigma}}_{h,k}^{-1}}^2\}} + 2Hd\log(1 + KH^5/d\lambda) \\
&\leqslant 2\sqrt{KHd\log(1 + KH^5/d\lambda)} + 2Hd\log(1 + KH^5/d\lambda),
\end{aligned}$$

where the first inequality is a direct implication from Lemma 2.8.13 and the second inequality is due to Cauchy-Schwarz inequality. The third inequality utilizes Lemma 2.8.15. As for I_6,

we have

$$\begin{aligned}
I_6 &\leqslant \sqrt{2}\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{1, \left\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)/\bar{\sigma}_h^k\right\|_{\widehat{\boldsymbol{\Sigma}}_{h,k}^{-1}}\right\} + 2Hd\log(1+KHd/\lambda) \\
&\leqslant \sqrt{2KH}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{1, \left\|\boldsymbol{\psi}_{V_{h+1}^k}(s,a)/\bar{\sigma}_h^k\right\|_{\widehat{\boldsymbol{\Sigma}}_{h,k}^{-1}}\right\}} + 2Hd\log(1+KHd/\lambda) \\
&\leqslant 2\sqrt{KHd\log(1+KH/\lambda)} + 2Hd\log(1+KHd/\lambda).
\end{aligned}$$

Finally, plugging I_5, I_6 into I_2 and I_1, I_2 into (2.8.33) we have

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H}\nu_h^k \leqslant{}& \frac{H^3K}{d} + 3H^2K + 3H^3\log(1/\delta) + 2H\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}(V_{h+1}^k - V_{h+1}^{\pi^k}](s_h^k, a_h^k) \\
&+ 2\widetilde{\beta}\sqrt{KHd\log(1+KH^5/d\lambda)} + 4\widetilde{\beta}Hd\log(1+KH^5/d\lambda) \\
&+ 8H^2\check{\beta}\sqrt{KHd\log(1+KH/\lambda)} + 8H^3d\check{\beta}\log(1+KHd/\lambda).
\end{aligned}$$

Using $\lambda = B^{-2}$, we could get the result in the statement of the lemma. □

2.8.7 Missing Proof in Section 2.8.3

2.8.7.1 Proof of Lemma 2.8.7

To start with, we recall that event $\mathcal{E}_{2.8.6}$ is defined by the the case when Lemma 2.8.6 holds. And the following lemmas are conditioned on $\mathcal{E}_{2.8.6}$ by default. We define function W_h for certain sequence $\{R_h\}$ recursively as

$$W_h\left(\{R_h\}\right) = \min\left\{1, R_h + W_{h+1}\left(\{R_h\}\right)\right\}.$$

In addition we denote the trajectory of first h steps as $\texttt{traj}_h := (s_1, a_1, \cdots, s_{h-1}, a_{h-1}, s_h)$, and the trajectory sampled from $(\pi, \mathbb{P})$ conditioned on $\texttt{traj}_h$ as $\texttt{traj} \sim (\pi, \mathbb{P})|\texttt{traj}_h$.

Lemma 2.8.22. For any policy π and reward function $r \in R$, we have

$$V_1(s_1; \boldsymbol{\theta}_K, \pi, r) - V_1(s_1; \boldsymbol{\theta}^*, \pi, r) = \mathbb{E}_{\texttt{traj}\sim(\pi,\mathbb{P})|\texttt{traj}_1} W_1(\{(\mathbb{P}_K - \mathbb{P})V_{h+1}(s_h; \boldsymbol{\theta}_K, \pi, r)\}) \tag{2.8.34}$$

Lemma 2.8.23. For any policy π and reward function $r \in R$, we have

$$\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\{u_{k,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r)\}\right) \leqslant \widehat{V}_{K,1}\left(s_1;\boldsymbol{\theta}_K,\pi_K,r_K\right).$$

Proof of Lemma 2.8.7. The proof follows the proof of Lemma 15 in Zhang et al. (2020). Firstly,

$$\begin{aligned}
&V_1^*(s_1;r) - V_1(s_1;\boldsymbol{\theta}^*,\widehat{\pi}_r,r) \\
&= (V_1^*(s_1;r) - V_1(s_1;\boldsymbol{\theta}_K,\widehat{\pi}_r,r)) + (V_1(s_1;\boldsymbol{\theta}_K,\widehat{\pi}_r,r) - V_1(s_1;\boldsymbol{\theta}^*,\widehat{\pi}_r,r)) \\
&\leqslant (V_1^*(s_1;r) - V_1(s_1;\boldsymbol{\theta}_K,\pi_r^*,r)) + (V_1(s_1;\boldsymbol{\theta}_K,\widehat{\pi}_r,r) - V_1(s_1;\boldsymbol{\theta}^*,\widehat{\pi}_r,r)),
\end{aligned} \tag{2.8.35}$$

where π_r^* is the optimal policy for $(\boldsymbol{\theta},r)$, and $\widehat{\pi}_r$ is the optimal policy for $(\boldsymbol{\theta}_K,r)$. Then for any policy $\pi \in \Pi$,

$$\begin{aligned}
&|V_1(s_1;\boldsymbol{\theta}_K,\pi,r) - V_1(s_1;\boldsymbol{\theta}^*,\pi,r)| \\
&= \left|\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\{(\mathbb{P}_K - \mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\}\right)\right| \\
&= \left|\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\left\{(\boldsymbol{\theta}_K - \boldsymbol{\theta}^*)\boldsymbol{\phi}_{V_{h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,a_h)\right\}\right)\right| \\
&\leqslant \mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\left\{\|\boldsymbol{\theta}_K - \boldsymbol{\theta}^*\|_{\dot{\boldsymbol{\Sigma}}_{k,0}} \left\|\boldsymbol{\phi}_{V_{h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,a_h)\right\|_{\dot{\boldsymbol{\Sigma}}_{k,0}^{-1}}\right\}\right) \\
&\leqslant \mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\left\{2\beta \left\|\boldsymbol{\phi}_{V_{h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,a_h)\right\|_{\dot{\boldsymbol{\Sigma}}_{k,0}^{-1}}\right\}\right) \\
&= 2\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\{u_h(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\}\right) \\
&\leqslant 2\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K).
\end{aligned} \tag{2.8.36}$$

The first equality holds due to Lemma 2.8.22, the second inequality holds due to Cauchy-Schwartz inequality, the third inequality holds due to Lemma 2.8.6, and the last inequality holds due to Lemma 2.8.23. Plugging (2.8.36) into (2.8.36), we obtain

$$\begin{aligned}
V_1^*(s_1;r) - V_1(s_1;\boldsymbol{\theta}^*,\widehat{\pi}_r,r) &\leqslant 2\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K) + 2\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K) \\
&= 4\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K).
\end{aligned}$$

□

2.8.7.2 Proof of Lemma 2.8.8

Proof of Lemma 2.8.8. The proof follows the proof of Lemma 14 in Chen et al. (2021). Firstly, we prove that $\widehat{V}_{k,1}(s;\boldsymbol{\theta},\pi,r)$ is non-increasing w.r.t. k for any fixed $\boldsymbol{\theta},\pi,r$ by induction in h. Suppose for any $k_1 \leqslant k_2$, $\widehat{V}_{k_1,h+1}(s;\boldsymbol{\theta},\pi,r) \geqslant \widehat{V}_{k_2,h+1}(s;\boldsymbol{\theta},\pi,r)$ for any s. By definition,

$$\begin{aligned}\widehat{V}_{k,h}(s;\boldsymbol{\theta},\pi,r) = \min\Bigg\{1, u_{k,h}(s,a;\boldsymbol{\theta},\pi,r) + 2\beta \left\|\boldsymbol{\phi}_{\widehat{V}_{k,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\right\|_{\dot{\hat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} \\ + \boldsymbol{\phi}^\top_{\widehat{V}_{k,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\boldsymbol{\theta}\Bigg\}\end{aligned}$$

$$u_{k,h}(s,a;\boldsymbol{\theta},\pi,r) = \beta\left\|\boldsymbol{\phi}_{V_h(\cdot;\boldsymbol{\theta},\pi,r)}(s,a)\right\|_{\dot{\tilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}$$

Since $\dot{\hat{\boldsymbol{\Sigma}}}_{k_1,0} \preceq \dot{\hat{\boldsymbol{\Sigma}}}_{k_2,0}$ and $\dot{\tilde{\boldsymbol{\Sigma}}}_{k_1,0} \preceq \dot{\tilde{\boldsymbol{\Sigma}}}_{k_2,0}$, we have

$$\begin{aligned} u_{k_1,h}(s,a;\boldsymbol{\theta},\pi,r) &\geqslant u_{k_2,h}(s,a;\boldsymbol{\theta},\pi,r) \\ \left\|\boldsymbol{\phi}_{\widehat{V}_{k_1,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\right\|_{\dot{\hat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} &\geqslant \left\|\boldsymbol{\phi}_{\widehat{V}_{k_2,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\right\|_{\dot{\hat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} \\ \boldsymbol{\phi}^\top_{\widehat{V}_{k_1,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\boldsymbol{\theta} &\geqslant \boldsymbol{\phi}^\top_{\widehat{V}_{k_2,h+1}(\cdot;\boldsymbol{\theta},\pi,r)}(s,\pi(s))\boldsymbol{\theta}\end{aligned}$$

Thus $\widehat{V}_{k_1,h}(s;\boldsymbol{\theta},\pi,r) \geqslant \widehat{V}_{k_2,h}(s;\boldsymbol{\theta},\pi,r)$ for any $k_1 \leqslant k_2$. Furthermore, since $\mathcal{U}_{k_2} \subset \mathcal{U}_{k_1}$, and $\boldsymbol{\theta}_k,\pi_k,r_k$ are argmax over $\mathcal{U}_k$, we have

$$\widehat{V}_{k_1,1}(s_1;\boldsymbol{\theta}_{k_1},\pi_{k_1},r_{k_1}) \geqslant \widehat{V}_{k_1,1}(s_1;\boldsymbol{\theta}_{k_2},\pi_{k_2},r_{k_2}) \geqslant \widehat{V}_{k_2,1}(s_1;\boldsymbol{\theta}_{k_2},\pi_{k_2},r_{k_2})$$

It follows that $\widehat{V}_{k,1}(s_1^k;\boldsymbol{\theta}_k,\pi_k,r_k)$ is non-increasing w.r.t. k. Thus,

$$K\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K) \leqslant \sum_{k=1}^{K}\widehat{V}_{k,1}(s_1;\boldsymbol{\theta}_k,\pi_k,r_k)$$

□

2.8.7.3 Proof of Lemma 2.8.9

Lemma 2.8.24. Conditioned on the event $\mathcal{E}$, let $\widetilde{V}_{k,h}$, $\widehat{V}_{k,h}$, $\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}$, $\dot{\widehat{\mathbf{\Sigma}}}_{k,m}$, $\widetilde{\boldsymbol{\phi}}_{k,h,m}$, $\widehat{\boldsymbol{\phi}}_{k,h,m}$ be defined in Algorithm 4, for any $k \in [K]$, $h \in [H]$, $m \in \overline{[M]}$, we have

$$\widehat{V}_{k,h}(s_h^k) - u_{k,h}(s_h^k, a_h^k) - \mathbb{P}\widehat{V}_{k,h+1}(s_h^k, a_h^k) \leqslant 4 \min\left\{1, \beta \left\|\widehat{\boldsymbol{\phi}}_{k,h,0}\right\|_{\dot{\widehat{\mathbf{\Sigma}}}_{k,0}^{-1}}\right\} \tag{2.8.37}$$

$$\widetilde{V}_{k,h}(s_h^k) - r_{k,h}(s_h^k, a_h^k) - \mathbb{P}\widetilde{V}_{k,h+1} \leqslant 2 \min\left\{1, \beta \left\|\widetilde{\boldsymbol{\phi}}_{k,h,0}\right\|_{\dot{\widetilde{\mathbf{\Sigma}}}_{k,0}^{-1}}\right\} \tag{2.8.38}$$

In order to prove Lemma 2.8.9, we introduce the following quantities used in Zhou and Gu (2022a) as

$$\widehat{R}_m = \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^j \min\left\{1, \beta \|\widehat{\boldsymbol{\phi}}_{k,h,m}\|_{\dot{\widehat{\mathbf{\Sigma}}}_{k,m}^{-1}}\right\}, \forall m \in \overline{[M]} \tag{2.8.39}$$

$$\widetilde{R}_m = \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^j \min\left\{1, \beta \|\widetilde{\boldsymbol{\phi}}_{k,h,m}\|_{\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}^{-1}}\right\}, \forall m \in \overline{[M]} \tag{2.8.40}$$

$$\widehat{A}_m = \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right) - \widehat{V}_{k,h+1}^{2^m}\left(s_{h+1}^k\right)\right], \forall m \in \overline{[M]} \tag{2.8.41}$$

$$\widetilde{A}_m = \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[\left[\mathbb{P}\widetilde{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right) - \widetilde{V}_{k,h+1}^{2^m}\left(s_{h+1}^k\right)\right], \forall m \in \overline{[M]} \tag{2.8.42}$$

$$\widehat{S}_m = \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[\mathbb{V}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right), \forall m \in \overline{[M]} \tag{2.8.43}$$

$$\widetilde{S}_m = \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[\mathbb{V}\widetilde{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right), \forall m \in \overline{[M]} \tag{2.8.44}$$

$$I_h^k = \mathbb{1}\left\{\forall m \in \overline{[M]}, \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}^{-1/2}\right) / \det\left(\widehat{\mathbf{\Sigma}}_{k,h,m}^{-1/2}\right) \leqslant 4 \right.$$
$$\left. \text{and } \det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}^{-1/2}\right) / \det\left(\widetilde{\mathbf{\Sigma}}_{k,h,m}^{-1/2}\right) \leqslant 4\right\} \tag{2.8.45}$$

$$G = \sum_{k=1}^{K}\left(1 - I_H^k\right), \tag{2.8.46}$$

Lemma 2.8.25. Let γ, α, be defined in Algorithm 5, $\{\widehat{R}_m\}_{m\in\overline{[M]}}$, $\{\widetilde{R}_m\}_{m\in\overline{[M]}}$, $\{\widehat{S}_m\}_{m\in\overline{[M]}}$,

$\{\widetilde{S}_m\}_{m\in\overline{[M]}}$ be defined in (2.8.39), (2.8.40), (2.8.43), (2.8.44). Then for $m \in \overline{[M-1]}$, we have

$$\widehat{R}_m \leqslant \min\left\{KH, 4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{\widehat{S}_m + 4\widehat{R}_m + 2\widehat{R}_{m+1} + KH\alpha^2}\right\} \tag{2.8.47}$$

$$\widetilde{R}_m \leqslant \min\left\{KH, 4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{\widetilde{S}_m + 4\widetilde{R}_m + 2\widetilde{R}_{m+1} + KH\alpha^2}\right\}, \tag{2.8.48}$$

where $\iota = \log(1+KH/(d\lambda\alpha^2))$. For $\widehat{R}_{M-1}$ and $\widetilde{R}_{M-1}$, we have the trivial bound $\widehat{R}_{M-1} \leqslant KH$ and $\widetilde{R}_{M-1} \leqslant KH$.

Lemma 2.8.26. Let $\{\widehat{R}_m\}_{m\in\overline{[M]}}$, $\{\widetilde{R}_m\}_{m\in\overline{[M]}}$, $\{\widehat{S}_m\}_{m\in\overline{[M]}}$, $\{\widetilde{S}_m\}_{m\in\overline{[M]}}$, $\{\widehat{A}_m\}_{m\in\overline{[M]}}$, $\{\widetilde{A}_m\}_{m\in\overline{[M]}}$, G be defined as (2.8.39), (2.8.40), (2.8.43), (2.8.44), (2.8.41), (2.8.42), (2.8.46). Then, conditioned on the event $\mathcal{E}$, for $m \in \overline{[M-1]}$, we have

$$\widehat{S}_m \leqslant \left|\widehat{A}_{m+1}\right| + G + 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right) \tag{2.8.49}$$

$$\widetilde{S}_m \leqslant \left|\widetilde{A}_{m+1}\right| + G + 2^{m+1}\left(K + 2\widetilde{R}_0\right) \tag{2.8.50}$$

Lemma 2.8.27. Let $\{\widehat{S}_m\}_{m\in\overline{[M]}}$, $\{\widetilde{S}_m\}_{m\in\overline{[M]}}$, $\{\widehat{A}_m\}_{m\in\overline{[M]}}$, $\{\widetilde{A}_m\}_{m\in\overline{[M]}}$ be defined as (2.8.43), (2.8.44), (2.8.41), (2.8.42). Then we have $\mathbb{P}(\mathcal{E}_{2.8.27}) > 1 - 2M\delta$, with $\mathcal{E}_{2.8.27}$ be defined as,

$$\begin{aligned}\mathcal{E}_{2.8.27} := \Bigg\{\forall m \in \overline{[M]}, \left|\widehat{A}_m\right| \leqslant \min\left\{\sqrt{2\zeta\widehat{S}_m} + \zeta, KH\right\} \\ \text{and } \left|\widetilde{A}_m\right| \leqslant \min\left\{\sqrt{2\zeta\widetilde{S}_m} + \zeta, KH\right\}\Bigg\},\end{aligned} \tag{2.8.51}$$

where $\zeta = 4\log(4\log(KH)/\delta)$.

Lemma 2.8.28. Let G be defined in (2.8.46). Then we have

$$G \leqslant Md\iota, \tag{2.8.52}$$

where $\iota = \log\left(1 + KH/\left(d\lambda\alpha^2\right)\right)$.

Proof of Lemma 2.8.9. All the following proofs are conditioned on $\mathcal{E}_{2.8.6} \cap \mathcal{E}_{2.8.27}$, which hap-

pens with probability at least $1-4M\delta$. Firstly, we have

$$
\begin{aligned}
&\sum_{k=1}^{K}\widehat{V}_{k,1}(s_h^k)\\
&=\sum_{k=1}^{K}\sum_{h=1}^{H}\left[I_h^k\left[\widehat{V}_{k,h}(s_h^k)-\widehat{V}_{k,h+1}(s_{h+1}^k)\right]+\left(1-I_h^k\right)\left[\widehat{V}_{k,h}(s_h^k)-\widehat{V}_{k,h+1}(s_{h+1}^k)\right]\right]\\
&=\sum_{k=1}^{K}\left[\sum_{h=1}^{H}I_h^k u_{k,h}(s_h^k,a_h^k)+\sum_{h=1}^{H}I_h^k\left[\widehat{V}_{h,k}(s_h^k)-u_{k,h}(s_h^k,a_h^k)-\mathbb{P}\widehat{V}_{k,h+1}(s_h^k,a_h^k)\right]\right.\\
&\quad\left.+\sum_{h=1}^{H}I_h^k\left[\mathbb{P}\widehat{V}_{k,h+1}(s_h^k,a_h^k)-\widehat{V}_{k,h+1}(s_{h+1}^k)\right]\right]+\sum_{k=1}^{K}\sum_{h=1}^{H}(1-I_h^k)\left[\widehat{V}_{k,h}(s_h^k)-\widehat{V}_{k,h+1}(s_{h+1}^k)\right]\\
&\leqslant\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k u_{k,h}(s_h^k,a_h^k)}_{I_1}+\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k\left[\widehat{V}_{h,k}(s_h^k)-u_{k,h}(s_h^k,a_h^k)-\mathbb{P}\widehat{V}_{k,h+1}(s_h^k,a_h^k)\right]}_{I_2}\\
&\quad+\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k\left[\mathbb{P}\widehat{V}_{k,h+1}(s_h^k,a_h^k)-\widehat{V}_{h+1,k}(s_{h+1}^k)\right]}_{I_3}+\underbrace{\sum_{k=1}^{K}\left(1-I_{h_k}^k\right)\widehat{V}_{k,h_k}(s_{h_k}^k)}_{I_4},
\end{aligned}
$$

where h_k is the smallest index such that $I_{h_k}^k=0$. Following the definition of $u_{k,h}$,

$$
I_1=\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k\min\left\{1,\beta\left\|\widetilde{\boldsymbol{\phi}}_{k,h,0}\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\}=\widetilde{R}_0.
$$

By Lemma 2.8.24,

$$
I_2\leqslant 4\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k\min\left\{1,\beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\}=4\widehat{R}_0
$$

By definitions,

$$
\begin{aligned}
I_3&=\widehat{A}_0,\\
I_4&\leqslant\sum_{k=1}^{K}\left(1-I_H^k\right)=G.
\end{aligned}
$$

Thus,

$$
\sum_{k=1}^{K}\widehat{V}_{k,1}(s_h^k)\leqslant\widetilde{R}_0+4\widehat{R}_0+\widehat{A}_0+G \tag{2.8.53}
$$

Substituting (2.8.49) in Lemma 2.8.26 into (2.8.47) in Lemma 2.8.25, we have

$$\begin{aligned}\widehat{R}_m &\leqslant 4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{\left|\widehat{A}_{m+1}\right| + G + 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right) + 4\widehat{R}_m + 2\widehat{R}_{m+1} + KH\alpha^2} \\ &\leqslant 2\beta\sqrt{d\iota}\sqrt{\left|\widehat{A}_{m+1}\right| + 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right) + 4\widehat{R}_m + 2\widehat{R}_{m+1}} \\ &\quad + \underbrace{4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{G + KH\alpha^2}}_{I_c}, \end{aligned} \tag{2.8.54}$$

where the second inequality holds due to $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$. Substituting (2.8.49) in Lemma 2.8.26 into (2.8.51) in Lemma 2.8.27, we have

$$\begin{aligned}\left|\widehat{A}_m\right| &\leqslant \sqrt{2\zeta}\sqrt{\left|\widehat{A}_{m+1}\right| + G + 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right)} + \zeta \\ &\leqslant \sqrt{2\zeta}\sqrt{\left|\widehat{A}_{m+1}\right| + 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right)} + \sqrt{2\zeta G} + \zeta \end{aligned} \tag{2.8.55}$$

Substituting (2.8.50) in Lemma 2.8.26 into (2.8.48) in Lemma 2.8.25, we have

$$\begin{aligned}\widetilde{R}_m &\leqslant 4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{\left|\widetilde{A}_{m+1}\right| + G + 2^{m+1}\left(K + 2\widetilde{R}_0\right) + 4\widetilde{R}_m + 2\widetilde{R}_{m+1} + KH\alpha^2} \\ &\leqslant 2\beta\sqrt{d\iota}\sqrt{\left|\widetilde{A}_{m+1}\right| + 2^{m+1}\left(K + 2\widetilde{R}_0\right) + 4\widetilde{R}_m + 2\widetilde{R}_{m+1}} \\ &\quad + \underbrace{4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{G + KH\alpha^2}}_{I_c} \end{aligned} \tag{2.8.56}$$

Substituting (2.8.50) in Lemma 2.8.26 into (2.8.51) in Lemma 2.8.27, we have

$$\begin{aligned}\left|\widetilde{A}_m\right| &\leqslant \sqrt{2\zeta}\sqrt{\left|\widetilde{A}_{m+1}\right| + G + 2^{m+1}\left(K + 2\widetilde{R}_0\right)} + \zeta \\ &\leqslant \sqrt{2\zeta}\sqrt{\left|\widetilde{A}_{m+1}\right| + 2^{m+1}\left(K + 2\widetilde{R}_0\right)} + \sqrt{2\zeta G} + \zeta \end{aligned} \tag{2.8.57}$$

Thus, calculating (2.8.56) + (2.8.57) + 4×(2.8.54) + (2.8.55) and using $\sqrt{a}+\sqrt{b}+\sqrt{c}+\sqrt{d} \leqslant$

$2\sqrt{a+b+c+d}$, we have

$$\begin{aligned}
&\widetilde{R}_m + \left|\widetilde{A}_m\right| + 4\widehat{R}_m + \left|\widehat{A}_m\right| \\
&\leqslant 5I_c + 2\sqrt{2\zeta G} + 2\zeta + 2\max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{2\left|\widehat{A}_{m+1}\right| + 2\cdot 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right)} \\
&\quad \overline{+4\widehat{R}_m + 2\widehat{R}_{m+1} + 2\left|\widetilde{A}_{m+1}\right| + 2\cdot 2^{m+1}\left(K + 2\widetilde{R}_0\right) + 4\widetilde{R}_m + 2\widetilde{R}_{m+1}} \\
&\leqslant 5I_c + +2\sqrt{2\zeta G} + 2\zeta + 4\max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{\left(\widetilde{R}_m + \left|\widetilde{A}_m\right| + 4\widehat{R}_m + \left|\widehat{A}_m\right|\right)} \\
&\quad \overline{+\left(\widetilde{R}_{m+1} + \left|\widetilde{A}_{m+1}\right| + 4\widehat{R}_{m+1} + \left|\widehat{A}_{m+1}\right|\right) + 2\cdot 2^{m+1}\left(K + \widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right|\right)}.
\end{aligned}$$

Then by Lemma 2.8.33 with $a_m = \widetilde{R}_m + \left|\widetilde{A}_m\right| + 4\widehat{R}_m + \left|\widehat{A}_m\right| \leqslant 7KH$ and $M = \log(7KH)/\log 2$, $\widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right|$ can be bounded as

$$\begin{aligned}
&\widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right| \\
&\leqslant 22\cdot 16\max\{64\beta^2 d\iota, 2\zeta\} + 30I_c + 12\sqrt{\zeta G} + 12\zeta \\
&\qquad + 32\max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{K + \widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right|} \\
&\leqslant 352\max\left\{64\beta^2 d\iota, 2\zeta\right\} + 30I_c + 12\sqrt{\zeta G} + 12\zeta + 32\max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{K} \\
&\qquad + 32\max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{\widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right|}. \qquad (2.8.58)
\end{aligned}$$

By the fact that $x \leqslant a\sqrt{x} + b \Rightarrow x \leqslant 2a^2 + 2b$, (2.8.58) implies that

$$\begin{aligned}
&\widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right| \\
&\leqslant 896\max\left\{64\beta^2 d\iota, 2\zeta\right\} + 60I_c + 24\sqrt{\zeta G} + 24\zeta + 64\max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{K}. \qquad (2.8.59)
\end{aligned}$$

Finally, plugging (2.8.59) into (2.8.53) and bounding G with Lemma 2.8.28, we have

$$\begin{aligned}
&\sum_{k=1}^{K} \widehat{V}_{k,1}(s_h^k) \\
&\leqslant \widetilde{R}_0 + \left|\widetilde{A}_0\right| + 4\widehat{R}_0 + \left|\widehat{A}_0\right| + G \\
&\leqslant 896 \max\left\{64\beta^2 d\iota, 2\zeta\right\} + 24\zeta + 64 \max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\}\sqrt{K} \\
&\quad + 60\left(4d\iota + 4\beta\gamma^2 d\iota + 2\beta\sqrt{d\iota}\sqrt{Md\iota + KH\alpha^2}\right) + 24\sqrt{\zeta M d\iota} + Md\iota \\
&\leqslant 896 \max\left\{64\beta^2 d\iota, 2\zeta\right\} + 24\zeta + 240d\iota + 240\beta\gamma^2 d\iota + 120\beta d\iota\sqrt{M} + 24\sqrt{\zeta M d\iota} + Md\iota \\
&\quad + \left(64 \max\left\{8\beta\sqrt{d\iota}, \sqrt{2\zeta}\right\} + 120\beta\sqrt{d\iota H\alpha^2}\right)\sqrt{K}
\end{aligned} \tag{2.8.60}$$

□

2.8.8 Proof of Lemmas in Section 2.8.7

2.8.8.1 Proof of Lemma 2.8.6

Lemma 2.8.29 (Theorem 4.3, Zhou and Gu (2022a))**.** Let $\{\mathcal{G}_k\}_{k=1}^{\infty}$ be a filtration, and $\{\mathrm{x}_k, \eta_k\}_{k\geqslant 1}$ be a stochastic process such that $\mathrm{x}_k \in \mathbb{R}^d$ is $\mathcal{G}_k$-measurable and $\eta_k \in \mathbb{R}$ is $\mathcal{G}_{k+1}$-measurable. Let $L, \sigma, \lambda, \varepsilon > 0, \boldsymbol{\mu}^* \in \mathbb{R}^d$. For $k \geqslant 1$, let $y_k = \langle \boldsymbol{\mu}^*, \mathbf{x}_k \rangle + \eta_k$ and suppose that $\eta_k, \mathbf{x}_k$ also satisfy

$$\mathbb{E}\left[\eta_k \mid \mathcal{G}_k\right] = 0, \mathbb{E}\left[\eta_k^2 \mid \mathcal{G}_k\right] \leqslant \sigma^2, |\eta_k| \leqslant R, \|\mathrm{x}_k\|_2 \leqslant L \tag{2.8.61}$$

For $k \geqslant 1$, let $\mathbf{Z}_k = \lambda\mathbf{I} + \sum_{i=1}^{k} \mathbf{x}_i\mathbf{x}_i^\top, \mathbf{b}_k = \sum_{i=1}^{k} y_i\mathbf{x}_i, \boldsymbol{\mu}_k = \mathbf{Z}_k^{-1}\mathbf{b}_k$, and

$$\begin{aligned}
\beta_k =& 12\sqrt{\sigma^2 d \log\left(1 + kL^2/(d\lambda)\right) \log\left(32(\log(R/\varepsilon) + 1)k^2/\delta\right)} + 6\log\left(32(\log(R/\varepsilon) + 1)k^2/\delta\right)\varepsilon \\
&+ 24\log\left(32(\log(R/\varepsilon) + 1)k^2/\delta\right) \max_{1\leqslant i\leqslant k}\left\{|\eta_i| \min\left\{1, \|\mathbf{x}_i\|_{\mathbf{Z}_{i-1}^{-1}}\right\}\right\}.
\end{aligned} \tag{2.8.62}$$

Then, for any $0 < \delta < 1$, we have with probability at least $1 - \delta$ that,

$$\forall k \geqslant 1, \left\|\sum_{i=1}^{k} \mathbf{x}_i\eta_i\right\|_{\mathbf{Z}_k^{-1}} \leqslant \beta_k, \quad \|\boldsymbol{\mu}_k - \boldsymbol{\mu}^*\|_{\mathbf{Z}_k} \leqslant \beta_k + \sqrt{\lambda}\,\|\boldsymbol{\mu}^*\|_2$$

Lemma 2.8.30. Let $\widetilde{V}_{k,h}$, $\widehat{V}_{k,h}$, $\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m}$, $\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}$, $\widetilde{\boldsymbol{\theta}}_{k,m}$, $\widehat{\boldsymbol{\theta}}_{k,m}$, $\widetilde{\boldsymbol{\phi}}_{k,h,m}$, $\widehat{\boldsymbol{\phi}}_{k,h,m}$ be defined in Algorithm 4, for any $k \in [K]$, $h \in [H]$, $m \in \overline{[M]}$. We have

$$\begin{aligned}
&\left|\mathbb{V}\widehat{V}_{k,h+1}^{2^m}\left(s_h^k, a_h^k\right) - \widehat{\mathbb{V}}\widehat{V}_{k,h+1}^{2^m}\left(s_h^k, a_h^k\right)\right| \\
&\leqslant \min\left\{1, \left\|\widehat{\boldsymbol{\phi}}_{k,h,m+1}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}^{-1}} \left\|\widehat{\boldsymbol{\theta}}_{k,m+1} - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}}\right\} \\
&\quad + \min\left\{1, 2\left\|\widehat{\boldsymbol{\phi}}_{k,h,m}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}^{-1}} \left\|\widehat{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}}\right\},
\end{aligned} \tag{2.8.63}$$

and

$$\begin{aligned}
&\left|\mathbb{V}\widetilde{V}_{k,h+1}^{2^m}\left(s_h^k, a_h^k\right) - \widetilde{\mathbb{V}}\widetilde{V}_{k,h+1}^{2^m}\left(s_h^k, a_h^k\right)\right| \\
&\leqslant \min\left\{1, \left\|\widetilde{\boldsymbol{\phi}}_{k,h,m+1}\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m+1}^{-1}} \left\|\widetilde{\boldsymbol{\theta}}_{k,m+1} - \boldsymbol{\theta}^*\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m+1}}\right\} \\
&\quad + \min\left\{1, 2\left\|\widetilde{\boldsymbol{\phi}}_{k,h,m}\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m}^{-1}} \left\|\widetilde{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,m}}\right\}.
\end{aligned} \tag{2.8.64}$$

Proof of Lemma 2.8.30. The proof follows the proof of Lemma C.1 in Zhou et al. (2021b). We first prove (2.8.63), and the proof of (2.8.64) is similar. We have

$$\begin{aligned}
&|[\widehat{\mathbb{V}}_{k,h}\widehat{V}_{k,h+1}^{2^m}](s_h^k, a_h^k) - [\mathbb{V}_{k,h}\widehat{V}_{k,h+1}](s_h^k, a_h^k)| \\
&= |[\langle\widehat{\boldsymbol{\phi}}_{k,h,m+1}, \widehat{\boldsymbol{\theta}}_{k,m+1}\rangle]_{[0,1]} - \langle\widehat{\boldsymbol{\phi}}_{k,h,m+1}, \boldsymbol{\theta}^*\rangle \\
&\qquad + (\langle\widehat{\boldsymbol{\phi}}_{k,h,m}, \boldsymbol{\theta}^*\rangle)^2 - [\langle\widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}\rangle]_{[0,1]}^2| \\
&\leqslant \underbrace{|[\langle\widehat{\boldsymbol{\phi}}_{k,h,m+1}, \widehat{\boldsymbol{\theta}}_{k,m+1}\rangle]_{[0,1]} - \langle\widehat{\boldsymbol{\phi}}_{k,h,m+1}, \boldsymbol{\theta}^*\rangle|}_{I_1} \\
&\qquad + \underbrace{|(\langle\widehat{\boldsymbol{\phi}}_{k,h,m}, \boldsymbol{\theta}^*\rangle)^2 - [\langle\widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}\rangle]_{[0,1]}^2|}_{I_2}
\end{aligned} \tag{2.8.65}$$

where the inequality holds due to triangle inequality. We have $I_1 \leqslant 1$ since both terms in I_1

lie in the interval $[0,1]$. Furthermore,

$$\begin{aligned} I_1 &\leqslant |[\langle \widehat{\boldsymbol{\phi}}_{k,h,m+1}, \widehat{\boldsymbol{\theta}}_{k,m+1}\rangle] - \langle \widehat{\boldsymbol{\phi}}_{k,h,m+1}, \boldsymbol{\theta}^*\rangle| \\ &= |[\langle \widehat{\boldsymbol{\phi}}_{k,h,m+1}, \widehat{\boldsymbol{\theta}}_{k,m+1} - \boldsymbol{\theta}^*\rangle| \\ &\leqslant \|\boldsymbol{\phi}_{k,h,m+1}\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}^{-1}} \|\widehat{\boldsymbol{\theta}}_{k,m+1} - \boldsymbol{\theta}^*\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}}, \end{aligned}$$

where the first inequality holds due to $\langle \widehat{\boldsymbol{\phi}}_{k,h,m+1}(s_h^k, a_h^k), \boldsymbol{\theta}^*\rangle \in [0,1]$, the second inequality holds due to Cauchy-Schwarz inequality. Thus, we obtain

$$I_1 \leqslant \min\{1, \|\boldsymbol{\phi}_{k,h,m+1}\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}^{-1}} \|\widehat{\boldsymbol{\theta}}_{k,m+1} - \boldsymbol{\theta}^*\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}}\} \tag{2.8.66}$$

For I_2, we have

$$\begin{aligned} I_2 &= \left|(\langle \widehat{\boldsymbol{\phi}}_{k,h,m}(s_h^k, a_h^k), \boldsymbol{\theta}^*\rangle) - [\langle \widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}\rangle]_{[0,1]}\right| \\ &\quad \times \left|(\langle \widehat{\boldsymbol{\phi}}_{k,h,m}(s_h^k, a_h^k), \boldsymbol{\theta}^*\rangle) + [\langle \widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}\rangle]_{[0,1]}\right| \\ &\leqslant 2\left|(\langle \widehat{\boldsymbol{\phi}}_{k,h,m}(s_h^k, a_h^k), \boldsymbol{\theta}^*\rangle) - \langle \widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}, \rangle\right| \\ &\leqslant 2\|\boldsymbol{\phi}_{k,h,m}(s_h^k, a_h^k)\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}^{-1}} \|\widehat{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}} \end{aligned}$$

where the first inequality holds due to that both $\langle \widehat{\boldsymbol{\phi}}_{k,h,m}(s_h^k, a_h^k), \boldsymbol{\theta}^*\rangle$ and $[\langle \widehat{\boldsymbol{\phi}}_{k,h,m}, \widehat{\boldsymbol{\theta}}_{k,m}\rangle]_{[0,1]}$ lie in the interval $[0,1]$, and the second inequality holds due to Cauchy-Schwarz inequality. Since I_2 belongs to the interval $[0,1]$, we have

$$I_2 \leqslant \min\{1, 2\|\boldsymbol{\phi}_{k,h,m}(s_h^k, a_h^k)\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}^{-1}} \|\widehat{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}}\} \tag{2.8.67}$$

Substituting (2.8.66) and (2.8.67) into (2.8.65), we obtain (2.8.63). The proof of 2.8.64 is nearly identical to the proof of (2.8.65). The only difference is to replace $\widehat{\boldsymbol{\phi}}$ with $\widetilde{\boldsymbol{\phi}}$, $\widehat{\boldsymbol{\theta}}$ with $\widetilde{\boldsymbol{\theta}}$, $\dot{\widehat{\boldsymbol{\Sigma}}}$ with $\dot{\widetilde{\boldsymbol{\Sigma}}}$. □

Proof of Lemma 2.8.6. The proof follows Lemma C.2 in Zhou and Gu (2022a). Symbols we used here may have small intuitively understandable modification compared to Algorithm 5

since we have to distinguish between Algorithm 5 applied to $\widetilde{V}_{k,h}$ and $\widehat{V}_{k,h}$. We first prove that Equation (2.8.9) holds with high probability. By definitions,

$$\widehat{\sigma}^2_{k,h,m} = \max\left\{\gamma^2 \left\|\widehat{\boldsymbol{\phi}}_{k,h,m}\right\|_{\widehat{\boldsymbol{\Sigma}}^{-1}_{k,h,m}}, \left[\widehat{\mathbb{V}}_{k,m}\widehat{V}^{2^m}_{k,h+1}\right](s^k_h, a^k_h) + \widehat{E}_{k,h,m}, \alpha^2\right\}$$
$$\widehat{\sigma}^2_{k,h,m} = \max\left\{\gamma^2 \left\|\widehat{\boldsymbol{\phi}}_{k,h,M-1}\right\|_{\widehat{\boldsymbol{\Sigma}}^{-1}_{k,h,M-1}}, 1, \alpha^2\right\}.$$

We define $\mathcal{C}_{k,m}$ as

$$\widehat{\mathcal{C}}_{k,m} := \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_{k,m}\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}} \leqslant \beta_k\}.$$

For each m, let

$$\begin{aligned}
\mathbf{x}_{k,h,m} &= \widehat{\sigma}^{-1}_{k,h,m}\widehat{\boldsymbol{\phi}}_{k,h,m},\\
\eta_{k,h,m} &= \widehat{\sigma}^{-1}_{k,h,m}\,\mathbb{1}\{\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k,m} \cap \widehat{\mathcal{C}}_{k,m+1}\}[\widehat{V}^{2^m}_{k,h+1}(s^k_{h+1}) - \langle\widehat{\boldsymbol{\phi}}_{k,h,m}, \boldsymbol{\theta}^*\rangle],\\
\eta_{k,h,M-1} &= \widehat{\sigma}^{-1}_{k,h,M-1}[\widehat{V}^{2^{M-1}}_{k,h+1} - \langle\widehat{\boldsymbol{\phi}}_{k,h,M-1}, \boldsymbol{\theta}^*\rangle],\\
\mathcal{G}_{k,h} &= \mathcal{F}_{k,h},\\
\boldsymbol{\mu}^* &= \boldsymbol{\theta}^*.
\end{aligned}$$

We have

$$\mathbb{E}[\eta_{k,h,m}|\mathcal{G}_{k,h}] = 0, \quad \|\mathbf{x}_{k,h,m}\|_2 \leqslant \widehat{\sigma}^{-1}_{k,h,m} \leqslant 1/\alpha, \quad |\eta_{k,h,m}| \leqslant 1/\alpha$$

Since $\mathbb{1}\{\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k,m} \cap \widehat{\mathcal{C}}_{k,m+1}\}$ is $\mathcal{G}_{k,h}$-measurable, then we can bound the variance for $m \in \overline{[M]}$

as follows:

$$\begin{aligned}
\mathbb{E}[\eta_{k,h,m}^2|\mathcal{G}_{k,h}] &= \widehat{\sigma}_{k,h,m}^{-2}\,\mathbb{1}\{\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k,m} \cap \widehat{\mathcal{C}}_{k,m+1}\}[\mathbb{V}\widehat{V}_{k,h+1}^{2^m}](s_h^k, a_h^k) \\
&\leqslant \widehat{\sigma}_{k,h,m}^{-2}\,\mathbb{1}\{\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k,m} \cap \widehat{\mathcal{C}}_{k,m+1}\}\Bigg[\widehat{\mathbb{V}}\widehat{V}_{k,h+1}^{2^m}\left(s_h^k, a_h^k\right) \\
&\qquad + \min\left\{1, \left\|\widehat{\boldsymbol{\phi}}_{k,h,m+1}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}^{-1}} \left\|\widehat{\boldsymbol{\theta}}_{k,m+1} - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}}\right\} \\
&\qquad + \min\left\{1, 2\left\|\widehat{\boldsymbol{\phi}}_{k,h,m}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}^{-1}} \left\|\widehat{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}}\right\}\Bigg] \\
&\leqslant \widehat{\sigma}_{k,h,m}^{-2}\Bigg[\widehat{\mathbb{V}}\widehat{V}_{k,h+1}^{2^m}\left(s_h^k, a_h^k\right) + \min\left\{1, \beta_k\left\|\widehat{\boldsymbol{\phi}}_{k,h,m+1}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m+1}^{-1}}\right\} \\
&\qquad + \min\left\{1, 2\beta_k\left\|\widehat{\boldsymbol{\phi}}_{k,h,m}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}^{-1}}\right\}\Bigg] \\
&\leqslant 1,
\end{aligned}$$

where the first inequality holds due to Lemma 2.8.30, the second inequality holds due to the definition of the indicator function, and the third inequality holds due to the definition of $\widehat{\sigma}_{k,h,m}^{-2}$. For $m = M-1$, we have $\mathbb{E}[\eta_{k,h,m}^2|\mathcal{G}_{k,h}] \leqslant 1$ directly by the definition of $\widehat{\sigma}_{k,h,m}^2$. For any $m \in \overline{[M]}$, we have

$$|\eta_{k,h,m}|\max\{1, \|\mathbf{x}_{k,h,m}\|_{\widehat{\boldsymbol{\Sigma}}_{k,h-1,m}^{-1}}\} \leqslant \widehat{\sigma}_{k,h,m}^{-2}\|\widehat{\boldsymbol{\phi}}_{k,h,m}\|_{\widehat{\boldsymbol{\Sigma}}_{k,h-1,m}^{-1}} \leqslant 1/\gamma^2,$$

where the first inequality follows from the definition of $\eta_{k,h,m}$ and $\mathbf{x}_{k,h,m}$, and the second inequality follows from the definition of $\widehat{\sigma}_{k,h,m}$. Let

$$\begin{aligned}
y_{k,h,m} &= \langle \boldsymbol{\mu}^*, \mathbf{x}_{x,h,m}\rangle + \eta_{k,h,m}, \\
\mathbf{Z}_{k,m} &= \lambda\mathbf{I} + \sum_{i=1}^{k}\sum_{h=1}^{H}\mathbf{x}_{i,h,m}\mathbf{x}_{i,h,m}^\top = \dot{\widehat{\boldsymbol{\Sigma}}}_{k+1,m}, \\
\boldsymbol{b}_{k,m} &= \sum_{i=1}^{k}\sum_{h=1}^{H}\mathbf{x}_{i,h,m}y_{i,h,m}, \\
\boldsymbol{\mu}_{k,m} &= \mathbf{Z}_{k,m}^{-1}\boldsymbol{b}_{k,m}, \\
\varepsilon &= 1/\gamma^2.
\end{aligned}$$

Then, by Lemma 2.8.29, for each $m \in \overline{[M]}$, with probability at least $1-\delta$, $\forall k \in [K+1]$,

$$\begin{aligned}\|\boldsymbol{\mu}_{k-1,m} - \boldsymbol{\theta}^*\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,m}} \leqslant & 12\sqrt{d\log(1+kH/(\alpha^2 d\lambda))\log(32(\log(\gamma^2/\alpha)+1)k^2H^2/\delta)} \\ & + 30\log(32(\log(\gamma^2/\alpha)+1)k^2H^2/\delta)/\gamma^2 + \sqrt{\lambda}B \\ = & \beta_k \end{aligned} \tag{2.8.68}$$

Define the event that (2.8.68) happens for all k and m as $\widehat{\mathcal{E}}$. Conditioned on $\widehat{\mathcal{E}}$, the following properties hold:

- For $k=1$, $m \in \overline{[M]}$, by the definition of $\widehat{\boldsymbol{\theta}}_{1,m}$ and $\dot{\widehat{\boldsymbol{\Sigma}}}_{1,m}$, we have $\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_{1,m}\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{1,m}} = \|\boldsymbol{\theta}^*\|_{\lambda\mathbf{I}} \leqslant \sqrt{\lambda}B = \beta_1$, which implies

$$\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{1,m} \tag{2.8.69}$$

- For $k \in [K]$ and $m = M-1$, we directly have $\boldsymbol{\mu}_{k,M-1} = \widehat{\boldsymbol{\theta}}_{k+1,M-1}$, which implies

$$\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k+1,M-1}. \tag{2.8.70}$$

- For $k \in [K]$ and $m \in \overline{[M-1]}$, we have

$$\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k,m} \cap \widehat{\mathcal{C}}_{k,m+1} \Rightarrow y_{k,h,m} = \widehat{\sigma}^{-1}\widehat{V}_{k,h+1}^{2^m}(s_{h+1}^k) \Rightarrow \boldsymbol{\mu}_{k,m} = \widehat{\boldsymbol{\theta}}_{k+1,m} \Rightarrow \boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k+1,m}. \tag{2.8.71}$$

Therefore, by induction based on initial conditions (2.8.69) and (2.8.70), induction rule (2.8.71), we have for $k \in [K]$ and $m \in [M]$, $\boldsymbol{\theta}^* \in \widehat{\mathcal{C}}_{k,m}$. Taking the union bound gives that (2.8.9) happens with probability at least $1 - M\delta$. We can use the nearly identical argument to prove that (2.8.10) holds with probability at least $1-M\delta$. The only difference is to replace $\widehat{\sigma}$ with $\widetilde{\sigma}$, $\widehat{\boldsymbol{\phi}}$ with $\widetilde{\boldsymbol{\phi}}$, $\widehat{\mathbb{V}}$ with $\widetilde{\mathbb{V}}$, $\widehat{V}$ with $\widetilde{\mathbb{V}}$, $\widehat{\boldsymbol{\Sigma}}$ with $\widetilde{\boldsymbol{\Sigma}}$, $\dot{\widehat{\boldsymbol{\Sigma}}}$ with $\dot{\widetilde{\boldsymbol{\Sigma}}}$, $\widehat{\boldsymbol{\theta}}$ with $\widetilde{\boldsymbol{\theta}}$. By taking the union bound, we obtain that with probability at least $1-2M\delta$, Equations (2.8.9) (2.8.10) both hold. For (2.8.11) and (2.8.12), we have

$$\begin{aligned}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}} &\leqslant \left\|\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_{k,m}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}} + \left\|\widehat{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}} \leqslant 2\beta_k, \\ \|\boldsymbol{\theta}_k - \boldsymbol{\theta}^*\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}} &\leqslant \left\|\boldsymbol{\theta}_k - \widetilde{\boldsymbol{\theta}}_{k,m}\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}} + \left\|\widetilde{\boldsymbol{\theta}}_{k,m} - \boldsymbol{\theta}^*\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}} \leqslant 2\beta_k \end{aligned}$$

□

2.8.8.2 Proof of Lemma 2.8.22

Proof of Lemma 2.8.22. We prove this inequality by induction. Suppose

$$V_{h+1}(s_{h+1};\boldsymbol{\theta}_K,\pi,r)-V_{h+1}(s_{h+1};\boldsymbol{\theta}^*,\pi,r) \\ = \mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_{h+1}} W_{h+1}(\{(\mathbb{P}_K-\mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\}), \quad (2.8.72)$$

which is true for $h=H$. Then, we have

$$\begin{aligned}
&V_h(s_h;\boldsymbol{\theta}_K,\pi,r)-V_h(s_h;\boldsymbol{\theta}^*,\pi,r)\\
&=\min\left\{1, r_h(s_h,a_h)+\mathbb{P}_K V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)-\left(r_h(s_h,a_h)+\mathbb{P}V_{h+1}(s_h,a_h;\boldsymbol{\theta}^*,\pi,r)\right)\right\}\\
&=\min\left\{1, \mathbb{P}_K V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)-\mathbb{P}V_{h+1}(s_h,a_h;\boldsymbol{\theta}^*,\pi,r)\right\}\\
&=\min\{1,(\mathbb{P}_K-\mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)+\mathbb{P}(V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)-V_{h+1}(s_h,a_h;\boldsymbol{\theta}^*,\pi,r))\}\\
&=\min\{1,(\mathbb{P}_K-\mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\\
&\quad+\mathbb{E}_{s_{h+1}\sim\mathbb{P}(\cdot|s_h,a_h)}\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_{h+1}} W_{h+1}(\{(\mathbb{P}_K-\mathbb{P})V_{h+1}(s_h;\boldsymbol{\theta}_K,\pi,r)\})\}\\
&=\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_h}\min\{1,(\mathbb{P}_K-\mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\\
&\quad+W_{h+1}(\{(\mathbb{P}_k-\mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\})\}\\
&=\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_h} W_h(\{(\mathbb{P}_k-\mathbb{P})V_{h+1}(s_h,a_h;\boldsymbol{\theta}_K,\pi,r)\}).
\end{aligned}$$

The first equality holds due to that $V_h(s_h;\boldsymbol{\theta}_K,\pi,r)$ and $V_h(s_h;\boldsymbol{\theta}^*,\pi,r)$ both belong to $[0,1]$, the third equality holds due to (2.8.72), and the forth equality holds due to that $\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_h}=\mathbb{E}_{s_{h+1}\sim\mathbb{P}(\cdot|s_h,a_h)}\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_{h+1}}$. Thus, by induction, we obtain the desired result (2.8.34). □

2.8.8.3 Proof of Lemma 2.8.23

Proof of Lemma 2.8.23. We first prove (2.8.73) by induction.

$$\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_1} W_1\left(\{u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r)\}\right) \leqslant \widehat{V}_{K,1}\left(s_1;\boldsymbol{\theta}_K,\pi,r\right). \quad (2.8.73)$$

Suppose

$$\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_{h+1}} W_{h+1}(\{u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r)\}) \leqslant \widehat{V}_{K,h+1}(s_1;\boldsymbol{\theta}_K,\pi,r), \tag{2.8.74}$$

which is true for $h = H$. Then,

$$\begin{aligned}
&\widehat{V}_{K,h}(s_h;\boldsymbol{\theta}_K,\pi,r) - \mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_h} W_h(\{u_{K,h}(s_h,\pi(a_h);\boldsymbol{\theta}_K,\pi,r)\}) \\
&\geqslant \min\Bigg\{0, u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r) + 2\beta\left\|\boldsymbol{\phi}_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{K,0}} \\
&\qquad + \boldsymbol{\phi}^\top_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\,\boldsymbol{\theta}_K - \mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_h} W_h(\{u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r))\}\Bigg\} \\
&\geqslant \min\Bigg\{0, u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r) + 2\beta\left\|\boldsymbol{\phi}_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{K,0}} \\
&\qquad + \boldsymbol{\phi}^\top_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\boldsymbol{\theta}_K - u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r) \\
&\qquad - \mathbb{E}_{s_{h+1}\sim\mathbb{P}(\cdot|s_h,\pi(s_h))}\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_{h+1}} W_{h+1}(\{u_{K,h}(s_h,\pi(s_h);\boldsymbol{\theta}_K,\pi,r))\Bigg\} \\
&\geqslant \min\Bigg\{0, 2\beta\left\|\boldsymbol{\phi}_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{K,0}} + \boldsymbol{\phi}^\top_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\boldsymbol{\theta}_K \\
&\qquad - \mathbb{E}_{s_{h+1}\sim\mathbb{P}(\cdot|s_h,\pi(s_h))}\widehat{V}_{K,h+1}(s_{h+1};\boldsymbol{\theta}_K,\pi,r)\Bigg\} \\
&\geqslant \min\Bigg\{0, 2\beta\left\|\boldsymbol{\phi}_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{K,0}} + \boldsymbol{\phi}^\top_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))(\boldsymbol{\theta}_K - \boldsymbol{\theta}^*)\Bigg\} \\
&\geqslant \min\Bigg\{0, 2\beta\left\|\boldsymbol{\phi}_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{K,0}} - 2\beta\left\|\boldsymbol{\phi}_{\widehat{V}_{K,h+1}(\cdot;\boldsymbol{\theta}_K,\pi,r)}(s_h,\pi(s_h))\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{K,0}}\Bigg\} \\
&\geqslant 0,
\end{aligned}$$

where the first inequality holds due to the definition of $\widehat{V}_{K,h}$, the second inequality holds due to the definition of $W_h(\cdot)$ and $\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_h} = \mathbb{E}_{s_{h+1}\sim\mathbb{P}(\cdot|s_h,\pi(s_h))}\mathbb{E}_{\mathtt{traj}\sim(\pi,\mathbb{P})|\mathtt{traj}_{h+1}}$, the third inequality holds due to 2.8.74, the fifth inequality holds due to Lemma 2.8.6. Thus, by induction, 2.8.73 holds. Thanks to the optimism of $\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K)$, we have

$$\widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi,r) \leqslant \widehat{V}_{K,1}(s_1;\boldsymbol{\theta}_K,\pi_K,r_K),$$

which concludes the proof. □

2.8.8.4 Proof of Lemma 2.8.24

Proof of Lemma 2.8.24. For the equation (2.8.37), we have

$$
\begin{aligned}
&\widehat{V}_{k,h}(s_h^k) - u_{k,h}(s_h^k, a_h^k) - \mathbb{P}\widehat{V}_{k,h+1}(s_h^k, a_h^k) \\
&\leqslant \min\left\{1, 2\beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}(s_h^k, a_h^k)\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} + \widehat{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\boldsymbol{\theta}_k - \widehat{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\boldsymbol{\theta}\right\} \\
&= \min\left\{1, 2\beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}(s_h^k, a_h^k)\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} + \widehat{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)(\boldsymbol{\theta}_k - \boldsymbol{\theta})\right\} \\
&\leqslant \min\left\{1, 2\beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}(s_h^k, a_h^k)\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}} + \left\|\widehat{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}}\right\} \\
&\leqslant \min\left\{1, 4\beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}(s_h^k, a_h^k)\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\} \\
&\leqslant 4\min\left\{1, \beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}(s_h^k, a_h^k)\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\}
\end{aligned}
$$

where the first inequality holds due to that each term lies in the interval $[0, 1]$, the second inequality holds due to Cauchy-Schwartz inequality, and the third inequality holds due to lemma 2.8.6. For the equation (2.8.38), we have

$$
\begin{aligned}
&\widetilde{V}_{k,h}(s_h^k) - r_{k,h}(s_h^k, a_h^k) - \mathbb{P}\widetilde{V}_{k,h+1}(s_h^k, a_h^k) \\
&\leqslant \min\left\{1, \widetilde{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\boldsymbol{\theta}_k - \widetilde{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\boldsymbol{\theta}\right\} \\
&= \min\left\{1, \widetilde{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)(\boldsymbol{\theta}_k - \boldsymbol{\theta})\right\} \\
&\leqslant \min\left\{1, \left\|\widetilde{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\|\boldsymbol{\theta}_k - \boldsymbol{\theta}\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}}\right\} \\
&\leqslant \min\left\{1, 2\beta\left\|\widetilde{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\} \\
&\leqslant 2\min\left\{1, \beta\left\|\widetilde{\boldsymbol{\phi}}_{k,h,0}^{\top}(s_h^k, a_h^k)\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\},
\end{aligned}
$$

where the first inequality holds due to that each term lies in the interval $[0, 1]$, the second inequality holds due to the Cauchy-Schwartz inequality, and the third inequality holds due to Lemma 2.8.6. □

2.8.8.5 Proof of Lemma 2.8.25

Lemma 2.8.31 (Lemma B.1, Zhou and Gu (2022a))**.** Let $\{\sigma_k, \beta_k\}_{k\geqslant 1}$ be a sequence of non-negative numbers, $\alpha, \gamma > 0, \{\mathbf{x}_k\}_{k\geqslant 1} \subset \mathbb{R}^d$ and $\|\mathbf{x}_k\|_2 \leqslant L$. Let $\{\mathbf{Z}_k\}_{k\geqslant 1}$ and $\{\bar{\sigma}_k\}_{k\geqslant 1}$ be recursively defined as follows: $\mathbf{Z}_1 = \lambda\mathbf{I}$

$$\forall k \geqslant 1, \bar{\sigma}_k = \max\left\{\sigma_k, \alpha, \gamma \left\|\mathbf{x}_k\right\|_{\mathbf{z}_k^{-1}}^{1/2}\right\}, \mathbf{Z}_{k+1} = \mathbf{Z}_k + \mathbf{x}_k\mathbf{x}_k^\top/\bar{\sigma}_k^2.$$

Let $\iota = \log\left(1 + KL^2/\left(d\lambda\alpha^2\right)\right)$. Then we have

$$\sum_{k=1}^{K} \min\left\{1, \beta_k \left\|\mathbf{x}_k\right\|_{\mathbf{Z}_k^{-1}}\right\} \leqslant 2d\iota + 2\max_{k\in[K]} \beta_k\gamma^2 d\iota + 2\sqrt{d\iota}\sqrt{\sum_{k=1}^{K} \beta_k^2\left(\sigma_k^2 + \alpha^2\right)}.$$

Proof of Lemma 2.8.25. The proof is nearly identical to the proof of Lemma C.5 in Zhou and Gu (2022a). The only difference is to replace $\widehat{\mathbf{\Sigma}}_{k,m}$ with $\dot{\widehat{\mathbf{\Sigma}}}_{k,m}$ (or $\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}$), $\widetilde{\mathbf{\Sigma}}_{k,h,m}$ with $\widehat{\mathbf{\Sigma}}_{k,h,m}$ (or still $\widetilde{\mathbf{\Sigma}}_{k,h,m}$), $\boldsymbol{\phi}_{k,h,m}$ with $\widehat{\boldsymbol{\phi}}_{k,h,m}$ (or $\widetilde{\boldsymbol{\phi}}_{k,h,m}$). □

2.8.8.6 Proof of Lemma 2.8.26

Proof of Lemma 2.8.26. The proof follows the proof of Lemma 25 in Zhang et al. (2021d) and Lemma C.6 in Zhou and Gu (2022a). We have,

$$\begin{aligned}
\widehat{S}_m =& \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left[\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^{m+1}}\right]\left(s_h^k, a_h^k\right) - \left(\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right)\right)^2\right] \\
=& \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left[\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^{m+1}}\right]\left(s_h^k, a_h^k\right) - \widehat{V}_{k,h+1}^{2^{m+1}}\left(s_{h+1}^k\right)\right] + \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\Bigg[\widehat{V}_{k,h}^{2^{m+1}}\left(s_h^k\right) \\
& - \left(\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right)\right)^2\Bigg] + \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left(\widehat{V}_{k,h+1}^{2^{m+1}}\left(s_{h+1}^k\right) - \widehat{V}_{k,h}^{2^{m+1}}\left(s_h^k\right)\right) \\
\leqslant& \widehat{A}_{m+1} + \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left[\widehat{V}_{k,h}^{2^{m+1}}\left(s_h^k\right) - \left(\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k, a_h^k\right)\right)^2\right] + \sum_{k=1}^{K} I_{h_k}^k\widehat{V}_{k,h_k+1}^{2^{m+1}}\left(s_{h_k+1}^k\right),
\end{aligned} \tag{2.8.75}$$

where h_k is the largest index satisfying $I_h^k = 1$. For the second term, we have

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[\widehat{V}_{k,h}^{2^{m+1}}\left(s_h^k\right) - \left(\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)\right)^2\right] \\
&\leqslant \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[\widehat{V}_{k,h}^{2^{m+1}}\left(s_h^k\right) - \left(\left[\mathbb{P}\widehat{V}_{k,h+1}\right]\left(s_h^k,a_h^k\right)\right)^{2^{m+1}}\right] \\
&= \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left(\widehat{V}_{k,h}\left(s_h^k\right) - \left[\mathbb{P}\widehat{V}_{k,h+1}\right]\left(s_h^k,a_h^k\right)\right)\prod_{i=0}^{m}\left(\widehat{V}_{k,h}^{2^i}\left(s_h^k\right) + \left[\mathbb{P}\widehat{V}_{k,h+1}\right]\left(s_h^k,a_h^k\right)^{2^i}\right) \\
&\leqslant 2^{m+1}\sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \max\left\{\widehat{V}_{k,h}\left(s_h^k\right) - \left[\mathbb{P}\widehat{V}_{k,h+1}\right]\left(s_h^k,a_h^k\right),0\right\} \\
&\leqslant 2^{m+1}\sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \left[u_{k,h}\left(s_h^k,a_h^k\right) + 4\min\left\{1,\beta\left\|\widehat{\boldsymbol{\phi}}_{k,h,0}\right\|_{\dot{\widehat{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\}\right] \\
&\leqslant 2^{m+1}\left(\widetilde{R}_0 + 4\widehat{R}_0\right),
\end{aligned} \tag{2.8.76}$$

where the first inequality holds due to using $\mathbb{E}X^2 \geqslant (\mathbb{E}X)^2$ recursively, the first equality holds due to the fact $x^{2^{m+1}} - y^{2^{m+1}} = (x-y)\prod_{i=0}^{m}(x^{2^m} - y^{2^m})$, the second inequality holds due to $\widehat{V}_{k,h}$ belongs to the interval $[0,1]$, the third inequality holds due to Lemma 2.8.24, and the last inequality holds due to $u_{k,h}(s_h^k,a_h^k) = \beta\|\boldsymbol{\phi}_{V_{h+1}(\cdot;\boldsymbol{\theta}_k,\pi_k,r_k)}(s_h^k,a_h^k)\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}} = \beta\|\widetilde{\boldsymbol{\phi}}_{k,h,0}\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}$. If $h_K \leqslant H$, we have $I_{h_k}^k \widehat{V}_{k,h_k+1}^{2^{m+1}}\left(s_{h_k+1}^k\right) \leqslant 1 = 1 - I_H^k$, and if $h_K = H$, $I_{h_k}^k \widehat{V}_{k,h_k+1}^{2^{m+1}}\left(s_{h_k+1}^k\right) = 0 = 1 - I_H^k$, which both give

$$\sum_{k=1}^{K} I_{h_k}^k \widehat{V}_{k,h_k+1}^{2^{m+1}}\left(s_{h_k+1}^k\right) \leqslant \sum_{k=1}^{K}(1 - I_H^k) = G \tag{2.8.77}$$

Substituting Equations (2.8.75), (2.8.76),(2.8.77) into (2.8.75), we can get (2.8.49). For Equation (2.8.44), similarly, we have

$$\widetilde{S}_m \leqslant \widetilde{A}_{m+1} + \sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left[\widetilde{V}_{k,h}^{2^{m+1}}\left(s_h^k\right) - \left(\left[\mathbb{P}\widetilde{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)\right)^2\right] + \sum_{k=1}^{K} I_{h_k}^k \widetilde{V}_{k,h_k+1}^{2^{m+1}}\left(s_{h_k+1}^k\right), \tag{2.8.78}$$

$$\sum_{k=1}^{K} I_{h_k}^k \widetilde{V}_{k,h_k+1}^{2^{m+1}}\left(s_{h_k+1}^k\right) \leqslant \sum_{k=1}^{K}\left(1 - I_H^k\right) = G. \tag{2.8.79}$$

And we have

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left[\widehat{V}_{k,h}^{2^{m+1}}\left(s_h^k\right)-\left(\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)\right)^2\right]\\
&\leqslant 2^{m+1}\sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k \max\left\{\widetilde{V}_{k,h}\left(s_h^k\right)-\left[\mathbb{P}\widetilde{V}_{k,h+1}\right]\left(s_h^k,a_h^k\right),0\right\}\\
&\leqslant 2^{m+1}\sum_{k=1}^{K}\sum_{h=1}^{H} I_h^k\left[r_{k,h}(s_h^k,s_h^k)+\min\left\{1,2\beta\left\|\widetilde{\boldsymbol{\phi}}_{k,h,0}\right\|_{\dot{\widetilde{\boldsymbol{\Sigma}}}_{k,0}^{-1}}\right\}\right]\\
&\leqslant 2^{m+1}\left(K+2\widetilde{R}_0\right) \qquad (2.8.80)
\end{aligned}$$

where the first inequality holds similar to the derivation of (2.8.76), second inequality follows Lemma 2.8.24, and the third inequality holds due to $\sum_{h=1}^{H} r_{k,h}(s_h^k,a_h^k)\leqslant 1$. Plugging Equations (2.8.79) (2.8.80) into 2.8.78, we obtain 2.8.50 □

2.8.8.7 Proof of Lemma 2.8.27

Proof of Lemma 2.8.27. The proof follows the proof of Lemma 25 in Zhang et al. (2021d) and Lemma C.7 in Zhou and Gu (2022a). We use Lemma 2.8.32 for $\widehat{A}_m$ and $\widetilde{A}_m$ for each m. To avoid confusion, we write ϵ,δ in Lemma 2.8.32 as ϵ',δ'.

Let $x_{k,h}=I_h^k\left[\left[\mathbb{P}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)-\widehat{V}_{k,h+1}^{2^m}(s_{h+1}^k)\right]$, $n=KH$, $\epsilon'=\sqrt{\log(1/\delta')}$, and $\delta'=\delta/(4\log(KH))$. Thus, $\mathbb{E}\left[\widehat{x}_{k,h}|\mathcal{F}_{k,h}\right]=0$ and $\mathbb{E}\left[\widehat{x}_{k,h}^2\middle|\mathcal{F}_{k,h}\right]=I_h^k\left[\mathbb{V}V_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)$. Therefore, for each $m\in\overline{[M]}$, with probability at least $1-\delta$, we have

$$|\widehat{A}_m|=\left|\sum_{k=1}^{K}\sum_{h=1}^{H}x_{k,h}\right|\leqslant\sqrt{2\zeta\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k\left[\mathbb{V}\widehat{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)}+\zeta.$$

Similarly, let $x_{k,h}=I_h^k\left[\left[\mathbb{P}\widetilde{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)-\widetilde{V}_{k,h+1}^{2^m}(s_{h+1}^k)\right]$, $n=KH$, $\epsilon'=\sqrt{\log(1/\delta')}$, and $\delta'=\delta/(4\log(KH))$. With probability at least $1-\delta$, we have

$$|\widetilde{A}_m|=\left|\sum_{k=1}^{K}\sum_{h=1}^{H}x_{k,h}\right|\leqslant\sqrt{2\zeta\sum_{k=1}^{K}\sum_{h=1}^{H}I_h^k\left[\mathbb{V}\widetilde{V}_{k,h+1}^{2^m}\right]\left(s_h^k,a_h^k\right)}+\zeta.$$

Taking union bound over $m\in\overline{[M]}$ completes the proof. □

2.8.8.8 Proof of Lemma 2.8.28

Proof of Lemma 2.8.28. By the fact that $\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k+1,m}^{-1/2}\right) < \det\left(\widehat{\mathbf{\Sigma}}_{k,H,m}^{-1/2}\right)$ and $\det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k+1,m}^{-1/2}\right) < \det\left(\widetilde{\mathbf{\Sigma}}_{k,H,m}^{-1/2}\right)$, we have

$$\begin{aligned}(1 - I_H^k) = 1 \Leftrightarrow \exists m \in \overline{[M]}, &\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}^{-1/2}\right) / \det\left(\widehat{\mathbf{\Sigma}}_{k,H,m}^{-1/2}\right) > 4 \\ &\text{or } \det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}^{-1/2}\right) / \det\left(\widetilde{\mathbf{\Sigma}}_{k,H,m}^{-1/2}\right) > 4 \\ \Rightarrow \exists m \in \overline{[M]}, &\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}^{-1/2}\right) / \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k+1,m}^{-1/2}\right) > 4 \\ &\text{or } \det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}^{-1/2}\right) / \det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k+1,m}^{-1/2}\right) > 4 \end{aligned} \tag{2.8.81}$$

Let $\widehat{\mathcal{D}}_m$ and $\widetilde{\mathcal{D}}_m$ denote the indices k such that

$$\begin{aligned}\widehat{\mathcal{D}}_m &:= \left\{k \in [K] : \ \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k+1,m}\right) / \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}\right) > 16\right\} \\ \widetilde{\mathcal{D}}_m &:= \left\{k \in [K] : \ \det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k+1,m}\right) / \det\left(\dot{\widetilde{\mathbf{\Sigma}}}_{k,m}\right) > 16\right\}\end{aligned}$$

Then we have

$$G \leqslant \left|\bigcup_{m=0}^{M-1} \widehat{\mathcal{D}}_m \cup \bigcup_{m=0}^{M-1} \widetilde{\mathcal{D}}_m\right| \leqslant \sum_{m=0}^{M-1} \left|\widehat{\mathcal{D}}_m\right| + \sum_{m=0}^{M-1} |\widetilde{\mathcal{D}}_m|$$

For each m, we have

$$\begin{aligned}2\left|\widehat{\mathcal{D}}_m\right| &< \sum_{k \in \widehat{\mathcal{D}}_m} \log 16 < \sum_{k \in \widehat{\mathcal{D}}_m} \log\left(\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k+1,m}\right) / \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}\right)\right) \\ &\leqslant \sum_{k=1}^{K} \log\left(\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k+1,m}\right) / \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}\right)\right)\end{aligned}$$

Furthermore, since $\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{K+1,m}\right) \leqslant \left(\operatorname{tr}\left(\dot{\widehat{\mathbf{\Sigma}}}_{K+1,m}\right)/d\right)^d$ and $\operatorname{tr}\left(\dot{\widehat{\mathbf{\Sigma}}}_{K+1,m}\right) \leqslant \operatorname{tr}(\lambda I) + \sum_{k,h} \left\|\widehat{\boldsymbol{\phi}}_{k,h.m}\right\|_2^2 / \widehat{\sigma}_{k,h,m}^2 \leqslant d\lambda + KH/\alpha^2$

$$\begin{aligned}\sum_{k=1}^{K} \log\left(\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k+1,m}\right) / \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{k,m}\right)\right) &= \log\left(\det\left(\dot{\widehat{\mathbf{\Sigma}}}_{K+1,m}\right) / \det\left(\dot{\widehat{\mathbf{\Sigma}}}_{1,m}\right)\right) \\ &\leqslant d\left(\log\left(\lambda + KH/(d\alpha^2)\right) - \log(\lambda)\right)\end{aligned}$$

Therefore $|\widehat{\mathcal{D}}_m|$ is upper bounded by

$$|\widehat{\mathcal{D}}_m| < d/2 \log(1 + KH/(d\lambda\alpha^2)).$$

And same for $|\widetilde{\mathcal{D}}_m|$. Taking summation gives the upper bound of G. □

2.8.9 Auxiliary Lemmas

Lemma 2.8.32 (Lemma 11, Zhang et al. 2021e)**.** Let $M > 0$ be a constant. Let $\{x_i\}_{i=1}^n$ be a stochastic process, $\mathcal{G}_i = \sigma(x_1, \ldots, x_i)$ be the σ-algebra of $x_1, \ldots, x_i$. Suppose $\mathbb{E}[x_i \mid \mathcal{G}_{i-1}] = 0, |x_i| \leqslant M$ and $\mathbb{E}[x_i^2 \mid \mathcal{G}_{i-1}] < \infty$ almost surely. Then, for any $\delta, \varepsilon > 0$, we have

$$\begin{aligned}
&\mathbb{P}\left(\left|\sum_{i=1}^n x_i\right| \leqslant 2\sqrt{2\log(1/\delta)\sum_{i=1}^n \mathbb{E}[x_i^2 \mid \mathcal{G}_{i-1}]} + 2\sqrt{\log(1/\delta)}\varepsilon + 2M\log(1/\delta)\right) \\
&\quad > 1 - 2\left(\log\left(M^2 n/\varepsilon^2\right) + 1\right)\delta. \qquad (2.8.82)
\end{aligned}$$

Lemma 2.8.33 (Lemma 12, Zhang et al. (2021d))**.** Let $\lambda_1, \lambda_2, \lambda_4 > 0, \lambda_3 \geqslant 1$ and $\kappa = \max\{\log_2 \lambda_1, 1\}$. Let $a_1, \ldots, a_\kappa$ be non-negative real numbers such that

$$a_i \leqslant \min\left\{\lambda_1, \lambda_2\sqrt{a_i + a_{i+1} + 2^{i+1}\lambda_3} + \lambda_4\right\}$$

for any $1 \leqslant i \leqslant \kappa$. Let $a_{\kappa+1} = \lambda_1$. Then we have $a_1 \leqslant 22\lambda_2^2 + 6\lambda_4 + 4\lambda_2\sqrt{2\lambda_3}$.

CHAPTER 3

Uncertainty-Aware Unsupervised Exploration in Deep Reinforcement Learning

3.1 Introduction

In Chapter 2, we discussed the theoretical framework of reward-free exploration, especially with linear function approximation. In this chapter, we aim to extend this analysis and algorithm in a more general and practical setting, which is aligned with the current practice of deep reinforcement learning. We also seek to build the foundation of unsupervised reinforcement learning through the lens of reward-free exploration.

Deep reinforcement learning (RL) has been the source of many breakthroughs in games (e.g., Atari game (Mnih et al., 2013) and Go game (Silver et al., 2016)) and robotic control (Levine et al., 2016) over the last ten years. A key component of RL is exploration, which requires the agent to explore different states and actions before finding a near-optimal policy. Traditional exploration strategy involves iteratively executing a policy guided by a specific reward function, limiting the trained agent to solving only the single task for which it was trained. Designing an efficient exploration strategy agnostic to reward functions is crucial, as it prevents the agent from repeated learning under different reward functions, thereby avoiding inefficiency and potential intractability in sample complexity.

Therefore, as discussed in Chapter 2, *reward-free exploration* (Jin et al., 2020a) is proposed to improve the efficiency of exploration without reward functions. A series of theoretical works have presented efficient exploration strategies with performance guarantees, as

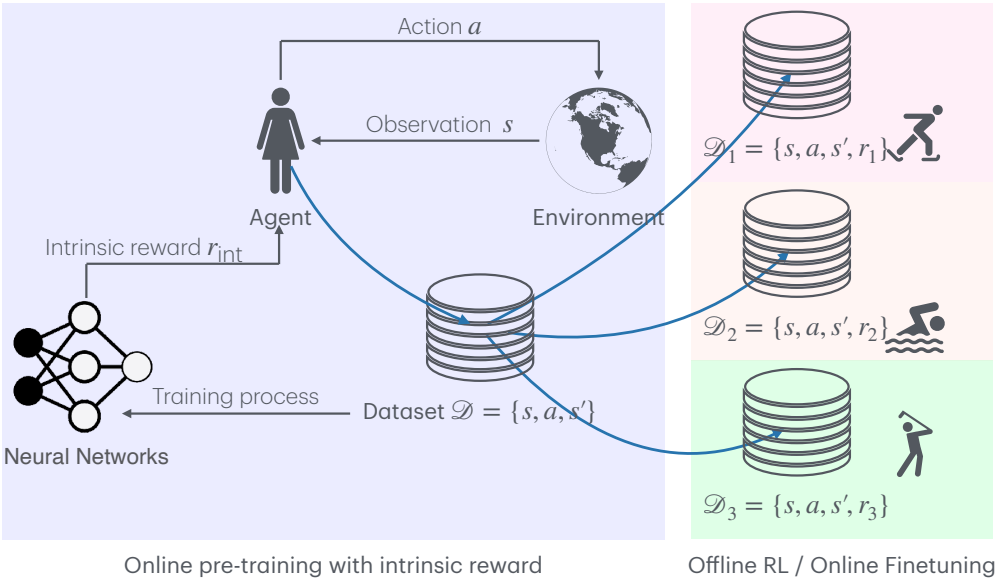


Figure 3.1: Diagram of the unsupervised reinforcement learning paradigm.

we discussed in Chapter 2. On the other hand, from the empirical perspective, *unsupervised reinforcement learning* (Laskin et al., 2021) has emerged as a new paradigm for encouraging the agent to explore without predefined supervision. Unsupervised RL diverges from classical RL approaches by not relying on a specific reward function for exploration. Instead, Unsupervised RL utilizes an "intrinsic reward", a.k.a., pseudo-reward function, defined based on all previously explored samples. This encourages the agent to venture into unexplored states and actions. In particular, in the realm of deep RL where no linear structural assumptions are made, recent studies (Pathak et al., 2017; Burda et al., 2018b; Eysenbach et al., 2018; Lee et al., 2019; Pathak et al., 2019; Liu and Abbeel, 2021a,b) have developed unsupervised RL algorithms by employing various intrinsic reward functions, demonstrating promising performance in finding the near-optimal policy. As presented in Figure 3.1, unsupervised reinforcement learning is similar with the reward-free exploration discussed in Chapter 2. Compared with the reward-free RL, empirical unsupervised reinforcement learning uses the intrinsic reward to motivate exploration and the additional application of online fine-tuning to learn the different rewards or objectives.

Despite the success of these heuristics on designing the intrinsic rewards for unsupervised RL, these empirical results lack rigorous justification and could be further optimized. From the theoretical analysis perspective, for example, Kong et al. (2021) defined an intrinsic reward based on the maximum difference between function pairs that show similarity in past data. This approach essentially treats each collected sample equally. It is a well-established principle in RL that in order to achieve optimal sample efficiency, different samples should be treated distinctively based on their importance. Notably, Zhang et al. (2023a) utilized variance-dependent weights to address the heteroscedasticity observed in samples, thereby achieving optimal sample complexity in linear mixture MDPs. However, this approach calculates its intrinsic reward by nested iterative optimization, which hampers computational efficiency and practical applicability. Therefore, for the unsupervised reinforcement learning tasks, we are faced with the following question:

Is it possible to craft an intrinsic reward function to explore the environment without supervision?

3.1.1 Organization of this Chapter

In this chapter, we will answer the above question affirmatively from both a theoretical perspective and an empirical perspective. This chapter is organized as follows. We first present the related works in Section 3.2 and preliminaries in Section 3.3. In in Section 3.4, we propose a variance-adaptive intrinsic reward for unsupervised reinforcement learning. In Section 3.5, we show that our method enjoys a finite sample complexity in finding the near-optimal policy for any given reward, and our theoretical guarantee is tighter than that of existing methods. In Section 3.6, we conduct experiments and show that by incorporating variance information, a series of existing baselines can be further improved in terms of sample efficiency. The conclusion is drawn in Section 3.7 and we defer detailed proof of the algorithm to Section 3.8.

3.2 Related Works

3.2.1 Unsupervised Reinforcement Learning

With recent advances in unsupervised CV and NLP tasks, unsupervised reinforcement learning has emerged as a new paradigm trying to learn the environment without supervision, such as the reward signals. As suggested in Laskin et al. (2021), these works are mainly separated into two lines: unsupervised representation learning in RL and unsupervised behavioral learning.

Unsupervised representation learning in RL mainly addresses issues on how to learn good representations for different states s, which can facilitate efficient learning of a policy $\pi(a|s)$. From the theoretical side, a list of works have identified how to select or learn good representations for various RL tasks with linear function approximations, by using MLE (Uehara et al., 2021), contrastive learning (Qiu et al., 2022) or model selection (Papini et al., 2021a; Zhang et al., 2021a). From the empirical side, various methods in unsupervised learning or self-supervised learning are applied to RL tasks, including contrastive learning (Laskin et al., 2020; Stooke et al., 2021; Yarats et al., 2021a), autoencoders (Yarats et al., 2021b) and world models (Hafner et al., 2019a,b).

Unsupervised behavioral learning in RL aims to eliminate this reward signal during exploration. Therefore, the agent can be adapted to different tasks in the downstream fine-tuning. To replace the 'extrinsic' reward signals, these methods usually leverage different 'intrinsic rewards' during exploration. Many recent algorithms have been proposed to learn from different types of intrinsic reward, which is based on the prediction error (Pathak et al., 2017; Burda et al., 2018a; Pathak et al., 2019), information gain (Eysenbach et al., 2018; Hansen et al., 2019; Sharma et al., 2019) and entropy (Liu and Abbeel, 2021a,b; Seo et al., 2021) of the observations. URLB (Laskin et al., 2021) provided a unified framework providing benchmarks for all these intrinsic rewards.

3.2.2 Reinforcement Learning with General Function Approximation

RL with general function approximation has been widely studied in recent years, due to its ability to describe a wide range of existing RL algorithms. To explore the theoretical limits of RL and understand the practical DRL algorithms, various statistical complexity measurements for general function approximation have been proposed and developed. For instance, Bellman rank (Jiang et al., 2017), Witness rank (Sun et al., 2019), eluder dimension (Russo and Van Roy, 2013), Bellman eluder dimension (Jin et al., 2021), Decision-Estimation Coefficient (DEC) (Foster et al., 2021), Admissible Bellman Characterization (Chen et al., 2022c), generalized eluder dimension (Agarwal et al., 2022), etc. Among different statistical complexity measurements, Foster et al. (2021) showed a DEC-based lower bound of regret which holds for any function class. Specifically, our algorithm falls into the category of generalized eluder dimension function class, which includes linear MDPs (Jin et al., 2020b) as its special realization.

3.3 Preliminaries

3.3.1 Time-Inhomogeneous Episodic MDPs

We model the sequential decision making problem via time-inhomogeneous episodic Markov decision processes (MDPs), which can be denoted as tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H, r = \{r_h\}_{h=1}^H)$ by convention. Here, $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces, H is the length of each episode, $\mathbb{P}_h : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition probability function at stage h for state s to transit to state s' after executing action a, and $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the deterministic reward function at stage h. For any policy $\pi = \{\pi_h\}_{h=1}^H$, reward $r = \{r_h\}_{h=1}^H$, and stage $h \in [H]$, the value function $V_h^\pi(s; r)$ and the state-action value function $Q_h^\pi(s, a; r)$ is defined

as:

$$Q_h^\pi(s,a;r) = \mathbb{E}\Big[\sum_{h'=h}^{H} r_{h'}\big(s_{h'},a_{h'}\big)\Big| s_h = s, a_h = a, s_{h'+1}\sim \mathbb{P}_{h'}(\cdot|s_{h'},a_{h'}), a_{h'+1}=\pi(s_{h'+1})\Big],$$

$$V_h^\pi(s;r) = Q_h^\pi(s,\pi_h(s);r).$$

Furthermore, the optimal value function $V_h^*(s;r)$ is defines as $\max_\pi V_h^\pi(s;r)$, and the optimal action-value function $Q_h^*(s,a;r)$ is defined as $\max_\pi Q_h^\pi(s,a;r)$. For simplicity, we utilize the following bounded total reward assumption:

Assumption 3.3.1. The total reward for every possible trajectory is assumed to be within the interval of $(0,1)$.

Up to rescaling, Assumption 3.3.1 is more general than the standard reward scale assumption where $r_h \in [0,1]$ for all $h \in [H]$. Assumption 3.3.1 also ensures that the value function $V_h^\pi(s)$ and action-value function $Q_h^\pi(s,a;r)$ belong to the interval $[0,1]$.

For any function $V : \mathcal{S} \to \mathbb{R}$ and stage $h \in [H]$, the first-order Bellman operator $\mathcal{T}_h$ is defined as:

$$\mathcal{T}_h V(s,a;r) = \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}\Big[r_h(s,a) + V(s';r)\Big].$$

For simplicity, we further define the shorthand:

$$[\mathbb{P}_h V](s,a;r) = \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot|s,a)}V(s';r), [\mathbb{V}_h V](s,a;r) = [\mathbb{P}_h V^2](s,a;r) - [\mathbb{P}_h V]^2(s,a;r).$$

Throughout the paper, if the reward r is clear in the context, we omit the notation r in Q and V for simplicity.

3.3.2 General Function Approximation

In this work, we focus on the model-free value-based RL methods, which require us to use a predefined function class to estimate the optimal value function $Q_h^*(s,a;r)$ for any reward r. We use $\mathcal{F} := \{\mathcal{F}_h\}_{h=1}^H$ to denote the function class we will use during all H stages. To build

the statistical complexity of using $\mathcal{F}$ to learn $Q_h^*(s,a;r)$, we require several assumptions and definitions that characterize the cardinality of the function class.

Assumption 3.3.2 (Completeness, Zhao et al. (2023))**.** Given $\mathcal{F} := \{\mathcal{F}_h\}_{h=1}^H$ which is composed of bounded functions $f_h : \mathcal{S} \times \mathcal{A} \to [0, L]$. We assume that for any h and function $V : \mathcal{S} \to [0,1]$ and $r : \mathcal{S} \times \mathcal{A} \to [0,1]$, there exist $f_1, f_2 \in \mathcal{F}_h$ such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$f_1(s,a) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)}\big[r(s,a) + V(s')\big], f_2(s,a) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)}\Big[\big(r(s,a) + V(s')\big)^2\Big].$$

We assume that $L = O(1)$ throughout the paper.

Definition 3.3.3 (Generalized eluder dimension, Agarwal et al. 2022)**.** Let $\lambda \geqslant 0$ and $h \in [H]$, a sequence of state-action pairs $Z_h = \{z_{i,h} = (s_h^i, a_h^i)\}_{i \in [K]}$ and a sequence of positive numbers $\boldsymbol{\sigma}_h = \{\sigma_{i,h}\}_{i \in [K]}$. The generalized eluder dimension of a function class $\mathcal{F}_h : \mathcal{S} \times \mathcal{A} \to [0, L]$ with respect to λ is defined by $\dim_{\alpha,K}(\mathcal{F}_h) := \sup_{Z_h, \boldsymbol{\sigma}_h : |Z_h| = K, \boldsymbol{\sigma}_h \geqslant \alpha} \dim(\mathcal{F}_h, Z_h, \boldsymbol{\sigma}_h)$

$$\dim(\mathcal{F}_h, Z_h, \boldsymbol{\sigma}_h) := \sum_{i=1}^K \min\left(1, \frac{1}{\sigma_i^2} D^2_{\mathcal{F}_h}(z_{i,h}; z_{[i-1],h}, \sigma_{[i-1],h})\right),$$

$$D^2_{\mathcal{F}_h}(z; z_{[i-1],h}, \sigma_{[i-1],h}) := \sup_{f_1, f_2 \in \mathcal{F}_h} \frac{(f_1(z) - f_2(z))^2}{\sum_{s \in [i-1]} \frac{1}{\sigma_{s,h}^2}(f_1(z_{s,h}) - f_2(z_{s,h}))^2 + \lambda}.$$

We write $\dim_{\alpha,K}(\mathcal{F}) := H^{-1} \cdot \sum_{h \in [H]} \dim_{\alpha,K}(\mathcal{F}_h)$ for short when $\mathcal{F}$ is a collection of function classes $\mathcal{F} = \{\mathcal{F}_h\}_{h=1}^H$ in the context.

Remark 3.3.4. Kong et al. (2021) introduced a similar definition called "sensitivity". In particular, it is defined by

$$\text{sensitivity}_{\mathcal{Z},\mathcal{F}}(z) := \sup_{f_1, f_2 \in \mathcal{F}} \frac{(f_1(z) - f_2(z))^2}{\min\{\sum_{(s,a) \in \mathcal{Z}} (f_1(s,a) - f_2(s,a))^2, \lambda\}},$$

where λ is defined by $T(H+1)^2$ for the RL task with $r_h(s,a) \in [0,1]$ [1]. The major difference between the generalized eluder dimension and sensitivity is that the generalized eluder dimension incorporates the variance σ_s^2 into the historical observation $\mathcal{Z}$ to craft the heterogeneous variance in $\mathcal{Z}$.

[1] We ignore the clipping process making $\text{sensitivity}_{\mathcal{Z},\mathcal{F}}(z) \leftarrow \min\{\text{sensitivity}_{\mathcal{Z},\mathcal{F}}(z)\}$ for the clarity of demonstration

Since $D^2_{\mathcal{F}_h}$ in Definition 3.3.3 is not computationally efficient in some circumstances, we approximate it via an oracle $\overline{D}^2_{\mathcal{F}_h}$, which is formally defined in Definition 3.3.5.

Definition 3.3.5 (Bonus oracle $\overline{D}^2_{\mathcal{F}_h}$)**.** The bonus oracle returns a computable function $\overline{D}^2_{\mathcal{F}_h}(z; z_{[t],h}, \sigma_{[t],h})$, which computes the estimated uncertainty of a state-action pair $z = (s, a) \in \mathcal{S} \times \mathcal{A}$ with respect to historical data $z_{[t],h}$ and corresponding weights $\sigma_{[t],h}$. It satisfies

$$D_{\mathcal{F}_h}(z; z_{[t],h}, \sigma_{[t],h}) \leqslant \overline{D}_{\mathcal{F}_h}(z; z_{[t],h}, \sigma_{[t],h}) \leqslant C \cdot D_{\mathcal{F}_h}(z; z_{[t],h}, \sigma_{[t],h}),$$

where C is a fixed constant.

The covering numbers of the value function class and the bonus function class are introduced in the following definition.

Definition 3.3.6 (Covering numbers of function classes)**.** For any $\epsilon > 0$, we define the following covering numbers of involved function classes:

1. For each $h \in [H]$, there exists an ϵ-cover $\mathcal{C}(\mathcal{F}_h, \epsilon) \subseteq \mathcal{F}_h$ with size $|\mathcal{C}(\mathcal{F}_h, \epsilon)| \leqslant N_{\mathcal{F}_h}(\epsilon)$, such that for any $f \in \mathcal{F}_h$, there exists $f' \in \mathcal{C}(\mathcal{F}_h, \epsilon)$ satisfying $\|f - f'\|_\infty \leqslant \epsilon$. For any $\epsilon > 0$, we define the uniform covering number of $\mathcal{F}$ with respect to ϵ as $N_{\mathcal{F}}(\epsilon) := \max_{h\in[H]} N_{\mathcal{F}_h}(\epsilon)$.

2. There exists a bonus function class $\mathcal{B} = \{B : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$ such that for any $t \geqslant 0$, $z_{[t]} \in (\mathcal{S} \times \mathcal{A})^t$, $\sigma_{[t]} \in \mathbb{R}^t$, $h \in [H]$, the bonus function $\overline{D}_{\mathcal{F}}(\cdot; z_{[t]}, \sigma_{[t]})$ returned by the bonus oracle in Definition 3.3.5 belongs to $\mathcal{B}$.

3. For the bonus function class $\mathcal{B}$, there exists an ϵ-cover $\mathcal{C}(\mathcal{B}, \epsilon) \subseteq \mathcal{B}$ with size $|\mathcal{C}(\mathcal{B}, \epsilon)| \leqslant N_{\mathcal{B}}(\epsilon)$, such that for any $b \in \mathcal{B}$, there exists $b' \in \mathcal{C}(\mathcal{B}, \epsilon)$, such that $\|b - b'\|_\infty \leqslant \epsilon$.

4. The optimistic function class at stage $h \in [H]$ is:

$$\mathcal{V}_h = \left\{V(\cdot) = \max_{a\in\mathcal{A}} \min\Big(1, f(\cdot, a) + \beta \cdot b(\cdot, a)\Big) \,\middle|\, f \in \mathcal{F}_h, b \in \mathcal{B}\right\}.$$

There exists an ϵ-cover $\mathcal{C}(\mathcal{V}_h, \epsilon)$ with size $|\mathcal{C}(\mathcal{V}_h, \epsilon)| \leqslant N_{\mathcal{V}_h}(\epsilon)$. For any $\epsilon > 0$, we define the uniform covering number of $\mathcal{V}$ with respect to ϵ as $N_{\mathcal{V}}(\epsilon) := \max_{h\in[H]} N_{\mathcal{V}_h}(\epsilon)$.

3.4 Proposed Algorithm

In this section, we introduce our algorithm GFA-RFE as presented in Algorithm 6 and Algorithm 7. GFA-RFE consists of two phases, where in the first exploration phase as Algorithm 6, GFA-RFE collects K episodes without reward signal. Then in the second planning phase as presented in Algorithm 7, GFA-RFE leverages the collected K episodes to learn a policy trying to maximize the cumulative reward given a specific reward function r. The details of these two phases are presented in the following subsections.

3.4.1 Exploration Phase: Efficient Exploration via Uncertainty-aware Intrinsic Reward

The ultimate goals of the exploration phase are exploring environments and collecting data in the absence of reward to facilitate finding the near-optimal policy in the next phase. At a high level, GFA-RFE achieves these goals by encouraging the agent to explore regions containing higher uncertainty, which intuitively guarantees the maximal information gained in each episode.

3.4.1.1 Intrinsic Reward

GFA-RFE evaluate the uncertainty by $D_{\mathcal{F}_h}$ in Definition 3.3.3, and uses its oracle $\overline{D}_{\mathcal{F}_h}$ as the intrinsic reward $r_{k,h}$ in Line 4 to generate an uncertainty-target policy in Line 8. Recall that $D^2_{\mathcal{F}_h}(z; z_{[k-1],h}, \sigma_{[k-1],h})$ is defined as

$$\sup_{f_1,f_2\in\mathcal{F}_h} \frac{(f_1(z) - f_2(z))^2}{\sum_{s\in[i-1]} \frac{1}{\sigma^2_{s,h}}(f_1(z_{s,h}) - f_2(z_{s,h}))^2 + \lambda}.$$

In particular, a high reward signal means that there exist functions in $\mathcal{F}_h$ close to each other on all historical observations but divergent for the current state and action pair. This further suggests that the past observations are not enough for the agent to make a precise value estimation for the current state-action pair.

3.4.1.2 Weighted Regression

The usage of the intrinsic reward $r_{k,h}$ induces an intrinsic action-value function $Q^*_{k,h}(\cdot,\cdot;r_k)$, which serves as a metric for cumulative uncertainty of remaining stages. As in model-free approaches, GFA-RFE aims to estimate $Q^*_{k,h}(\cdot,\cdot;r_k)$ and further finds a policy π^k_h that would maximize the cumulative uncertainty over H stages. This part is presented in Algorithm 6 through Line 5 to Line 8.

To reduce the estimation error, GFA-RFE incorporates the weighted regression proposed in Zhao et al. (2023) into estimating $Q^*_{k,h}(s,a;r_k)$. The algorithm starts at final stage $h=H$ and estimating the $Q^*_{k,h}(s,a;r_k)$ approximated by function $\widehat{f}_{k,h}$ using Bellman equation:

$$\widehat{f}_{k,h}(s_h,a_h) = r_{k,h}(s_h,a_h) + [\mathbb{P}_h V_{k,h+1}](s_h,a_h) \approx r_{k,h}(s_h,a_h) + V_{k,h+1}(s_{h+1}).$$

However, estimating $[\mathbb{P}_h V_{k,h+1}](s_h,a_h)$ using $V_{k,h+1}(s_{h+1})$ may also introduce error since the variance of distribution $\mathbb{P}_h(\cdot|s,a)$ varies among different state-action pair. Therefore, we tackle this heterogeneous variance issue by minimizing the Bellman residual loss weighted by using the estimated variance $\bar{\sigma}_{k,h}$ of observed state-action pairs s^i_h, a^i_h:

$$\sum_{i\in[k-1]} \frac{(f_{i,h}(s^i_h,a^i_h) - r_{i,h}(s^i_h,a^i_h) - V_{i,h+1}(s^i_{h+1}))^2}{\bar{\sigma}^2_{i,h}}.$$

Obviously, a lower variance $\bar{\sigma}_{i,h}$ yields a larger weight during the regression. The calculation of variances $\bar{\sigma}_{i,h}$ involves both *aleatoric uncertainty* and *epistemic uncertainty* (Kendall and Gal, 2017; Mai et al., 2022), where the *aleatoric uncertainty* is $\sigma_{k,h}$ calculated in Line 13 caused by indeterminism of the transition and *epistemic uncertainty* is $\overline{D}^{1/2}_{\mathcal{F}_h}$ caused by limited data. Such an approach can be proved to improve the sample efficiency of our algorithm

GFA-RFE (see Theorem 3.5.1 and its discussion). Similar approaches have been used in Zhou et al. (2021b); Ye et al. (2023) to provide more robust and efficient estimation.

After obtaining the $\widehat{f}_{k,h}$ function through weighted regression, GFA-RFE follows the standard optimism design in online exploration methods to add the bonus term $b_{k,h}$ for overestimating the $Q^*_{k,h}(s,a;r)$ function in Line 6. Using this optimistic estimation, GFA-RFE thus takes the greedy policy and estimates the value function $V_{k,h}$ in Line 7 before proceeding to the previous stage $h-1$.

Algorithm 6 GFA-RFE – Phase I: Exploration Phase

Input: Confidence radius β^E, regularization parameter λ

1: **for** $k = 1, 2, \cdots, K$ **do**

2: **for** $h = H, H-1, \cdots, 1$ **do**

3: $b_{k,h}(\cdot,\cdot) \leftarrow 2\beta^E \cdot \overline{\mathcal{D}}_{\mathcal{F}_h}(\cdot,\cdot; z_{[k-1],h}, \bar{\sigma}_{[k-1],h})$.

4: $r_{k,h}(\cdot,\cdot) \leftarrow b_{k,h}(\cdot\cdot)/2$.

5: $\widehat{f}_{k,h} \leftarrow \operatorname{argmin}_{f_h \in \mathcal{F}_h} \sum_{i\in[k-1]} \frac{1}{\bar{\sigma}^2_{i,h}} (f_h(s^i_h, a^i_h) - r_{k,h}(s^i_h, a^i_h) - V_{k,h+1}(s^i_{h+1}))^2$.

6: $Q_{k,h}(s,a) \leftarrow \min\Big\{\widehat{f}_{k,h}(s,a) + b_{k,h}(s,a), 1\Big\}$.

7: $V_{k,h}(s) \leftarrow \max_a Q_{k,h}(s,a)$.

8: Set the policy $\pi^k_h(\cdot) \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} Q_{k,h}(\cdot,a)$.

9: **end for**

10: Receive the initial state s^k_1.

11: **for** stage $h = 1, \ldots, H$ **do**

12: Take action $a^k_h \leftarrow \pi^k_h(s^k_h)$, receive next state s^k_{h+1}.

13: $\sigma_{k,h} \leftarrow 2\sqrt{\log N_{\mathcal{V}}(\epsilon) \cdot \min\{\widehat{f}_{k,h}(s^k_h, a^k_h), 1\}}$.

14: $\bar{\sigma}_{k,h} \leftarrow \max\big\{\gamma \cdot \overline{D}^{1/2}_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}),\, \sigma_{k,h}, \alpha\big\}$.

15: **end for**

16: **end for**

Algorithm 7 GFA-RFE – Phase II: Planning Phase

Input: Dataset $\{(s_h^k, a_h^k, \bar{\sigma}_{k,h}^2)\}_{(k,h)\in[K]\times[H]}$, confidence radius β^P

Input: Reward function $r = \{r_h\}_{h\in[H]}$

1: Initiate $\widehat{V}_{H+1}(\cdot) \leftarrow 0$, $\widehat{Q}_{H+1}(\cdot,\cdot) \leftarrow 0$

2: **for** step $h = H, \cdots, 1$ **do**

3: $b_h(\cdot,\cdot) \leftarrow \min\{\beta^P \overline{\mathcal{D}}_{\mathcal{F}_h}(z; z_{[K],h}, \bar{\sigma}_{[K],h}), 1\}$.

4: $\widehat{f}_h \leftarrow \operatorname{argmin}_{f_h\in\mathcal{F}_h} \sum_{i\in[K]} \frac{1}{\bar{\sigma}_{i,h}^2}(f_h(s_h^i, a_h^i) - r_h(s_h^i, a_h^i) - \widehat{V}_{h+1}(s_{h+1}^i))^2$.

5: $\widehat{Q}_h(s,a) \leftarrow \min\left\{\widehat{f}_h(s,a) + b_h(s,a), 1\right\}$.

6: $\widehat{V}_h(\cdot) \leftarrow \max_{a\in\mathcal{A}} \widehat{Q}_h(\cdot, a)$.

7: $\pi_h(\cdot) \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} \widehat{Q}_h(\cdot, a)$.

8: **end for**

Output: Policy π

3.4.2 Planning Phase: Effective Planning Using Weighted Regression

After exploring environments and collecting data in the exploration phase, the agent is now given the reward for a specific task, but no longer interacts with the environment. GFA-RFE enters its planning phase and ensures a policy to maximize the cumulative reward of r_h across all H stages. GFA-RFE estimates $Q_h^*(s,a;r)$ by weighted regression and further finds the optimal policy π_h, which is the same process as in the exploration phase. This part is presented in Algorithm 7 through Line 3 to Line 7.

Remark 3.4.1. Compared with Kong et al. (2021), our algorithm leverages the advantage of generalized elude dimension and incorporates the estimated variance σ into 1) weighted regression in Line 4 in the planning phase and Line 5 in exploration phase; 2) intrinsic reward design in Line 4. Also, our algorithm does not set the reward $r_{k,h} = b_{k,h}/H$ as of Kong et al. (2021); Wang et al. (2020b), thus the agent can explore more aggressively and more efficiently using the knowledge of variance of the observation. Therefore, GFA-RFE is more sample efficient compared with Kong et al. (2021), which is discussed in detail in Remark 3.5.7.

3.5 Sample Complexity Analysis

We analyze GFA-RFE theoretically in this section. The uncertainty-aware reward-free exploration mechanism leads to efficient learning with provable sample complexity guarantees. The first theorem characterizes how the sub-optimality decays as exploration time grows.

Theorem 3.5.1. For GFA-RFE, set confidence radius $\beta^E = \widetilde{O}\big(\sqrt{H \log N_{\mathcal{V}}(\epsilon)}\big)$ and $\beta^P = \widetilde{O}\big(\sqrt{H \log N_{\mathcal{F}}(\epsilon)}\big)$, and take $\alpha = 1/\sqrt{H}$ and $\gamma = \sqrt{\log N_{\mathcal{V}}(\epsilon)}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$, after collecting K episodes of samples, for any reward function $r = \{r_h\}_{h=1}^H$ such that $\sum_{h=1}^H r_h(s_h, a_h) \leqslant 1$, GFA-RFE outputs a policy π satisfying the following sub-optimality bound,

$$\mathbb{E}_{s_1 \sim \mu}[V_1^*(s_1; r) - V_1^\pi(s_1; r)] = \widetilde{O}\Big(H\sqrt{\log N_{\mathcal{F}}(\epsilon)}\sqrt{\dim_{\alpha,K}(\mathcal{F})/K}\Big).$$

We are now ready to present the sample complexity of GFA-RFE for the reward-free exploration.

Corollary 3.5.2. Under the same conditions in Theorem 3.5.1, with probability at least $1-\delta$, for any reward function $r = \{r_h\}_{h=1}^H$ such that $\sum_{h=1}^H r_h(s_h, a_h) \leqslant 1$, GFA-RFE returns an ϵ-optimal policy after collecting $K \leqslant \widetilde{\mathcal{O}}(H^2 \log N_{\mathcal{F}}(\epsilon) \dim_{\alpha,K}(\mathcal{F}) \epsilon^{-2})$ episodes during the exploration phase.

Remark 3.5.3. Let $d_{K,\delta}$ be $\max\{\log N_{\mathcal{F}}(\epsilon), \dim_{\alpha,K}(\mathcal{F})\}$, GFA-RFE yields an $\widetilde{\mathcal{O}}(H^2 d_{K,\delta}^2 \epsilon^{-2})$ sample complexity for reward-free exploration with high probability. In tabular setting, $d_{K,\delta} = \widetilde{\mathcal{O}}(SA)$, thus yields an $\widetilde{\mathcal{O}}(H^2 S^2 A^2 \epsilon^{-2})$ sample complexity. In linear MDPs and generalized linear MDPs with dimension d, $d_{K,\delta} = \widetilde{\mathcal{O}}(d)$, thus yields an $\widetilde{\mathcal{O}}(H^2 d^2 \epsilon^{-2})$ sample complexity which matches the result from Hu et al. (2022). For a more general setting where the function class with eluder dimension d, $d_{K,\delta} = \widetilde{\mathcal{O}}(d)$, which yields a $\widetilde{\mathcal{O}}(H^2 d^2 \epsilon^{-2})$ sample complexity.

For a fair comparison with some existing works, we translate our sample complexity

result to the case where the reward scale is $r_h \in [0,1]$, $\forall h \in [H]$. The result can be trivially obtained by replacing $r \to r/H$ in GFA-RFE.

Corollary 3.5.4. With probability at least $1-\delta$, for any reward function such that $r_h(s,a) \in [0,1]$ or the total reward is bounded by $\sum_{h=1}^{H} r_h(s_h,a_h) \leqslant H$, GFA-RFE returns an ϵ-optimal policy after collecting $K \leqslant \widetilde{\mathcal{O}}(H^4 d_{K,\delta}^2 \epsilon^{-2})$ episodes in the exploration phase.

Remark 3.5.5. Compared with Chen et al. (2022a) which provides a $\widetilde{\mathcal{O}}(d \log|\mathcal{P}|\epsilon^{-2})$ sample complexity for model-based RL, GFA-RFE is a *model-free* algorithm which does not need to directly sample transition kernel $\mathbb{P}_h(\cdot|\cdot,\cdot)$ from all possible transitions $\widetilde{\Delta}(\Pi)$, therefore, GFA-RFE is computationally efficient and can be easily implemented based on the current empirical DRL algorithms.

Remark 3.5.6. Compared with Chen et al. (2022b) which achieves a $\widetilde{\mathcal{O}}(H^7 d^3 \epsilon^{-2})$ sample complexity, one can find our result significantly improves the dependency on H, d. Chen et al. (2022b) didn't optimize the exploration policy by constructing intrinsic rewards but by updating Bellman error constraints on the value function class. It sacrificed the sample complexity to adapt the general function approximation settings. In addition, this approach is generally computationally intractable as it explicitly maintains feasible function classes. For its V-type variant, it even maintains a finite cover of the function class, which can be exponentially large.

Remark 3.5.7. Kong et al. (2021) leveraged the "sensitivity" as the intrinsic reward during the exploration and achieved a $\widetilde{\mathcal{O}}(H^6 d^4 \epsilon^{-2})$ reward-free sample complexity. Compare their algorithm and ours, ours improves a $H^2 d^2$ factor from 1) using weighted regression to handle heterogeneous observations 2) using a "truncated Bellman equation" (Chen et al., 2021) in our analysis, and 3) a properly improved uncertainty metric $\overline{D}_{\mathcal{F}_h}^2$ instead of the sensitivity.

Table 3.1: Cumulative reward for various exploration algorithms across different environments and tasks. The cumulative reward is averaged over 8 individual runs for both online exploration and offline planning. The result for each individual run is obtained by evaluating the policy network using the last-iteration parameter. Standard deviation is calculated across these runs. Results presented in **boldface** denote the best performance for each task, and those underlined represent the second-best outcomes. The cyan background highlights results of our algorithms.

Environment	Task	Baselines							Ours
		ICM	APT	DIAYN	APS	Dis.	SMM	RND	GFA-RFE
Walker	Flip	177 ± 80	523 ± 57	207 ± 119	246 ± 103	**570 ± 32**	242 ± 71	507 ± 48	554 ± 64
	Run	108 ± 41	304 ± 38	113 ± 38	132 ± 39	**340 ± 37**	116 ± 21	306 ± 34	339 ± 34
	Stand	466 ± 17	891 ± 62	587 ± 169	573 ± 177	726 ± 79	443 ± 104	750 ± 62	**925 ± 50**
	Walk	411±237	772±60	432 ± 222	645 ± 156	**851 ± 63**	273 ± 162	709 ± 115	826 ± 89
Quadruped	Run	93 ± 68	452 ± 49	158 ± 64	159 ± 82	**524 ± 24**	162 ± 140	522 ± 30	460 ± 36
	Jump	89 ± 47	740 ± 91	218 ± 114	123 ±67	**829 ± 22**	211 ± 127	790 ± 38	719 ± 68
	Stand	207 ± 134	910 ± 45	331 ± 81	308 ± 147	**953 ± 16**	239 ± 104	940 ± 27	867 ± 61
	Walk	94 ± 60	680 ± 117	171 ± 72	141 ± 80	720 ± 175	125 ± 36	**820 ± 94**	726 ± 146

3.6 Numerical Results

3.6.1 Experiment Setup

Based on our theoretical perspective, we integrate our algorithm in the unsupervised reinforcement learning (URL) framework and evaluate the performance of the proposed algorithm in URL benchmark (Laskin et al., 2021). As suggested by Ye et al. (2023), we use the variance of n-ensembled Q functions as the estimation of the bonus oracle $\overline{D}^2_{\mathcal{F}}$ which will be used in (1) intrinsic reward $r_{k,h}$; (2) exploration bonus $b_{k,h}$; and (3) weights $\sigma^2_{k,h}$ for the value target regression. All these Q networks are trained by Q-learning with different mini-batches in the replay buffer. Obviously, the variance of these Q networks comes from the randomness of initialization and the randomness of different mini-batches used in training. The pseudo

code for the practical algorithm is deferred to Section 3.9.

The original implementation of Laskin et al. (2021) involves two phases where the neural network is first *pretrained* by interacting with the environment without receiving reward signals and then *finetuned* by interacting with the environment again with reward signal. However, in our experiments, we strictly follow the design of reward-free exploration by first exploring the environment without the reward. The explored trajectories are collected into a dataset $\mathcal{D} = \{(s, a, s')\}$. Then we call a reward oracle r to assign rewards to this dataset $\mathcal{D}$ and learn the optimal policy using the offline dataset $\mathcal{D}_r = \{(s, a, s', r(s, a, s'))\}$ without interacting the dataset anymore. Intuitively speaking, this *online exploration + offline planning* paradigm is more challenging than the *online pretraining + online finetuning* and would be more practical, especially with different reward signals.

3.6.1.1 Unsupervised Reinforcement Learning Benchmarks

We conduct our experiments on *Unsupervised Reinforcement Learning Benchmarks* (Laskin et al., 2021), which consists of two multi-tasks environments (*Walker*, *Quadruped*) from DeepMind Control Suite (Tunyasuvunakool et al., 2020). Each environment is equipped with several reward functions and goals. For example, *Walker-run* consists of rewards encouraging the walker to run at speed and *Walker-stand* consists of rewards indicating the walker should stead steadily. We consider the state-based input in our experiments where the agent can directly observe the current state instead of image inputs (a.k.a. pixel-based).

3.6.1.2 Baseline Algorithms

We inherit the baseline algorithms ICM (Pathak et al., 2017), Disagreement (Pathak et al., 2019), RND (Burda et al., 2018b), APT (Liu and Abbeel, 2021b), DIAYN (Eysenbach et al., 2018), APS (Liu and Abbeel, 2021a), SMM (Lee et al., 2019). All these algorithms provide different 'intrinsic rewards' in place of ours during exploration. We make all these baseline

algorithms align with our settings which first collect an exploration dataset and then do offline training on the collected dataset with rewards.

3.6.2 Experiment Results

Experimental results are presented in Table 3.1. It's obvious that GFA-RFE can efficiently explore the environment without the reward function and then output a near-optimal policy given various reward functions. For the baseline algorithms, APT, Disagreement, and RND perform consistently better than the rest of the 4 algorithms on all 2 environments and 8 tasks. The performance of GFA-RFE enjoys compatible or superior performance compared with these top-level methods (APT, Disagreement, and RND), on these tasks. These promising numerical results justify our theoretical results and show that GFA-RFE can indeed efficiently learn the environment in a practical setting.

3.6.2.1 Ablation Study

To verify the performance of our algorithm, we also did ablation studies on 1) the relationship between offline training processes and episodic reward 2) the relationship between the quantity of online exploration data used in offline training and the achieve episodic reward. The details of the ablation study are deferred to Section 3.9.

3.7 Conclusion

In this chapter, we study the reward-free exploration under general function approximation, which can be viewed as a theoretical framework of the unsupervised reinforcement learning. We show that, with an uncertainty-aware intrinsic reward and variance-weighted regression on learning the environment, GFA-RFE can be theoretically proved to explore the environment efficiently without the existence of reward signals. Experiments show that our design of

intrinsic reward can be efficiently implemented and effectively used in an unsupervised reinforcement learning paradigm. In addition, experiment results verify that adding uncertainty estimation to the learning processes can improve the sample efficiency of the algorithm, which is aligned with our theoretical results of weighted regression.

3.8 Proofs

3.8.1 Proof of Theorems in Section 3.5

3.8.1.1 Additional Definitions and High Probability Events

In this section, we introduce additional definitions that will be used in the proofs. Also, we define the good events that GFA-RFE is guaranteed to have near-optimal sample complexity.

Definition 3.8.1 (Truncated Optimal Value Function)**.** We define the following truncated value functions for any reward r:

$$\begin{aligned}
\widetilde{V}^*_{H+1}(s;r) &= 0, \quad \forall s \in \mathcal{S} \\
\widetilde{Q}^*_h(s,a;r) &= \min\{r_h(s,a) + \mathbb{P}_h \widetilde{V}^*_{h+1}(s,a;r), 1\}, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A} \\
\widetilde{V}^*_h(s;r) &= \max_{a \in \mathcal{A}} \widetilde{Q}^*_h(s,a;r). \quad \forall s \in \mathcal{S}, h \in [H].
\end{aligned}$$

The good event $\mathcal{E}^E_{k,h}$ at stage h of episode k in exploration phase is defined to be:

$$\mathcal{E}^E_{k,h} = \Big\{ \lambda + \sum_{i \in [k-1]} \frac{1}{\bar{\sigma}^2_{i,h}} \Big(\widehat{f}_{k,h}(s^i_h, a^i_h) - \mathcal{T}_h V_{k,h+1}(s^i_h, a^i_h) \Big)^2 \leqslant (\beta^E)^2 \Big\}.$$

The intersection of all good events in exploration phase is:

$$\mathcal{E}^E := \bigcap_{k \geqslant 1, h \in [H]} \mathcal{E}^E_{k,h}.$$

The following lemma indicates that $\mathcal{E}$ holds with high probability for GFA-RFE.

Lemma 3.8.2. In Algorithm 6, for any $\delta \in (0,1)$ and fixed $h \in [H]$, with probability at least $1-\delta$, $\mathcal{E}^E$ holds.

In the planning phase, we define the good events for exploration phase with indicator functions as

$$\overline{\mathcal{E}}_h^P = \Big\{\lambda + \sum_{i\in[K]} \frac{\mathbb{1}_h}{\bar{\sigma}_{i,h}^2}\Big(\widehat{f}_h(s_h^i, a_h^i) - \mathcal{T}_h\widehat{V}_{h+1}(s_h^i, a_h^i)\Big)^2 \leqslant (\widehat{\beta}^P)^2\Big\},$$
$$\overline{\mathcal{E}}^P = \bigcap_{h\in[H]} \overline{\mathcal{E}}_h^P,$$

where $\widehat{\mathbb{1}}_h = \mathbb{1}(V_{h+1}^*(s) \leqslant \widehat{V}_{h+1}(s),\ \forall s \in \mathcal{S}) \cdot \mathbb{1}(\widehat{V}_{h+1}(s) \leqslant V_{k,h+1}(s) + V^*(s;r),\ \forall s \in \mathcal{S}) \cdot \mathbb{1}([\mathbb{V}_h(\widehat{V}_{h+1} - V_{h+1}^*)](s_h^k, a_h^k) \leqslant \eta^{-1}\bar{\sigma}_{k,h}^2,\ \forall k \in [K])$ and $\eta = \log N_{\mathcal{V}}(\epsilon)$. Like in the exploration phase, we also have that $\overline{\mathcal{E}}^P$ holds with high probability for GFA-RFE.

Lemma 3.8.3. In Algorithm 6, for any $\delta \in (0,1)$ and fixed $h \in [H]$, with probability at least $1-\delta$, $\overline{\mathcal{E}}^P$ holds.

Furthermore, we have the following good events in the planning phase without indicator function:

$$\mathcal{E}_h^P = \Big\{\lambda + \sum_{i=1}^{k-1} \frac{1}{(\bar{\sigma}_{i,h'})^2}\Big(\widehat{f}_{h'}(s_{h'}^i, a_{h'}^i) - \mathcal{T}_{h'}V_{h'+1}(s_{h'}^i, a_{h'}^i)\Big)^2 \leqslant (\beta^P)^2,\ \forall h \leqslant h' \leqslant H, k \in [K]\Big\}.$$

And we define $\mathcal{E}^P := \mathcal{E}_1^P$. We shows that $\mathcal{E}^P$ holds if both $\mathcal{E}^E, \overline{\mathcal{E}}^P$ hold with the help of the following lemma:

Lemma 3.8.4. If the event $\mathcal{E}^E, \overline{\mathcal{E}}^P, \mathcal{E}_{h+1}^P$ all hold, then event $\mathcal{E}_h^P$ holds.

Since $\mathcal{E}_H^P$ holds trivially, Lemma 3.8.4 indicates that $\mathcal{E}^P$ holds.

3.8.1.2 Covering Number

The optimistic value functions at stage $h \in [H]$ in our construction belong to the following function class:

$$\mathcal{V}_h = \left\{V(\cdot) = \max_{a\in\mathcal{A}} \min\left(1, f(\cdot, a) + \beta \cdot b(\cdot, a)\right) \middle| f \in \mathcal{F}_h, b \in \mathcal{B}\right\}. \tag{3.8.1}$$

Lemma 3.8.5 (ϵ-covering number of optimistic value function classes)**.** For optimistic value function class $\mathcal{V}_{k,h}$ defined in (3.8.1), we define the distance between two value functions V_1 and V_2 as $\|V_1 - V_2\|_\infty := \max_{s\in\mathcal{S}} |V_1(s) - V_2(s)|$. Then the ϵ-covering number with respect to the distance function can be upper bounded by

$$N_{\mathcal{V}_h}(\epsilon) := N_{\mathcal{F}_h}(\epsilon/2) \cdot N_{\mathcal{B}}(\epsilon/2\beta). \tag{3.8.2}$$

Lemma 3.8.5 further indicates that

$$N_{\mathcal{V}}(\epsilon) = \max_{h\in[H]} N_{\mathcal{V}_h}(\epsilon) = \max_{h\in[H]} N_{\mathcal{F}_h}(\epsilon/2) \cdot N_{\mathcal{B}}(\epsilon/2\beta) = N_{\mathcal{F}}(\epsilon/2) \cdot N_{\mathcal{B}}(\epsilon/2\beta).$$

3.8.1.3 Proof of Theorems

We first introduce the following lemmas to build the path to Theorem 3.5.1.

Lemma 3.8.6. On the event $\mathcal{E}^P$, we have

$$|\widehat{f}_h(s,a) - \mathcal{T}_h\widehat{V}_{h+1}| \leqslant \beta^P D_{\mathcal{F}_h}(z; z_{[K],h}, \bar{\sigma}_{[K],h}).$$

Lemma 3.8.7 (Optimism in the planning phase)**.** On the event $\mathcal{E}^P$, for any $h \in [H]$, we have

$$V_h^*(s;r) \leqslant \widehat{V}_h(s), \quad \forall s \in \mathcal{S}.$$

Lemma 3.8.8. On the event $\underline{\mathcal{E}}^E$, with probability at least $1 - 3\delta$, we have

$$\sum_{k=1}^K V_{k,1}(s_1^k) = O(\beta^E\sqrt{\dim_{\alpha,K}(\mathcal{F})}H\sqrt{K}).$$

Lemma 3.8.9. With probability $1 - \delta$, we have

$$\Big|\sum_{k=1}^K \big(\mathbb{E}_{s\sim\mu}\big[\widetilde{V}_1^*(s;r_k)\big] - \widetilde{V}_1^*(s;r_k)\big)\Big| \leqslant \sqrt{2K\log(1/\delta)}.$$

We denote the event that Lemma 3.8.8 holds as Φ, and the event that Lemma 3.8.9 holds as Ψ.

Lemma 3.8.10. Under event $\mathcal{E}^E \cap \Phi \cap \Psi$, we have

$$\mathbb{E}_{s\sim\mu}\Big[\widetilde{V}_1^*(s;b)\Big] = O\Big(\beta^E\sqrt{H\dim_{\alpha,K}(\mathcal{F})/K}\sqrt{\log N_{\mathcal{F}}(\epsilon)/\log N_{\mathcal{V}}(\epsilon)}\Big),$$

where $b = \{b_h\}_{h=1}^H$ is the UCB bonus in planning phase.

With these lemmas, we are ready to prove Theorem 3.5.1.

Proof of Theorem 3.5.1. By Lemma 3.8.7, we can upper bound the suboptimality as

$$\mathbb{E}_{s_1\sim\mu}[V_1^*(s_1;r) - V_1^\pi(s_1;r)] \leqslant \mathbb{E}_{s_1\sim\mu}[\widehat{V}_1(s_1) - V_1^\pi(s_1;r)].$$

Then, we can decompose the difference between optimistic estimate of value function and the true value function in the following:

$$\begin{aligned}
&\mathbb{E}_{s_1\sim\mu}[\widehat{V}_1(s_1) - V_1^\pi(s_1;r)] \\
&= \mathbb{E}_{s_1\sim\mu}\Big[\min\{\widehat{f}_1(s_1,\pi(s_1)) + b_1(s_1,\pi(s_1)),1\} - r_1(s_1,\pi(s_1)) - \mathbb{P}_1V_2^\pi(s_1,\pi(s_1);r)\Big] \\
&\leqslant \mathbb{E}_{s_1\sim\mu}\Big[\min\{\widehat{f}_1(s_1,\pi(s_1)) + b_1(s_1,\pi(s_1)) - r_1(s_1,\pi(s_1)) - \mathbb{P}_1V_2^\pi(s_1,\pi(s_1);r),1\}\Big] \\
&= \mathbb{E}_{s_1\sim\mu}\Big[\min\Big\{\widehat{f}_1(s_1,\pi(s_1)) - r_1(s_1,\pi(s_1)) - \mathbb{P}_1\widehat{V}_2^\pi(s_1,\pi(s_1);r) \\
&\quad + \mathbb{P}_1\widehat{V}_2^\pi(s_1,\pi(s_1);r) + b_1(s_1,\pi(s_1)) - \mathbb{P}_1V_2^\pi(s_1,\pi(s_1);r),1\Big\}\Big] \\
&= \mathbb{E}_{s_1\sim\mu}\Big[\min\Big\{\widehat{f}_1(s_1,\pi(s_1)) - \mathcal{T}_1\widehat{V}_2^\pi(s_1,\pi(s_1)) + \mathbb{P}_1\widehat{V}_2^\pi(s_1,\pi(s_1);r) \\
&\qquad + b_1(s_1,\pi(s_1)) - \mathbb{P}_1V_2^\pi(s_1,\pi(s_1);r),1\Big\}\Big] \\
&\leqslant \mathbb{E}_{s_1\sim\mu}\Big[\min\Big\{2b_1(s_1,\pi(s_1)) + \mathbb{P}_1\widehat{V}_2^\pi(s_1,\pi(s_1);r) - \mathbb{P}_1V_2^\pi(s_1,\pi(s_1);r),1\Big\}\Big],
\end{aligned}$$

where the last inequality holds due to Lemma 3.8.6. Then, by the induction, we have

$$
\begin{aligned}
&\mathbb{E}_{s_1\sim\mu}[\widehat{V}_1(s_1) - V_1^{\pi}(s_1;r)] \\
&\leqslant \mathbb{E}_{s_1\sim\mu}\Big[\min\Big\{2b_1(s_1,\pi(s_1)) + \mathbb{P}_1\widehat{V}_2^{\pi}(s_1,\pi(s_1);r) - \mathbb{P}_1 V_2^{\pi}(s_1,\pi(s_1);r), 1\Big\}\Big] \\
&= \mathbb{E}_{s_1\sim\mu, s_2\sim\mathbb{P}(\cdot|s_1,\pi(s_1))}\Big[\min\Big\{2b_1(s_1,\pi(s_1)) + \widehat{V}_2^{\pi}(s_2;r) - V_2^{\pi}(s_2;r), 1\Big\}\Big] \\
&\leqslant \mathbb{E}_{\tau\sim d^{\pi}}\Big[\min\Big\{\sum_{h=1}^{H} 2b_h(s_h,\pi(s_h)), 1\Big\}\Big] \\
&\leqslant 2\mathbb{E}_{s_1\sim\mu}\Big[\widetilde{V}_1^{\pi}(s_1;b)\Big] \\
&\leqslant 2\mathbb{E}_{s_1\sim\mu}\Big[\widetilde{V}_1^{*}(s_1;b)\Big] \\
&= O\Big(\beta^E\sqrt{H\dim_{\alpha,K}(\mathcal{F})/K}\sqrt{\log N_{\mathcal{F}}(\epsilon)/\log N_{\mathcal{V}}(\epsilon)}\Big).
\end{aligned}
$$

Therefore, by substituting $\beta^E = \widetilde{O}\big(\sqrt{H\log N_{\mathcal{V}}(\epsilon)}\big)$, we complete the proof:

$$
\mathbb{E}_{s_1\sim\mu}[V_1^{*}(s_1;r) - V_1^{\pi}(s_1;r)] = O\Big(H\sqrt{\dim_{\alpha,K}(\mathcal{F})/K}\sqrt{\log N_{\mathcal{F}}(\epsilon)}\Big).
$$

□

Proof of Corollary 3.5.2. By solving $\mathbb{E}_{s_1\sim\mu}[V_1^{*}(s_1;r) - V_1^{\pi}(s_1;r)] \leqslant \epsilon$, we have that

$$
K \geqslant \frac{H^2\log N_{\mathcal{F}}(\epsilon)\dim_{\alpha,K}(\mathcal{F})}{\epsilon^2}.
$$

□

3.8.2 Proof of Lemmas in Section 3.8.1

In this section, we prove the lemmas used in Section 3.8.1.

Proof of Lemma 3.8.2. We first prove that $\mathcal{E}_{k,h}^{E}$ holds with probability $1-\delta/(KH)$. We have $\mathcal{T}_h V_{k,h+1} \in \mathcal{F}_h$ due to Assumption 3.3.2. For any function $V : S \to [0,1]$, let $\eta_h^k(V) =$

$r_{k,h}(s_h^k, a_h^k) + V(s_{h+1}^k) - \mathcal{T}_h V(s_h^k, a_h^k)$. For all $f \in \mathcal{F}_h$, since $a^2 - 2ab = (a-b)^2 - b^2$, we have

$$\begin{aligned}
&\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i, a_h^i)\Big)^2 \\
&\quad - 2\underbrace{\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i, a_h^i)\Big)\eta_h^k(V_{k,h+1})}_{I(f, \mathcal{T}_h V_{k,h+1}, V_{k,h+1})} \\
&= \sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(r_{k,h}(s_h^i, a_h^i) + V_{k,h+1}(s_{h+1}^i) - f(s_h^i, a_h^i)\Big)^2 - \sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\eta_h^k(V_{k,h+1})^2.
\end{aligned}$$

Take $f = \widehat{f}_{k,h}$. By the the definition of $\widehat{f}_{k,h}$, we have

$$\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_{k,h}(s_h^i, a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i, a_h^i)\Big)^2 - 2I(\widehat{f}_{k,h}, \mathcal{T}_h V_{k,h+1}, V_{k,h+1}) \leqslant 0$$

Applying Lemma 3.8.19, for fixed f, $\bar{f}$, and V, with probability at least $1-\delta$,

$$\begin{aligned}
I(f, \bar{f}, V) &:= \sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)\eta_h^k(V) \\
&\leqslant \frac{2\tau}{\alpha^2} \sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)^2 + \frac{1}{\tau}\cdot\log\frac{1}{\delta}.
\end{aligned}$$

Applying a union bound and take $\tau = \frac{\alpha^2}{8}$, for any k, with probability at least $1-\delta$, we have for all V^c in the ϵ-net $\mathcal{V}_{h+1}$ that

$$I(f, \bar{f}, V^c) \leqslant \frac{1}{4} \sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)^2 + \frac{2}{\alpha^2}\cdot\log\frac{N_{\mathcal{V}}(\epsilon)}{\delta}$$

For all V such that $\|V - V^c\|_\infty \leqslant \epsilon$, we have $|\eta_h^i(V) - \eta_h^i(V^c)| \leqslant 4\epsilon$. Thus,

$$I(f, \bar{f}, V_{k,h+1}) \leqslant \frac{1}{4} \sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)^2 + \frac{2}{\alpha^2}\cdot\log\frac{N_{\mathcal{V}}(\epsilon)}{\delta} + 4\epsilon\cdot kL/\alpha^2$$

Applying a union bound, for any k, with probability at least $1-\delta$, we have for all f^a, f^b in the ϵ-net $\mathcal{C}(\mathcal{F}_h, \epsilon)$ that

$$\begin{aligned}
I(f^a, f^b, V_{k,h+1}) \leqslant \frac{1}{4} &\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f^a(s_h^i, a_h^i) - f^b(s_h^i, a_h^i)\Big)^2 \\
&+ \frac{2}{\alpha^2}\cdot\log\frac{N_{\mathcal{V}}(\epsilon)\cdot N_{\mathcal{F}}(\epsilon)^2}{\delta} + 4\epsilon\cdot kL/\alpha^2.
\end{aligned}$$

Therefore, with probability at least $1-\delta$, we have

$$
\begin{aligned}
&I(\widehat{f}_{k,h}, \mathcal{T}_h V_{k,h+1}, V_{k,h+1}) \leqslant I(f^a, f^b, V_{k,h+1}) + 8\epsilon \cdot k/\alpha^2 \\
&\leqslant \frac{1}{4}\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(f^a(s_h^i,a_h^i) - f^b(s_h^i,a_h^i)\Big)^2 + \frac{4}{\alpha^2}\cdot\log\frac{N_{\mathcal{V}}(\epsilon)\cdot N_{\mathcal{F}}(\epsilon)}{\delta} + \frac{4\epsilon kL + 8\epsilon k}{\alpha^2} \\
&\leqslant \frac{1}{4}\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_{k,h}(s_h^i,a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i,a_h^i)\Big)^2 + \frac{4}{\alpha^2}\cdot\log\frac{N_{\mathcal{V}}(\epsilon)\cdot N_{\mathcal{F}}(\epsilon)}{\delta} + 4\epsilon\cdot kL/\alpha^2 \\
&\qquad + 8\epsilon\cdot k/\alpha^2 + 2L\epsilon\cdot k/\alpha^2 \\
&\leqslant \frac{1}{4}\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_{k,h}(s_h^i,a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i,a_h^i)\Big)^2 + \frac{4}{\alpha^2}\cdot\log\frac{N_{\mathcal{V}}(\epsilon)\cdot N_{\mathcal{F}}(\epsilon)}{\delta} + 14L\epsilon\cdot k/\alpha^2.
\end{aligned}
$$

Substituting it back, with probability at least $1-\delta/(KH)$, we have

$$
\frac{1}{4}\sum_{i\in[k-1]} \frac{1}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_{k,h}(s_h^i,a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i,a_h^i)\Big)^2 \leqslant \frac{16}{\alpha^2}\cdot\log\frac{KH\cdot N_{\mathcal{V}}(\epsilon)\cdot N_{\mathcal{F}}(\epsilon)}{\delta} + 56L\epsilon k/\alpha^2
$$

Take $\alpha = 1/\sqrt{H}$ and let

$$
\beta^E = \sqrt{16H\log\frac{KH\cdot N_{\mathcal{V}}(\epsilon)\cdot N_{\mathcal{F}}(\epsilon)}{\delta} + 56L\epsilon\cdot K/\alpha^2 + \lambda}.
$$

Then we complete the proof by taking a union bound for all $k\in[K]$ and $h\in[H]$. □

Proof of Lemma 3.8.3. We have $\mathcal{T}_h\widehat{V}_{h+1}\in\mathcal{F}_h$ due to Assumption 3.3.2. For any function $V:\ S\to[0,1]$, let $\eta_h^k(V) = r_h(s_h^k,a_h^k) + V(s_{h+1}^k) - \mathcal{T}_h V(s_h^k,a_h^k)$. For all $f\in\mathcal{F}_h$, we have

$$
\begin{aligned}
&\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i,a_h^i) - \mathcal{T}_h\widehat{V}_{h+1}(s_h^i,a_h^i)\Big)^2 - 2\underbrace{\sum_{i\in[K]}\frac{\widehat{\mathbb{1}}_h\big(f(s_h^i,a_h^i) - \mathcal{T}_h\widehat{V}_{h+1}(s_h^i,a_h^i)\big)\eta_h^k(\widehat{V}_{h+1})}{(\bar{\sigma}_{i,h})^2}}_{I(f,\mathcal{T}_h\widehat{V}_{h+1},\widehat{V}_{h+1})} \\
&= \sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(r_h(s_h^i,a_h^i) + \widehat{V}_{h+1}(s_{h+1}^i) - f(s_h^i,a_h^i)\Big)^2 - \sum_{i\in[K]}\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\eta_h^k(\widehat{V}_{h+1})^2.
\end{aligned}
$$

By definition, we have that

$$
\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_h(s_h^i,a_h^i) - \mathcal{T}_h\widehat{V}_{h+1}(s_h^i,a_h^i)\Big)^2 - 2I(\widehat{f}_h, \mathcal{T}_h\widehat{V}_{h+1}, \widehat{V}_{h+1}) \leqslant 0.
$$

We decompose $I(\widehat{f}_h, \mathcal{T}_h\widehat{V}_{h+1}, \widehat{V}_{h+1})$ into two parts:

$$I(\widehat{f}_h, \mathcal{T}_h\widehat{V}_{h+1}, \widehat{V}_{h+1}) = \sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_h(s_h^i, a_h^i) - \mathcal{T}_h\widehat{V}_{h+1}(s_h^i, a_h^i)\Big)\eta_h^k(\widehat{V}_{h+1} - V_{h+1}^*) + \sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_h(s_h^i, a_h^i) - \mathcal{T}_h\widehat{V}_{h+1}(s_h^i, a_h^i)\Big)\eta_h^k(V_{h+1}^*). \tag{3.8.3}$$

For the first term in (3.8.3), we have

$$\mathbb{E}\bigg[\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)\eta_h^k(\widehat{V}_{h+1} - V_{h+1}^*)\bigg] = 0.$$

Furthermore, we can bound the maximum as following:

$$\begin{aligned}
&\max_{i\in[K]}\bigg|\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)\eta_h^k(\widehat{V}_{h+1} - V_{h+1}^*)\bigg| \\
&\leqslant 2\max_{i\in[K]}\bigg|\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)\bigg| \\
&\leqslant 2\max_{i\in[K]}\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\sqrt{D_{\mathcal{F}_h}^2(z_{i,h}; z_{[i-1],h}, \bar{\sigma}_{[i-1],h})\bigg(\sum_{s\in[i-1]}\frac{1}{(\bar{\sigma}_{s,h})^2}(f(s_h^s, a_h^s) - \bar{f}(s_h^s, a_h^s))^2 + \lambda\bigg)} \\
&\leqslant 2\max_{i\in[K]}\frac{1}{(\bar{\sigma}_{i,h})^2}\sqrt{D_{\mathcal{F}_h}^2(z_{i,h}; z_{[i-1],h}, \bar{\sigma}_{[i-1],h})\bigg(\sum_{s\in[i-1]}\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{s,h})^2}(f(s_h^s, a_h^s) - \bar{f}(s_h^s, a_h^s))^2 + \lambda\bigg)} \\
&\leqslant 2\cdot\gamma^{-2}\sqrt{\sum_{s\in[K]}\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{s,h})^2}(f(s_h^s, a_h^s) - \bar{f}(s_h^s, a_h^s))^2 + \lambda},
\end{aligned}$$

where the first inequality is due to bounded total rewards assumption, the second inequality holds due to Definition 3.3.3, and the last inequality holds due to Line 14 in Algorithm 6 and Definition 3.3.6.

We further define $\mathrm{var}(V - V_{h+1}^*)$ as

$$\begin{aligned}
\mathrm{var}(V - V_{h+1}^*) &:= \sum_{i\in[K]}\mathbb{E}\bigg[\frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^4}\Big(f(s_h^i, a_h^i) - \bar{f}(s_h^i, a_h^i)\Big)^2\eta_h^k(\widehat{V}_{h+1} - V_{h+1}^*)^2\bigg] \\
&\leqslant L^2K/\alpha^4.
\end{aligned}$$

By the definition of the indicator function, we have

$$\mathrm{var}(V - V^*_{h+1}) \leqslant \frac{4}{\eta} \sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2} \Big(f(s^i_h, a^i_h) - \bar{f}(s^i_h, a^i_h) \Big)^2$$

For fixed f, $\bar{f}$, by applying Lemma 3.8.20 with $V^2 = L^2K/\alpha^4, M = 2L/\alpha^2, v = \eta^{-1/2}, m = v^2$, and probability at least $1 - \delta/(N_{\mathcal{F}}(\epsilon)^2 N_{\mathcal{V}}(\epsilon)H)$ we have

$$\begin{aligned}
&\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2} \Big(f(s^i_h, a^i_h) - \bar{f}(s^i_h, a^i_h) \Big) \eta^k_h (V - V^*_{h+1}) \\
&\leqslant \iota \sqrt{2\left(2\mathrm{var}(V - V^*_{h+1}) + \eta^{-1}\right)} \\
&\quad + \frac{2}{3}\iota^2 \left(4\gamma^{-2} \sqrt{\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2} \left(f(s^i_h, a^i_h) - \bar{f}(s^i_h, a^i_h) \right)^2 + \lambda} + \eta^{-1} \right),
\end{aligned}$$

where

$$\iota^2(k, h, \delta) := \log \frac{N_{\mathcal{F}}(\epsilon)^2 \cdot N_{\mathcal{V}}(\epsilon) \cdot (\log(L^2K\eta/\alpha^4) + 2) \cdot (\log(2L\eta/\alpha^2) + 2)}{\delta/H}$$

Using a union bound over all $(f, \bar{f}, V) \in \mathcal{C}(\mathcal{F}_h, \epsilon) \times \mathcal{C}(\mathcal{F}_h, \epsilon) \times \mathcal{C}(\mathcal{V}_{h=1}, \epsilon)$, we have the inequality above holds for all such $f, \bar{f}, V$ with probability at least $1 - \delta/H$. There exist a V^c_{h+1} in the ϵ-net such that $\|\widehat{V}_{h+1} - V^c_{h+1}\| \leqslant \epsilon$. Then we have

$$\begin{aligned}
&\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2} \Big(\widehat{f}_h(s^i_h, a^i_h) - \mathcal{T}_h \widehat{V}_{h+1}(s^i_h, a^i_h) \Big) \eta^k_h (\widehat{V}_{h+1} - V^*_{h+1}) \\
&\leqslant O\left(\iota(k, h, \delta)\eta^{-1/2} + \iota(k, h, \delta)^2 \gamma^{-2} \right) \cdot \sqrt{\sum_{\tau\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{\tau,h})^2} (\widehat{f}_h(s^\tau_h, a^\tau_h) - \mathcal{T}_h V_{h+1}(s^\tau_h, a^\tau_h))^2 + \lambda} \\
&\quad + O(\epsilon k L/\alpha^2) + O(\iota^2(k, h, \delta)\eta^{-1}) + O(\iota(k, h, \delta)\eta^{-1/2}). \qquad (3.8.4)
\end{aligned}$$

For the second term in (3.8.3), applying Lemma 3.8.19, for fixed f, $\bar{f}$, and V^*_{h+1}, with probability at least $1 - \delta$, we have

$$\begin{aligned}
&\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2} \Big(f(s^i_h, a^i_h) - \bar{f}(s^i_h, a^i_h) \Big) \eta^k_h (V^*_{h+1}) \\
&\leqslant \frac{1}{4} \sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2} \Big(f(s^i_h, a^i_h) - \bar{f}(s^i_h, a^i_h) \Big)^2 + \frac{8}{\alpha^2} \cdot \log \frac{1}{\delta}.
\end{aligned}$$

Applying a union bound, for any k, with probability at least $1-\delta$, we have for all f^a, f^b in the ϵ-net $\mathcal{F}_h$

$$I(f^a, f^b, V^*_{h+1}) \leqslant \frac{1}{4}\sum_{i\in[k-1]} \frac{\widehat{\mathbb{1}}_{i,h}}{(\bar{\sigma}_{i,h})^2}\Big(f^a(s^i_h,a^i_h) - f^b(s^i_h,a^i_h)\Big)^2 + \frac{8}{\alpha^2}\cdot\log\frac{N_{\mathcal{F}}(\epsilon)^2}{\delta}.$$

Therefore, with probability at least $1-\delta$, we have

$$\begin{aligned}
&I(\widehat{f}_h, \mathcal{T}_h V_{h+1}, V^*_{h+1}) \leqslant I(f^a, f^b, V^*_{h+1}) + 8\epsilon\cdot K/\alpha^2\\
&\leqslant \frac{1}{4}\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(f^a(s^i_h,a^i_h) - f^b(s^i_h,a^i_h)\Big)^2 + \frac{8}{\alpha^2}\cdot\log\frac{\cdot N_{\mathcal{F}}(\epsilon)^2}{\delta} + 8\epsilon\cdot k/\alpha^2\\
&\leqslant \frac{1}{4}\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_h(s^i_h,a^i_h) - \mathcal{T}_h V_{h+1}(s^i_h,a^i_h)\Big)^2 + \frac{8}{\alpha^2}\cdot\log\frac{N_{\mathcal{F}}(\epsilon)^2}{\delta}\\
&\quad + 8\epsilon\cdot k/\alpha^2 + 2L\epsilon\cdot k/\alpha^2\\
&\leqslant \frac{1}{4}\sum_{i\in[K]} \frac{\widehat{\mathbb{1}}_h}{(\bar{\sigma}_{i,h})^2}\Big(\widehat{f}_h(s^i_h,a^i_h) - \mathcal{T}_h V_{h+1}(s^i_h,a^i_h)\Big)^2 + \frac{8}{\alpha^2}\cdot\log\frac{N_{\mathcal{F}}(\epsilon)^2}{\delta} + 10L\epsilon\cdot k/\alpha^2. \qquad (3.8.5)
\end{aligned}$$

Taking $\eta = \log N_{\mathcal{V}}(\epsilon)$, $\gamma = \widetilde{O}\big(\sqrt{\log N_{\mathcal{V}}(\epsilon)}\big)$ and $\alpha = 1/\sqrt{H}$ and substituting (3.8.4) and (3.8.5) back into (3.8.3), we have

$$\begin{aligned}
&\lambda + \sum_{i\in[K]} \frac{\mathbb{1}_h}{\bar{\sigma}^2_{i,h}}\Big(\widehat{f}_h(s^i_h,a^i_h) - \mathcal{T}_h\widehat{V}_{h+1}(s^i_h,a^i_h)\Big)^2\\
&\leqslant O\bigg(H\log N_{\mathcal{F}}(\epsilon)\bigg) + O\bigg((\log N_{\mathcal{V}}(\epsilon))^{-1}\log\frac{(\log(L^2K/\alpha^4)+2)\cdot(\log(2L/\alpha^2)+2)}{\delta/H}\bigg) + O(\lambda).
\end{aligned}$$

□

Lemma 3.8.11. On the event $\mathcal{E}^E \cap \mathcal{E}^P_h$, for any $h\in[H]$, we have

$$V^*_h(s;r) + V_{k,h}(s) \geqslant \widehat{V}_h(s). \qquad (3.8.6)$$

Lemma 3.8.12. On the event $\mathcal{E}^E \cap \mathcal{E}^P_{h+1}$, for each episode $k\in[K]$, we have

$$\log N_{\mathcal{V}}(\epsilon)\cdot[\mathbb{V}_h(\widehat{V}_{h+1} - V^*_{h+1})](s^k_h,a^k_h) \leqslant \sigma^2_{k,h},$$

where $\sigma^2_{k,h} = 4\log N_{\mathcal{V}}(\epsilon)\cdot\min\{\widehat{f}_{k,h}(s^k_h,a^k_h), 1\}$.

Proof of Lemma 3.8.4. Recall that the indicator function in event $\overline{\mathcal{E}}^P$ is

$$\widehat{\mathbb{1}}_h = \underbrace{\mathbb{1}(V^*_{h+1}(s) \leqslant \widehat{V}_{h+1}(s),\ \forall s \in \mathcal{S})}_{I_1} \cdot \underbrace{\mathbb{1}(\widehat{V}_{h+1}(s) \leqslant V_{k,h+1}(s) + V^*(s;r),\ \forall s \in \mathcal{S}, \forall k \in [K])}_{I_2}$$
$$\cdot \underbrace{\mathbb{1}([\mathbb{V}_h(\widehat{V}_{h+1} - V^*_{h+1})](s^k_h, a^k_h) \leqslant \eta^{-1}\bar{\sigma}^2_{k,h},\ \forall k \in [K])}_{I_3},$$

where $\eta = \log N_{\mathcal{V}}(\epsilon)$. Lemma 3.8.11, Lemma 3.8.7, and Lemma 3.8.12 indicate that $I_1 = I_2 = I_3 = 1$. □

Proof of Lemma 3.8.5. There exists an $\epsilon/2$-net of $\mathcal{F}$, denoted by $\mathcal{C}(\mathcal{F}_h, \epsilon/2)$, such that for any $f \in \mathcal{F}_h$, we can find $f' \in \mathcal{C}(\mathcal{F}, \epsilon/2)$ such that $\|f - f'\|_\infty \leqslant \epsilon/2$. Also, there exists an $\epsilon/2\beta$-net of $\mathcal{B}$, $\mathcal{C}(\mathcal{B}, \epsilon/2\beta)$.

Then we consider the following subset of $\mathcal{V}_h$,

$$\mathcal{V}^c_h = \left\{ V(\cdot) = \max_{a \in \mathcal{A}} \min\left(1, f(\cdot, a) + \beta \cdot b(\cdot, a)\right) \middle| f \in \mathcal{C}(\mathcal{F}_h, \epsilon/2), b \in \mathcal{C}(\mathcal{B}, \epsilon/2\beta) \right\}.$$

Consider an arbitrary $V \in \mathcal{V}$ where $V = \max_{a\in\mathcal{A}} \min(1, f_i(\cdot, a) + \beta \cdot b_i(\cdot, a))$. For each f_i, there exists $f^c_i \in \mathcal{C}(\mathcal{F}_h, \epsilon/2)$ such that $\|f_i - f^c_i\|_\infty \leqslant \epsilon/2$. There also exists $b^c \in \mathcal{C}(\mathcal{B}, \epsilon/2\beta)$ such that $\|b_i - b^c\|_\infty \leqslant \epsilon/2\beta$. Let $V^c = \max_{a\in\mathcal{A}} \min(1, f^c_i(\cdot, a) + \beta \cdot b^c(\cdot, a)) \in \mathcal{V}^c$. It is then straightforward to check that $\|V - V^c\|_\infty \leqslant \epsilon/2 + \beta \cdot \epsilon/2\beta = \epsilon$.

By direct calculation, we have $|\mathcal{V}^c_h| = N_{\mathcal{F}_h}(\epsilon/2) \cdot N_{\mathcal{B}}(\epsilon/2\beta)$. □

Proof of Lemma 3.8.6. According to the definition of $D^2_{\mathcal{F}}$ function, we have

$$\begin{aligned}
&\left(\widehat{f}_{k,h}(s,a) - \mathcal{T}_h V_{k,h+1}(s,a)\right)^2 \\
&\leqslant D^2_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}) \times \left(\lambda + \sum_{i=1}^{k-1} \frac{1}{(\bar{\sigma}_{i,h})^2} \left(\widehat{f}_{k,h}(s^i_h, a^i_h) - \mathcal{T}_h V_{k,h+1}(s^i_h, a^i_h)\right)^2\right) \\
&\leqslant \left(\beta^E\right)^2 \times D^2_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}),
\end{aligned}$$

where the first inequality holds due the definition of $D^2_{\mathcal{F}}$ function with the Assumption 3.3.2 and the second inequality holds due to the events $\mathcal{E}^E_h$. Thus, we have

$$\left|\widehat{f}_{k,h}(s,a) - \mathcal{T}_h V_{k,h+1}(s,a)\right| \leqslant \beta^E D_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}).$$

□

Proof of Lemma 3.8.7. We prove this statement by induction. Note that $V^*_{H+1}(s;r) = \widehat{V}_{H+1}(s)$. Assume that the statement holds for $h+1$. If $\widehat{V}_h(s) = 1$, then the statement holds trivially for h; otherwise, we have for any $(s,a) \in \mathcal{S} \times \mathcal{A}$ that

$$
\begin{aligned}
&\widehat{Q}_h(s,a) - Q^*_h(s,a;r) \\
&= \widehat{f}_h(s,a) + b_h(s,a) - [r_h(s,a;r) + \mathbb{P}_h V^*_{h+1}(s,a;r)] \\
&= [\widehat{f}_h(s,a) - r_h(s,a;r) - \mathbb{P}_h \widehat{V}_{h+1}(s,a;r)] + b_h(s,a) + \mathbb{P}_h \widehat{V}_{h+1}(s,a;r) - \mathbb{P}_h V^*_{h+1}(s,a;r) \\
&\geqslant [\widehat{f}_h(s,a) - r_h(s,a;r) - \mathbb{P}_h \widehat{V}_{h+1}(s,a;r)] + b_h(s,a) \\
&\geqslant -\beta^P D_{\mathcal{F}_h}(z; z_{[K],h}, \bar{\sigma}_{[K],h}) + \beta^P \overline{D}_{\mathcal{F}_h}(z; z_{[K],h}, \bar{\sigma}_{[K],h}) \\
&\geqslant 0,
\end{aligned}
$$

where the first inequality holds due to the induction assumption, and the second inequality holds due to Lemma 3.8.6. □

In order to prove Lemma 3.8.8, we need the following three lemmas.

Lemma 3.8.13 (Simulation Lemma)**.** On the event $\underline{\mathcal{E}}^E$, we have

$$
0 \leqslant V_{k,h}(s^k_h) \leqslant \mathbb{E}_{\tau^k_h \sim d^{\pi^k}_h(s^k_h)} \min\Big\{3\beta^E \sum_{h'=h}^{H} \overline{\mathcal{D}}(z_{k,h'}; z_{[k-1],h'}, \overline{\sigma}_{[k-1],h'}), 1\Big\}.
$$

Lemma 3.8.14. [Lemma C.13 in Zhao et al. (2023)]For any parameters $\beta \geqslant 1$ and stage $h \in [H]$, the summation of confidence radius over episode $k \in [K]$ is upper bounded by

$$
\begin{aligned}
&\sum_{k=1}^{K} \min\Big(\beta D_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}), 1\Big) \\
&\leqslant (1 + C\beta\gamma^2) \dim_{\alpha,K}(\mathcal{F}_h) + 2\beta\sqrt{\dim_{\alpha,K}(\mathcal{F}_h)}\sqrt{\sum_{k=1}^{K}(\sigma^2_{k,h} + \alpha^2)},
\end{aligned}
$$

where $z = (s,a)$ and $z_{[k-1],h} = \{z_{1,h}, z_{2,h}, .., z_{k-1,h}\}$.

Lemma 3.8.15. Under event $\underline{\mathcal{E}}^E$, we have

$$\begin{aligned}\sum_{k=1}^{K}\sum_{h=1}^{H}\sigma_{k,h}^2 \leqslant\ & 2304C^2H^3(\log N_{\mathcal{V}}(\epsilon))^2(\beta^E)^2\dim_{\alpha,K}(\mathcal{F}) \\ & + 48H^2\log N_{\mathcal{V}}(\epsilon)(1+C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F}_h) \\ & + 16H\log N_{\mathcal{V}}(\epsilon)\sqrt{2HK\log(H/\delta)} + K.\end{aligned}$$

Now we can prove Lemma 3.8.8.

Proof of Lemma 3.8.8. We have

$$\begin{aligned}\sum_{k=1}^{K}V_{k,1}(s_1^k) &\leqslant \sum_{k=1}^{K}\mathbb{E}_{\tau_h^k\sim d_h^{\pi^k}(s_h^k)}\min\Big\{3\beta^E\sum_{h'=1}^{H}\overline{\mathcal{D}}(z_{k,h};z_{[k],h},\overline{\sigma}_{[k],h}),1\Big\} \\ &\leqslant \sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{E}_{\tau_h^k\sim d_h^{\pi^k}(s_h^k)}\min\Big\{3\beta^E\overline{\mathcal{D}}(z_{k,h};z_{[k-1],h},\overline{\sigma}_{[k-1],h}),1\Big\} \\ &\leqslant H(1+4C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F}_h) + 8\beta^E\sqrt{\dim_{\alpha,K}(\mathcal{F})}\sqrt{H\sum_{k=1}^{K}\sum_{h=1}^{H}(\sigma_{k,h}^2+\alpha^2)} \\ &= O(\beta^E\sqrt{KH\dim_{\alpha,K}(\mathcal{F})}),\end{aligned}$$

where the first inequality follows from Lemma 3.8.13, the third inequality follows from Lemma 3.8.14, and the last equality holds due to Lemma 3.8.15. □

Proof of Lemma 3.8.9. Denote $\Delta_k = \mathbb{E}_{s\sim\mu}\big[\widetilde{V}_1^*(s;r_k)\big] - \widetilde{V}_1^*(s_1^k;r_k)$. By Azuma-Hoeffding inequality (Lemma 3.8.21), we have

$$\Big|\sum_{k=1}^{K}\Delta_k\Big| \leqslant \sqrt{2K\log(1/\delta)}.$$

□

Lemma 3.8.16. On the event $\underline{\mathcal{E}}^E$, for any $k\in[K]$ and $h\in[H]$, we have

$$\widetilde{V}_h^*(s;r_k) \leqslant V_{k,h}(s), \quad \forall s\in\mathcal{S}.$$

Proof of Lemma 3.8.10. Since $\beta^E = O(\sqrt{H \log N_{\mathcal{V}}(\epsilon)})$ and $\beta^P = O(\sqrt{H \log N_{\mathcal{F}}(\epsilon)})$, for some constant c, we have

$$\beta^E \geqslant c\sqrt{\log N_{\mathcal{V}}(\epsilon)/\log N_{\mathcal{F}}(\epsilon)} \cdot \beta^P.$$

Therefore, for any $h \in [H]$, we have $r_{k,h}(\cdot,\cdot) \geqslant r_{K,h}(\cdot,\cdot) \geqslant c\sqrt{\log N_{\mathcal{V}}(\epsilon)/\log N_{\mathcal{F}}(\epsilon)} \cdot b_h(\cdot,\cdot)$. Hence,

$$\begin{aligned}
&c\sqrt{\log N_{\mathcal{V}}(\epsilon)/\log N_{\mathcal{F}}(\epsilon)} \cdot \mathbb{E}_{s\sim\mu}\Big[\widetilde{V}_1^*(s;b)\Big] \\
&= \mathbb{E}_{s\sim\mu}\Big[\widetilde{V}_1^*(s;c\sqrt{\log N_{\mathcal{V}}(\epsilon)/\log N_{\mathcal{F}}(\epsilon)} \cdot b)\Big] \\
&\leqslant \mathbb{E}_{s\sim\mu}\Big[\widetilde{V}_1^*(s;r_k)\Big]/K \\
&= \Big[\sum_{k=1}^K \widetilde{V}_1^*(s_1^k;r_k) + \sum_{k=1}^K \Big[\mathbb{E}_{s\sim\mu}\Big[\widetilde{V}_1^*(s;r_k)\Big] - \widetilde{V}_1^*(s_1^k;r_k)\Big]\Big]/K \\
&\leqslant \Big(\sum_{k=1}^K \widetilde{V}_1^*(s;r_k)\Big)/K + \sqrt{2\log(1/\delta)/K} \\
&\leqslant \Big(\sum_{k=1}^K V_{k,1}(s;r_k)\Big)/K + \sqrt{2\log(1/\delta)/K} \\
&= O\Big(\beta^E\sqrt{H\dim_{\alpha,K}(\mathcal{F})/K}\Big),
\end{aligned}$$

where the second inequality follows from Lemma 3.8.9, and the third inequality follows from Lemma 3.8.16. Therefore, we have

$$\mathbb{E}_{s\sim\mu}\Big[\widetilde{V}_1^*(s;b)\Big] = O\Big(\beta^E\sqrt{H\dim_{\alpha,K}(\mathcal{F})/K}\sqrt{\log N_{\mathcal{F}}(\epsilon)/\log N_{\mathcal{V}}(\epsilon)}\Big).$$

□

3.8.3 Proofs of Lemmas in Section 3.8.2

Proof of Lemma 3.8.11. We see that

$$\begin{aligned}
Q^*(\cdot,\cdot;r) &= r_h(\cdot,\cdot) + \mathbb{P}_h V_{h+1}(\cdot,\cdot;r), \\
Q_{k,h}(\cdot,\cdot) &= \min\{\widehat{f}_{k,h}(\cdot,\cdot) + b_{k,h}(\cdot,\cdot), 1\}, \\
\widehat{Q}_h(\cdot,\cdot) &= \min\{\widehat{f}_h(\cdot,\cdot) + b_h(\cdot,\cdot), 1\}.
\end{aligned}$$

We prove this statement by induction. Note that $V^*_{H+1}(s;r) + V_{k,H+1}(s) = \widehat{V}_{H+1}(s) = 0$. Assume the statement holds for $h+1$. By definition, we have

$$Q^*_h(s,a;r) + 1 \geqslant \widehat{Q}_h(s,a).$$

Therefore, we only need to prove

$$Q^*_h(s,a;r) + \widehat{f}_{k,h}(s,a) + b_{k,h}(s,a) - \widehat{Q}_h(s,a) \geqslant 0.$$

We have

$$\begin{aligned}
&Q^*_h(s,a;r) + \widehat{f}_{k,h}(s,a) + b_{k,h}(s,a) - \widehat{Q}_h(s,a) \\
&= r_h(s,a) + \mathbb{P}_h V^*_{h+1}(s,a;r) + \widehat{f}_{k,h}(s,a) + b_{k,h}(s,a) - \min\{\widehat{f}_h(s,a) + b_h(s,a), 1\} \\
&\geqslant r_h(s,a) + \mathbb{P}_h V^*_{h+1}(s,a;r) + \widehat{f}_{k,h}(s,a) + b_{k,h}(s,a) - (\widehat{f}_h(s,a) + b_h(s,a)) \\
&= \mathbb{P}_h V^*_{h+1}(s,a;r) + \mathbb{P}_h V_{k,h+1}(s,a) - \mathbb{P}_h \widehat{V}_{h+1}(s,a) + \widehat{r}_{k,h}(s,a) + b_{k,h}(s,a) - b_h(s,a) \\
&\qquad + (\widehat{f}_{k,h}(s,a) - \widehat{r}_{k,h}(s,a) - \mathbb{P}_h V_{k,h+1}(s,a)) + (r_h(s,a) + \mathbb{P}_h \widehat{V}_h(s,a) - \widehat{f}_h(s,a)) \\
&\geqslant \widehat{r}_{k,h}(s,a) + b_{k,h}(s,a) - b_h(s,a) + (\widehat{f}_{k,h}(s,a) - \widehat{r}_{k,h}(s,a) - \mathbb{P}_h V_{k,h+1}(s,a)) \\
&\qquad + (r_h(s,a) + \mathbb{P}_h \widehat{V}_h(s,a) - \widehat{f}_h(s,a)) \\
&\geqslant 3\beta^E \overline{\mathcal{D}}_{\mathcal{F}_h}(z; z_{[k-1],h}, \overline{\sigma}_{[k-1],h}) - \beta^P \overline{\mathcal{D}}_{\mathcal{F}_h}(z; z_{[K],h}, \overline{\sigma}_{[K],h}) - \beta^E \mathcal{D}_{\mathcal{F}_h}(z; z_{[k-1],h}, \overline{\sigma}_{[k-1],h}) \\
&\qquad - \beta^P \mathcal{D}_{\mathcal{F}_h}(z; z_{[K],h}, \overline{\sigma}_{[K],h}) \\
&\geqslant 0,
\end{aligned}$$

where the second inequality holds due to induction assumption, the third inequality holds by high probability events, and the last inequality holds by $\beta^E \geqslant \beta^P$, $\bar{\mathcal{D}}_{\mathcal{F}_h}(z; z_{[k],h}, \overline{\sigma}_{[k],h})$ decreasing with k, and Definition 3.3.5. □

Lemma 3.8.17. On the event $\underline{\mathcal{E}}^E$, we have

$$|\widehat{f}_{k,h}(s,a) - \mathcal{T}_h V_{k,h+1}| \leqslant \beta^E D_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h})$$

Proof of Lemma 3.8.12. We have Lemma 3.8.7 and 3.8.11 both hold on $\mathcal{E}_{h+1}^P$. Therefore, we have

$$
\begin{aligned}
&[\mathbb{V}_h(\widehat{V}_{h+1} - V_{h+1}^*)](s_h^k, a_h^k) \\
&\leqslant [\mathbb{P}_h(\widehat{V}_{h+1} - V_{h+1}^*)^2](s_h^k, a_h^k) \\
&\leqslant 2[\mathbb{P}_h(\widehat{V}_{h+1} - V_{h+1}^*)](s_h^k, a_h^k) \\
&\leqslant 2[\mathbb{P}_h V_{k,h+1}](s_h^k, a_h^k) \\
&= 2(\mathcal{T}_h V_{k,h+1}(s_h^k, a_h^k) - r_{k,h}(s_h^k, a_h^k)) \\
&\leqslant 2(\widehat{f}_{k,h}(s_h^k, a_h^k) + \beta^E D_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}) - \beta^E \overline{D}_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h})) \\
&\leqslant 2\widehat{f}_{k,h}(s_h^k, a_h^k),
\end{aligned}
$$

where the second inequality holds due to Lemma 3.8.7 and $\widehat{V}_{h+1}, V_{h+1}^* \in [0, 1]$, the third inequality holds due to Lemma 3.8.11, the fourth inequality holds due to Lemma 3.8.17, and the last inequality holds due to Definition 3.3.5. □

Proof of Lemma 3.8.13. According to Algorithm 6, we have that

$$
\begin{aligned}
Q_{k,h}(\cdot, \cdot) &= \min\{\widehat{f}_{k,h}(\cdot, \cdot) + b_{k,h}(\cdot, \cdot), 1\}, \\
V_{k,h}(\cdot) &= \max_a Q_{k,h}(\cdot, a), \\
a_h^k &= \pi_h^k(s_h^k) = \operatorname*{argmax}_a Q_{k,h}(s_h^k, a).
\end{aligned}
$$

For all k and all h, we have that $V_{k,h}(s_h^k) = Q_{k,h}(s_h^k, a_h^k)$ and thus

$$
\begin{aligned}
&V_{k,h}(s_h^k) \\
&\leqslant \widehat{f}_{k,h}(s_h^k, a_h^k) + b_{k,h}(s_h^k, a_h^k) \\
&= 2\beta^E \overline{\mathcal{D}}(z_{k,h}; z_{[k-1],h}, \overline{\sigma}_{[k-1],h}) + (\widehat{f}_{k,h}(s_h^k, a_h^k) - \mathcal{T}_h V_{k,h+1}(s_h^k, a_h^k)) + \mathcal{T}_h V_{k,h+1}(s_h^k, a_h^k) \\
&\dots \\
&= \mathbb{E}_{\tau_h^k \sim d_h^{\pi^k}(s_h^k)} \sum_{h'=h}^{H} \left[(\widehat{f}_{k,h'}(s_{h'}^k, a_{h'}^k) - \mathcal{T}_h V_{k,h'+1}(s_{h'}^k, a_{h'}^k)) + 2\beta^E \overline{\mathcal{D}}(z_{k,h'}; z_{[k-1],h'}, \overline{\sigma}_{[k-1],h'}) \right] \\
&\leqslant \mathbb{E}_{\tau_h^k \sim d_h^{\pi^k}(s_h^k)} \sum_{h'=h}^{H} 3\beta^E \overline{\mathcal{D}}(z_{k,h'}; z_{[k-1],h'}, \overline{\sigma}_{[k-1],h'}),
\end{aligned}
$$

where the last inequality holds due to Lemma 3.8.17 and Definition 3.3.5. □

Lemma 3.8.18. On the event $\underline{\mathcal{E}}^E$, with probability at least $1 - \delta$,

$$
\begin{aligned}
\sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{P}_h V_{k,h+1}(s_h^k, , a_h^k) \leqslant H \sum_{k=1}^{K} \sum_{h=1}^{H} \min \left\{ 4\beta^E D_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}), 1 \right\} \\
+ (H+1)\sqrt{2HK \log(1/\delta)}
\end{aligned}
$$

Proof of Lemma 3.8.15. Recall $\sigma_{k,h}^2 = 4\log N_{\mathcal{V}}(\epsilon)\cdot\min\{\widehat{f}_{k,h}(s_h^k,a_h^k),1\}$. We have

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H}\sigma_{k,h}^2 &= 4\log N_{\mathcal{V}}(\epsilon)\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{\widehat{f}_{k,h}(s_h^k,a_h^k),1\}\\
&\leqslant 4\log N_{\mathcal{V}}(\epsilon)\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{\mathcal{T}_h V_{k,h+1}(s_h^k,a_h^k)+\beta^E D_{\mathcal{F}_h}(z_{k,h};z_{[k-1].h},\bar{\sigma}_{[k-1],h}),1\}\\
&\leqslant 4\log N_{\mathcal{V}}(\epsilon)\sum_{k=1}^{K}\sum_{h=1}^{H}\min\{\mathbb{P}_h V_{k,h+1}(s_h^k,a_h^k)+2\beta^E\overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h};z_{[k-1].h},\bar{\sigma}_{[k-1],h}),1\}\\
&\leqslant 4\log N_{\mathcal{V}}(\epsilon)\sum_{k=1}^{K}\sum_{h=1}^{H}\mathbb{P}_h V_{k,h+1}(s_h^k,a_h^k)\\
&\quad + 8\log N_{\mathcal{V}}(\epsilon)\sum_{k=1}^{K}\sum_{h=1}^{H}\{\beta^E\overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h};z_{[k-1],h},\bar{\sigma}_{[k-1],h}),1\}\\
&\leqslant 24H\log N_{\mathcal{V}}(\epsilon)\underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{\beta^E\overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h};z_{[k-1],h},\bar{\sigma}_{[k-1],h}),1\right\}}_{I}\\
&\quad + 8H\log N_{\mathcal{V}}(\epsilon)\sqrt{2HK\log(H/\delta)},
\end{aligned}$$

where the first inequality holds due to Lemma 3.8.17, the second inequality holds due to Definition 3.3.5, and the last inequality holds due to Lemma 3.8.18. For the term I, we further have

$$\begin{aligned}
&\sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{\beta^E\overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h};z_{[k-1],h},\bar{\sigma}_{[k-1],h}),1\right\}\\
&\leqslant \sum_{k=1}^{K}\sum_{h=1}^{H}\min\left\{C\beta^E\mathcal{D}_{\mathcal{F}_h}(z_{k,h};z_{[k-1],h},\bar{\sigma}_{[k-1],h}),1\right\}\\
&\leqslant \sum_{h=1}^{H}(1+C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F}_h)+2C\beta^E\sum_{h=1}^{H}\sqrt{\dim_{\alpha,K}(\mathcal{F}_h)}\sqrt{\sum_{k=1}^{K}(\sigma_{k,h}^2+\alpha^2)}\\
&\leqslant H(1+C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F})+2C\beta^E\sqrt{\sum_{h=1}^{H}\dim_{\alpha,K}(\mathcal{F}_h)}\sqrt{\sum_{k=1}^{K}\sum_{h=1}^{H}(\sigma_{k,h}^2+\alpha^2)}\\
&\leqslant H(1+C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F}_h)+2C\beta^E\sqrt{\dim_{\alpha,K}(\mathcal{F})}\sqrt{H\sum_{k=1}^{K}\sum_{h=1}^{H}(\sigma_{k,h}^2+\alpha^2)},
\end{aligned}$$

where the first inequality holds due to Definition 3.3.5, the second inequality holds due to Lemma 3.8.14, the third inequality holds due to Cauchy-Schwarz inequality. Therefore, we can get

$$\begin{aligned}\sum_{k=1}^{K}\sum_{h=1}^{H}\sigma_{k,h}^2 \leqslant\ & 24H^2\log N_{\mathcal{V}}(\epsilon)(1+C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F}_h)\\ &+48CH\log N_{\mathcal{V}}(\epsilon)\beta^E\sqrt{\dim_{\alpha,K}(\mathcal{F})}\sqrt{H\sum_{k=1}^{K}\sum_{h=1}^{H}(\sigma_{k,h}^2+\alpha^2)}\\ &+8H\log N_{\mathcal{V}}(\epsilon)\sqrt{2HK\log(H/\delta)}.\end{aligned}$$

Since $x \leqslant a\sqrt{x}+b$ implies $x \leqslant a^2+2b$, taking $\alpha = 1/\sqrt{H}$, we have that

$$\begin{aligned}\sum_{k=1}^{K}\sum_{h=1}^{H}\sigma_{k,h}^2 \leqslant\ & 2304C^2H^3(\log N_{\mathcal{V}}(\epsilon))^2(\beta^E)^2\dim_{\alpha,K}(\mathcal{F})\\ &+48H^2\log N_{\mathcal{V}}(\epsilon)(1+C\beta^E\gamma^2)\dim_{\alpha,K}(\mathcal{F}_h)\\ &+16H\log N_{\mathcal{V}}(\epsilon)\sqrt{2HK\log(H/\delta)}+K.\end{aligned}$$

□

Proof of Lemma 3.8.16. We prove this statement by induction. Note that $\widetilde{V}^*_{H+1}(s;r_k) = V_{k,H+1}(s) = 0$. Assume that the statement holds for $h+1$. If $V_{k,h}(s) = 1$, then the statement holds trivially for h; otherwise, we have for any $(s,a) \in \mathcal{S}\times\mathcal{A}$ that

$$\begin{aligned}\widehat{Q}_{k,h}(s,a) - \widetilde{Q}^*_h(s,a;r_k) &\geqslant \widehat{f}_{k,h}(s,a) + b_{k,h}(s,a) - [r_{k,h}(s,a;r) + \mathbb{P}_h V^*_{h+1}(s,a;r)]\\ &= [\widehat{f}_{k,h}(s,a) - r_{k,h}(s,a;r) - \mathbb{P}_h V_{k,h+1}(s,a;r)] + b_{k,h}(s,a)\\ &\quad + \mathbb{P}_h V_{k,h+1}(s,a;r) - \mathbb{P}_h V^*_{h+1}(s,a;r)\\ &\geqslant [\widehat{f}_{k,h}(s,a) - r_{k,h}(s,a;r) - \mathbb{P}_h V_{k,h+1}(s,a;r)] + b_{k,h}(s,a)\\ &\geqslant -\beta^E D_{\mathcal{F}_h}(z; z_{[K],h}, \bar{\sigma}_{[K],h}) + 2\beta^E \overline{D}_{\mathcal{F}_h}(z; z_{[K],h}, \bar{\sigma}_{[K],h})\\ &\geqslant 0,\end{aligned}$$

where the first inequality holds due to Definition 3.8.1, the second inequality holds due to induction hypothesis, the third inequality holds due to Lemma 3.8.17, and the forth inequality holds due to Definition 3.3.5. □

3.8.4 Proof of Lemmas in Section 3.8.3

Proof of Lemma 3.8.17. According to the definition of $D^2_{\mathcal{F}}$ function, we have

$$\begin{aligned}
&\left(\widehat{f}_{k,h}(s,a) - \mathcal{T}_h V_{k,h+1}(s,a)\right)^2 \\
&\leqslant D^2_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}) \times \left(\lambda + \sum_{i=1}^{k-1} \frac{1}{(\bar{\sigma}_{i,h})^2} \left(\widehat{f}_{k,h}(s_h^i, a_h^i) - \mathcal{T}_h V_{k,h+1}(s_h^i, a_h^i)\right)^2\right) \\
&\leqslant \left(\beta^E\right)^2 \times D^2_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}),
\end{aligned}$$

where the first inequality holds due the definition of $D^2_{\mathcal{F}}$ function with the Assumption 3.3.2 and the second inequality holds due to the events $\mathcal{E}_h^E$. Thus, we have

$$\left|\widehat{f}_{k,h}(s,a) - \mathcal{T}_h V_{k,h+1}(s,a)\right| \leqslant \beta^E D_{\mathcal{F}_h}(z; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}).$$

□

Proof of Lemma 3.8.18. By Lemma 3.8.21, we have

$$\begin{aligned}
\sum_{k=1}^{K}\sum_{h=1}^{H} \mathbb{P}_h V_{k,h+1}(s_h^k, a_h^k) &= \sum_{k=1}^{K}\sum_{h=1}^{H} V_{k,h+1}(s_{h+1}^k) + \sum_{k=1}^{K}\sum_{h=1}^{H} (\mathbb{P}_h V_{k,h+1}(s_h^k, a_h^k) - V_{k,h+1}(s_{h+1}^k)) \\
&\leqslant \sum_{k=1}^{K}\sum_{h=1}^{H} V_{k,h+1}(s_{h+1}^k) + \sqrt{2KH\log(1/\delta)}.
\end{aligned}$$

Then, under event $\underline{\mathcal{E}}^E$, we have

$$\begin{aligned}
V_{k,h}(s_h^k) &= Q_{k,h}(s_h^k, a_h^k) \\
&= \min\{\widehat{f}_{k,h}(s_h^k, a_h^k) + 2\beta^E \overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}), 1\} \\
&\leqslant \min\{\mathbb{P}_h V_{k,h+1}(s_h^k, a_h^k) + 4\beta^E \overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}), 1\} \\
&= \min\{1, V_{k,h+1}(s_h^k) + (\mathbb{P}_h V_{k,h+1}(s_h^k, a_h^k) - V_{k,h+1}(s_h^k)) \\
&\qquad + 4\beta^E \overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h})\},
\end{aligned}$$

where the inequality holds due to Lemma 3.8.17 and Definition 3.3.5. Therefore, for fixed h, we have

$$\begin{aligned}
\sum_{k=1}^{K} V_{k,h}(s_h^k) &\leqslant \sum_{k=1}^{K} \min\Big\{ \sum_{h'=h}^{H} \big[4\beta^E \overline{\mathcal{D}}_{\mathcal{F}_{h'}}(z_{k,h'}; z_{[k-1],h'}, \bar{\sigma}_{[k-1],h'}) \\
&\qquad + (\mathbb{P}_h V_{k,h'+1}(s_{h'}^k, a_{h'}^k) - V_{k,h'+1}(s_{h'}^k))\big], 1\Big\} \\
&\leqslant \sum_{k=1}^{K} \sum_{h'=h}^{H} \min\Big\{4\beta^E \overline{\mathcal{D}}_{\mathcal{F}_{h'}}(z_{k,h'}; z_{[k-1],h'}, \bar{\sigma}_{[k-1],h'}), 1\Big\} \\
&\qquad + \sum_{k=1}^{K} \sum_{h'=h}^{H} (\mathbb{P}_h V_{k,h'+1}(s_{h'}^k, a_{h'}^k) - V_{k,h'+1}(s_{h'}^k)) \\
&\leqslant \sum_{k=1}^{K} \sum_{h'=h}^{H} \min\Big\{4\beta^E \overline{\mathcal{D}}_{\mathcal{F}_{h'}}(z_{k,h'}; z_{[k-1],h'}, \bar{\sigma}_{[k-1],h'}), 1\Big\} + \sqrt{2HK\log(1/\delta)},
\end{aligned}$$

where the first inequality holds due to induction, and the last inequality holds due to Lemma 3.8.21. Hence, by combining the above two inequalities, we have

$$\begin{aligned}
&\sum_{k=1}^{K} \sum_{h=1}^{H} \mathbb{P}_h V_{k,h+1}(s_h^k, a_h^k) \\
&\leqslant \sum_{k=1}^{K} \sum_{h=1}^{H} V_{k,h+1}(s_{h+1}^k) + \sqrt{2KH\log(1/\delta)} \\
&\leqslant H \sum_{k=1}^{K} \sum_{h=1}^{H} \min\Big\{4\beta^E \overline{\mathcal{D}}_{\mathcal{F}_h}(z_{k,h}; z_{[k-1],h}, \bar{\sigma}_{[k-1],h}), 1\Big\} + (H+1)\sqrt{2HK\log(1/\delta)}.
\end{aligned}$$

□

3.8.5 Auxiliary Lemmas

Lemma 3.8.19 (Self-normalized bound for scalar-valued martingales)**.** Consider random variables $(v_n | n \in \mathbb{N})$ adapted to the filtration $(\mathcal{H}_n : n = 0, 1, ...)$. Let $\{\eta_i\}_{i=1}^{\infty}$ be a sequence of real-valued random variables which is $\mathcal{H}_{i+1}$-measurable and is conditionally σ-sub-Gaussian. Then for an arbitrarily chosen $\lambda > 0$, for any $\delta > 0$, with probability at least $1 - \delta$, it holds

that

$$\sum_{i=1}^{n} \eta_i v_i \leqslant \frac{\lambda\sigma^2}{2} \cdot \sum_{i=1}^{n} v_i^2 + \log(1/\delta)/\lambda \qquad \forall n \in \mathbb{N}.$$

Lemma 3.8.20 (Corollary 2, Agarwal et al. (2022))**.** Let $M > 0, V > v > 0$ be constants, and $\{x_i\}_{i\in[t]}$ be stochastic process adapted to a filtration $\{\mathcal{H}_i\}_{i\in[t]}$. Suppose $\mathbb{E}[x_i|\mathcal{H}_{i-1}] = 0$, $|x_i| \leqslant M$ and $\sum_{i\in[t]} \mathbb{E}[x_i^2|\mathcal{H}_{i-1}] \leqslant V^2$ almost surely. Then for any $\delta, \epsilon > 0$, let $\iota = \sqrt{\log \frac{(2\log(V/v)+2)\cdot(\log(M/m)+2)}{\delta}}$ we have

$$\mathbb{P}\left(\sum_{i\in[t]} x_i > \iota \sqrt{2\left(2\sum_{i\in[t]} \mathbb{E}[x_i^2|\mathcal{H}_{i-1}] + v^2\right)} + \frac{2}{3}\iota^2 \left(2\max_{i\in[t]} |x_i| + m\right)\right) \leqslant \delta.$$

Lemma 3.8.21 (Azuma-Hoeffding Inequality)**.** Let $\{x_i\}_{i=1}^n$ be a martingale difference sequence with respect to a filtration $\{\mathcal{G}_i\}_{i=1}^{n+1}$ such that $|x_i| \leqslant M$ almost surely. That is, x_i is $\mathcal{G}_{i+1}$-measurable and $\mathbb{E}[x_i|\mathcal{G}_i]$ a.s. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$,

$$\sum_{i=1}^{n} x_i \leqslant M\sqrt{2n\log(1/\delta)}.$$

3.9 Experiment details

3.9.1 Details of exploration algorithm

We present the practical algorithm in this subsection. We start by introducing the notation $\boldsymbol{\phi}_i$ as the parameter for the i-th Q networks, which is a three-layer MLP with 1024 hidden size, same as other benchmark algorithms implemented in URLB (Laskin et al., 2021). For the ease of presentation, we ignore the Q network as $Q_{\boldsymbol{\phi}_i}$ as Q_i and the target network $Q_{\bar{\boldsymbol{\phi}}_i}$ as $\bar{Q}_i$ when there is no confusion. We initialize the parameters in $\boldsymbol{\phi}_i$ using Kaiming distribution (He et al., 2015).

The algorithm works in the discounted MDP with the discounted factor γ. For each t in training steps, the algorithm updates the $t\%N$-th Q function by taking the gradient descent

Algorithm 8 GFA-RFE – Exploration Phase – Implementation

Input: Number of ensemble N, update speed η, exploration step T, (reward-free) environment `env`,

Input: Action variance σ^2, minibatch size B, exploration bonus β, discount factor γ

For all $i \in [N]$, initialize $\boldsymbol{\phi}_i$, let $\bar{\boldsymbol{\phi}}_i \leftarrow \boldsymbol{\phi}_i$

Initialize policy network $\pi_{\boldsymbol{\theta}}$, replay buffer $\mathcal{D} = \varnothing$

Observe initial state s_1

for $t = 1, \cdots, T$ **do**

  Sample $\zeta \sim$ Unif.$[0,1]$, sample $a_t \sim \Big\{ N(\pi(\cdot|s_t), \sigma^2)$ **if** $\zeta \leqslant 1-\epsilon$ **else** Unif.$(\mathcal{A}) \Big\}$

  Observe s_{t+1}, let $\mathcal{D} \leftarrow \mathcal{D} \cup (s_t, a_t, s_{t+1})$

  If `env.done`, restart `env` and observe initial state s_{t+1}

  Sample a minibatch $\mathcal{B} = \{(s, a, s')\} \subseteq \mathcal{D}$ with size B

  For each (s, a, s') triplet, calculate $\sigma^2(s,a), r_{\text{int}}(s,a), b(s,a)$ according to (3.9.2).

  Update Q-network $Q_{t\%N}$ by taking one step minimizing $\mathcal{L}(\boldsymbol{\phi}_{t\%N})$ according to (3.9.1)

  Update actor $\pi_{\boldsymbol{\theta}}(\cdot|s)$ by taking one step maximizing $\mathcal{L}(\boldsymbol{\theta})$ according to (3.9.4)

  Update target Q-network following (3.9.3)

end for

regarding the loss function

$$\mathcal{L}(\boldsymbol{\phi}_{t\%N}) = \sum_{(s,a,s') \in \mathcal{B}} \frac{1}{\sigma^2(s,a)} \bigg(Q_{t\%N}(s,a) - \Big(r_{\text{int}}(s,a) + \gamma Q_{\text{target}}(s,a) + b(s,a) \Big) \bigg)^2, \quad (3.9.1)$$

where the target Q function is the average of N target Q network, i.e., $Q_{\text{target}}(s,a) = \sum_{i \in [N]} \bar{Q}_i(s,a)/N$, $\mathcal{B}$ is the minibatch randomly sampled from replay buffer $\mathcal{D}$. We encourage the diversity of different Q function by using different batch $\mathcal{B}$ for updating different Q functions. As the key components of our algorithm, weighted regression $\sigma^2(s,a)$; intrinsic reward $r_{\text{int}}(s,a)$, exploration bonus $b(s,a)$ is calculated based on the variance of the target

Algorithm 9 GFA-RFE – Planning Phase – Implementation (DDPG)

Input: Update speed η, training K, environment `env`, reward function $r(\cdot,\cdot)$

Input: Action variance σ^2, minibatch size B, discount factor γ, offline training data $\mathcal{D}$

Initialize $\boldsymbol{\phi}$, let $\bar{\boldsymbol{\phi}} \leftarrow \boldsymbol{\phi}$

Initialize policy network $\pi_{\boldsymbol{\theta}}$

Update every (s,a,s') in $\mathcal{D}$ to $(s,a,s',r(s,a))$

for $k = 1, \cdots, K$ **do**

  Sample a minibatch $\mathcal{B} = \{(s,a,s',r(s,a))\} \subseteq \mathcal{D}$

  Calculate $\mathcal{L}(\boldsymbol{\phi}) = \sum_{(s,a,s')\in\mathcal{B}} \left(Q_{\boldsymbol{\phi}}(s,a) - \left(r(s,a) + \gamma Q_{\text{target}}(s', \pi_{\boldsymbol{\theta}}(s')) \right) \right)^2$

  Update Q-network $Q_{t\%N}$ by taking one step minimizing $\mathcal{L}(\boldsymbol{\phi})$

  Calculate actor loss $\mathcal{L}(\boldsymbol{\theta}) = \sum_{(s,a,s')\in\mathcal{B}} Q_{\boldsymbol{\phi}}(s, \pi_{\boldsymbol{\theta}}(a|s))$

  Update actor $\pi_{\boldsymbol{\theta}}(\cdot|s)$ by taking one step maximizing $\mathcal{L}(\boldsymbol{\theta})$

  Update target Q-network by $\bar{\boldsymbol{\phi}} \leftarrow (1-\eta)\bar{\boldsymbol{\phi}} + \eta\boldsymbol{\phi}$

end for

Q network across $\bar{Q}_i$ instances:

$$\sigma^2(s,a) = \text{Var}[\bar{Q}_i(s,a)];\ r_{\text{int}}(s,a) = (1-\gamma)\sqrt{\text{Var}[\bar{Q}_i(s,a)]};\ b(s,a) = \beta\sqrt{\text{Var}[\bar{Q}_i(s,a)]}, \tag{3.9.2}$$

where we simply set $\beta = 1$ to align with our theory, the factor $(1-\gamma)$ before the intrinsic reward is because we want to balance the horizon $1/H \approx (1-\gamma)$ in the setting. The reason for choosing the target Q function $\bar{Q}_i$ instead of the updating Q function is to update the intrinsic reward, exploration bonus slower than the update of Q function, therefore give the agent more time to explore the optimal policy for maximizing a certain intrinsic reward $r_{\text{int}}(s,a)$. After updating the parameter $\boldsymbol{\phi}_{t\%N}$, we perform a soft update for the target network as

$$\bar{\boldsymbol{\phi}}_{t\%N} \leftarrow (1-\eta)\bar{\boldsymbol{\phi}}_{t\%N} + \eta\boldsymbol{\phi}_{t\%N}, \tag{3.9.3}$$

where we follow the setting in URLB to set $\eta = 0.01$. After updating the Q function, the

algorithm then updates the actor $\pi_{\boldsymbol{\theta}}(a|s)$ following DDPG in maximizing

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{(s,a,s')\in\mathcal{B}} \sum_{i\in[N]} Q_i(s, \pi_{\boldsymbol{\theta}}(a|s)) \tag{3.9.4}$$

We summarize the exploration algorithm in Algorithm 8, in particular, we use Adam to optimize the loss function defined by (3.9.1) and (3.9.3).

3.9.2 Details of offline training algorithm

After collecting the dataset $\mathcal{D}$, we call a reward oracle to label the reward r for any triplet $(s, a, s') \in \mathcal{D}$. Then the DDPG algorithm is called to learn the optimal policy. For the fair comparison with other benchmark algorithm, we do not add weighted regression in the planning phase, thus the algorithm stays the same with the one presented in URLB, as stated in Algorithm 9

3.9.3 Hyper-parameters

We present a common set of hyper-parameters used in our experiments in Table 3.2. And we list individual hyper-parameters for each method in table 3.3. All common hyper-parameters and individual hyper-parameters for baseline algorithms are the same as what is used in Laskin et al. (2021) and its implementations.

3.9.4 Ablation Study

3.9.4.1 Learning Processes

Figure 3.2 illustrate the episode rewards for each algorithm across training steps for various tasks, demonstrating that the performance of our algorithm (Algorithm 6) ranks among the top tier in all tasks.

Table 3.2: The common set of hyper-parameters.

Hyper-parameter	**Value**
Replay buffer capacity	10^6
Action repeat	1
n-step returns	3
Mini-batch size	1024
Discount (γ)	0.99
Optimizer	Adam
Learning rate	10^{-4}
Agent update frequency	2
Critic target EMA rate (τ_Q)	0.01
Features dim.	50
Hidden dim.	1024
Exploration stddev clip	0.3
Exploration stddev value	0.2
# frames per episode	1×10^3
# online exploration frames	up to 1×10^6
# offline planning frames	1×10^5
Critic network	$(\|O\| + \|A\|) \rightarrow 1024 \rightarrow$ LN $\rightarrow$ Tanh $\rightarrow 1024 \rightarrow$ RELU $\rightarrow 1$
Actor network	$\|O\| \rightarrow 50 \rightarrow$ LN $\rightarrow$ Tanh $\rightarrow 1024 \rightarrow$ RELU $\rightarrow$ action dim

3.9.4.2 Numbers of Exploration Episodes

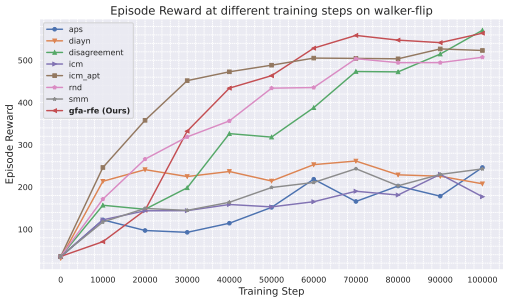
Figure 3.3 show the episode rewards for top-performing algorithms, including our algorithm (GFA-RFE), RND, Disagreement, and APT, across varying numbers of exploration episodes for different tasks. Notably, GFA-RFE competes with these leading unsupervised algorithms effectively, matching their performance across a range of exploration episodes.

Table 3.3: Hyper-parameters of for GFA-RFE and baseline (ICM, Disagreement, RND).

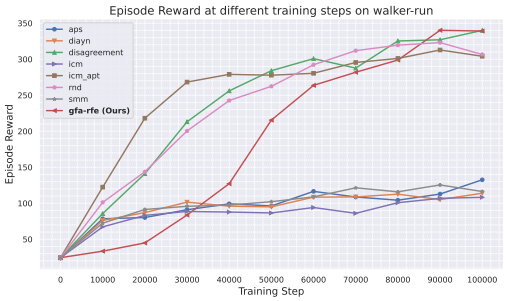
GFA-RFE	Value
Ensemble size	10
Exploration bonus	2
Exploration ϵ	0.2
ICM hyper-parameter	**Value**
Reward transformation	$\log(r + 1.0)$
Forward net arch.	$(\|O\| + \|A\|) \rightarrow 1024 \rightarrow 1024 \rightarrow \|O\|$ ReLU MLP
Inverse net arch.	$(2 \times \|O\|) \rightarrow 1024 \rightarrow \|A\|$ ReLU MLP
Disagreement hyper-parameter	**Value**
Ensemble size	5
Forward net arch:	$(\|O\| + \|A\|) \rightarrow 1024 \rightarrow 1024 \rightarrow \|O\|$ ReLU MLP
RND hyper-parameter	**Value**
Representation dim.	512
Predictor & target net arch.	$\|O\| \rightarrow 1024 \rightarrow 1024 \rightarrow 512$ ReLU MLP
Normalized observation clipping	5

Table 3.4: Hyper-parameters of for baseline algorithms (APT, SMM, DIAYN, APS).

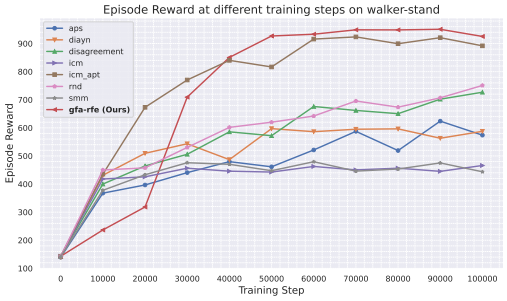
APT hyper-parameter	Value
Representation dim.	512
Reward transformation	$\log(r + 1.0)$
Forward net arch.	$(512 + \|A\|) \rightarrow 1024 \rightarrow 512$ ReLU MLP
Inverse net arch.	$(2 \times 512) \rightarrow 1024 \rightarrow \|A\|$ ReLU MLP
k in NN	12
Avg top k in NN	True
SMM hyper-parameter	**Value**
Skill dim.	4
Skill discrim lr	10^{-3}
VAE lr	10^{-2}
DIAYN hyper-parameter	**Value**
Skill dim	16
Skill sampling frequency (steps)	50
Discriminator net arch.	$512 \rightarrow 1024 \rightarrow 1024 \rightarrow 16$ ReLU MLP
APS hyper-parameter	**Value**
Reward transformation	$\log(r + 1.0)$
Successor feature dim.	10
Successor feature net arch.	$\|O\| \rightarrow 1024 \rightarrow 10$ ReLU MLP
k in NN	12
Avg top k in NN	True
Least square batch size	4096



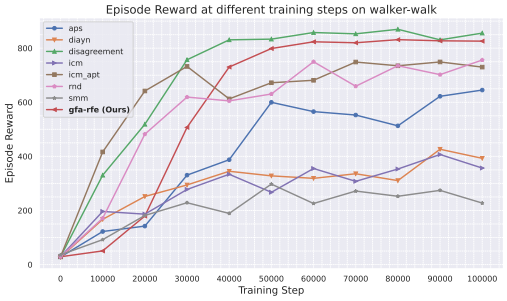
(a) Walker-Flip



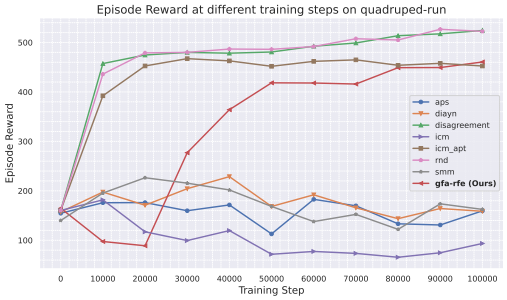
(b) Walker-Run



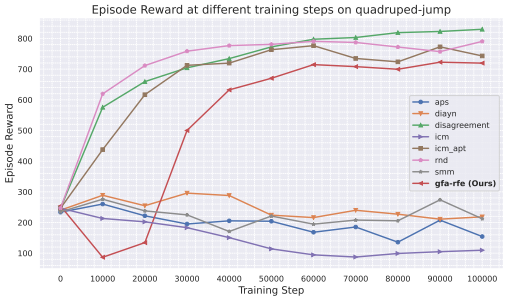
(c) Walker-Stand



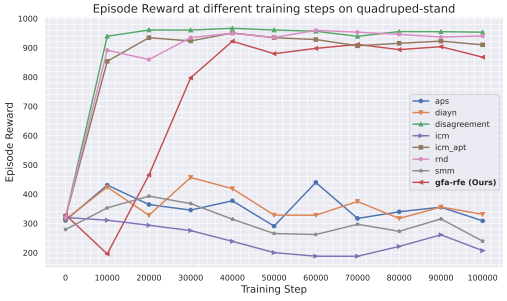
(d) Walker-Walk



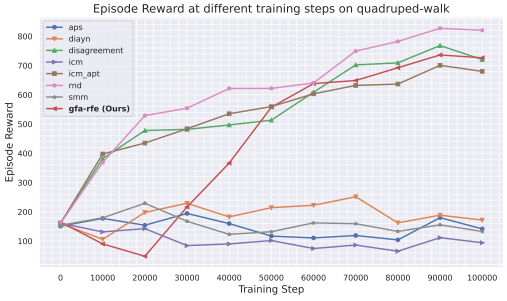
(e) Quadruped-Run



(f) Quadruped-Jump

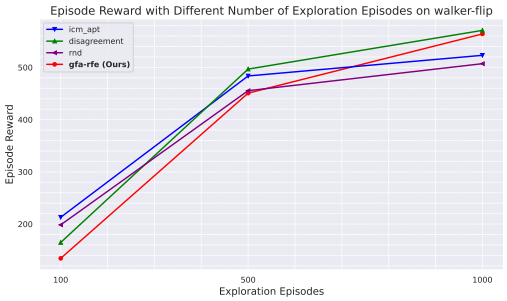


(g) Quadruped-Stand

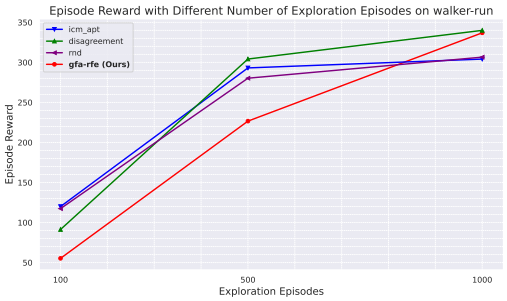


(h) Quadruped-Walk

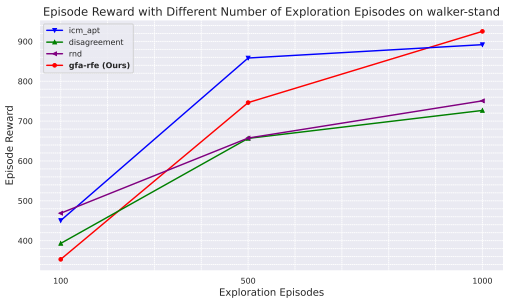
Figure 3.2: Episode reward at different training steps for tasks on *walker* and *quadruped*.



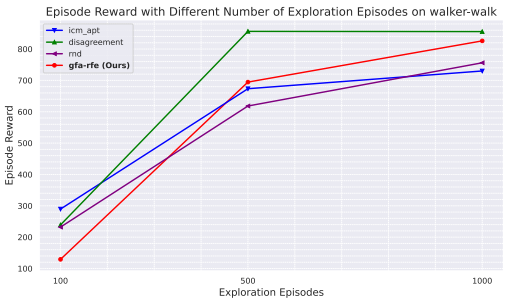
(a) Walker-Flip



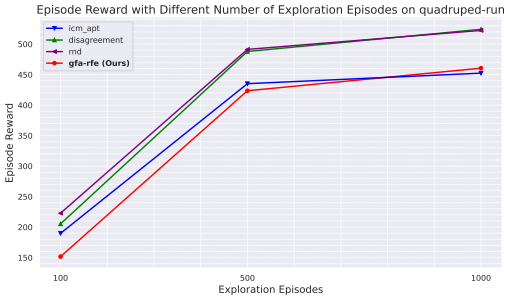
(b) Walker-Run



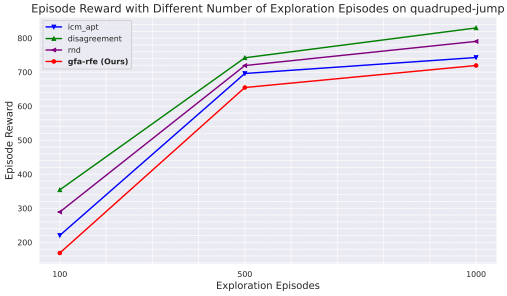
(c) Walker-Stand



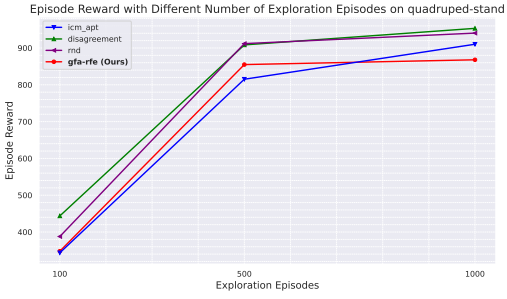
(d) Walker-Walk



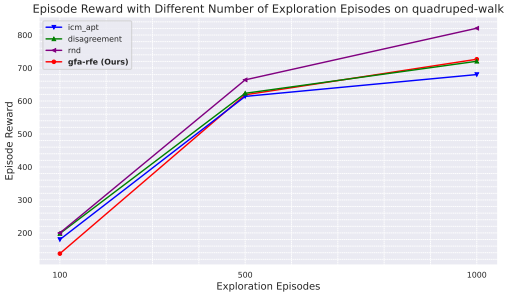
(e) Quadruped-Run



(f) Quadruped-Jump



(g) Quadruped-Stand



(h) Quadruped-Walk

Figure 3.3: Episode reward with different exploration episodes on *walker* and *quadruped*.

CHAPTER 4

Uncertainty-Aware Robust Linear Contextual Bandits

4.1 Introduction

From this chapter, we move on to the second topic: how to design robust decision making systems by leveraging the uncertainty quantification. In this chapter, we start from (linear) contextual bandits. A *contextual bandit* is a task in which, in each round, the agent observes a set of *contextual vectors* describing the features of different actions. The agent needs to select the action that has the maximum reward, where the reward can be viewed as a function of the contextual vectors. For example, in a recommender system as demonstrated in Figure 4.1[1]. For each round, the agent observes different possible choices of food. These choices are described in terms of their category, calories, or style as the contextual vector (feature) of the foods. The goal for the agent is to select a type of food that the user is most likely to eat and then recommend it to the user. The reward is 1 when the user picks the recommendation and 0 otherwise and can be viewed as a function of the contextual vectors with some noise. It is obvious that the contextual bandit task can be viewed as the most simplified reinforcement learning tasks with only one decision step instead of making *sequential* decision that may affect each other.

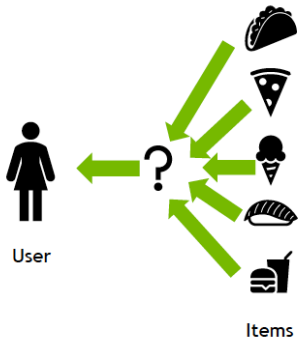


Figure 4.1: An illustration of the recommender system.

[1]Image credit: `https://www.nvidia.com/en-us/glossary/recommendation-system`

Linear contextual bandits (Li et al., 2010; Chu et al., 2011; Abbasi-Yadkori et al., 2011; Agrawal and Goyal, 2013) have been extensively studied when the reward function can be represented as a linear function of the contextual vectors. However, such a well-specified linear model assumption sometimes does not hold in practice. This motivates the study of misspecified linear models. In particular, we only assume that the reward function can be approximated by a linear function up to some worst-case error ζ called the *misspecification level*. Existing algorithms for misspecified linear contextual bandits (Lattimore et al., 2020; Foster et al., 2020) can only achieve an $\widetilde{\mathcal{O}}(d\sqrt{K} + \zeta K\sqrt{d}\log K)$ regret bound, where K is the total number of rounds and d is the dimension of the contextual vector. Such a regret, however, suggests that the performance of these algorithms will degenerate to be linear in K when K is sufficiently large. The reason for this performance degeneration is because existing algorithms, such as OFUL (Abbasi-Yadkori et al., 2011) and linear Thompson sampling (Agrawal and Goyal, 2013), utilize all the collected data without selection. This makes these algorithms vulnerable to "outliers" caused by the misspecified model. Meanwhile, the aforementioned results do not consider the sub-optimality gap in the expected reward between the best arm and the second best arm. Intuitively speaking, if the sub-optimality gap is smaller than the misspecification level, there is no hope to obtain a sublinear regret. Therefore, it is sensible to take into account the sub-optimality gap in the misspecified setting, and pursue a gap-dependent regret bound.

The same misspecification issue also appears in reinforcement learning with linear function approximation, when a linear function cannot exactly represent the transition kernel or value function of the underlying MDP. In this case, Du et al. (2019) provided a negative result showing that if the misspecification level is larger than a certain threshold, any RL algorithm will suffer from an exponentially large sample complexity. This result was later revisited in the stochastic linear bandit setting by Lattimore et al. (2020), which shows that a large misspecification error will make the bandit model not efficiently learnable. However, these results cannot explain the tremendous success of deep reinforcement learning on vari-

ous tasks (Mnih et al., 2013; Schulman et al., 2015, 2017), where the deep neural networks are used as function approximators with misspecification error.

4.1.1 Organization of this Chapter

In this chapter, we aim to understand the role of model misspecification in linear contextual bandits through the lens of suboptimality gap. This chapter is organized as follows. We present the related works in Section 4.2 and the preliminaries in Section 4.3. In Section 4.4, we propose and analyze a new algorithm with data selection, which can handle misspecified bandits with the knowledge of sub-optimality gap Δ. In Section 4.5, we move on to eliminating the dependence of the knowledge of Δ and show that the existing algorithm, SupLinUCB (Chu et al., 2011) can be also viewed as a bootstrapped version of our proposed algorithm. Empirical results are presented in Section 4.7 and the conclusion is drawn in Section 4.8. We defer the detailed proof for several key lemmas to Section 4.9.

4.2 Related Works

In this section, we review the related work for misspecified linear bandits and misspecified reinforcement learning.

4.2.1 Linear Contextual Bandits

There is a large body of literature on linear contextual bandits. For example, Auer (2002); Chu et al. (2011); Agrawal and Goyal (2013) studied linear contextual bandits when the number of arms is finite. Abbasi-Yadkori et al. (2011) proposed an algorithm called OFUL to deal with the infinite arm set. All these works come with an $\widetilde{\mathcal{O}}(\sqrt{K})$ problem-independent regret bound, and an $\mathcal{O}(d^2\Delta^{-1}\log(K))$ gap-dependent regret bound is also given by Abbasi-Yadkori et al. (2011).

4.2.2 Misspecified Linear Bandits.

There is a long history of the robust contextual bandits in the face of misspecification. Agarwal et al. (2014) considered using an oracle to learn the contextual bandits with function approximation and showed that the proposed algorithm is robust when misspecification exists. Ghosh et al. (2017) considered the misspecified linear bandits and showed that the OFUL (Abbasi-Yadkori et al., 2011) algorithm cannot achieve a sublinear regret in the presence of misspecification. They, therefore, proposed a new algorithm with a hypothesis testing module for linearity to determine whether to use OFUL (Abbasi-Yadkori et al., 2011) or the multi-armed UCB algorithm. Their algorithm enjoys the same performance guarantee as OFUL in the well-specified setting and can avoid the linear regret under certain misspecification setting. Lattimore et al. (2020) proposed a phase-elimination algorithm for misspecified stochastic linear bandits, which achieves an $\widetilde{\mathcal{O}}(\sqrt{dK}+\zeta K\sqrt{d})$ regret bound. For contextual linear bandits, both Lattimore et al. (2020) and Foster et al. (2020) proved an $\widetilde{\mathcal{O}}(d\sqrt{K}+\zeta K\sqrt{d})$ regret bound under misspecification. Takemura et al. (2021) showed that SupLinUCB can achieve a similar regret bound without the knowledge of the misspecification level. Van Roy and Dong (2019) proved a lower bound of sample complexity, which suggests when $\zeta\sqrt{d} \geqslant \sqrt{8\log|\mathcal{D}|}$, any best arm identification algorithm will suffer a $\Omega(2^d)$ sample complexity, where $\mathcal{D}$ is the decision set. When the reward is deterministic and does not contain noise, they provided an algorithm using $\widetilde{\mathcal{O}}(d)$ sample complexity to identify a Δ-optimal arm when $\zeta \leqslant \Delta/\sqrt{d}$. Lattimore et al. (2020) also mentioned that if $\zeta\sqrt{d} \leqslant \Delta$, there exists a best arm identification algorithm that only needs to pull $\widetilde{\mathcal{O}}(d)$ arms to find a Δ-optimal arm with the knowledge of ζ. Note that although the exponential sample complexity lower bound for best-arm identification can be translated into a regret lower bound in linear contextual bandits, the algorithms for best-arm identification and the corresponding upper bounds cannot be easily extended to linear contextual bandits. Besides these works on misspecification, He et al. (2022b) studied the linear contextual bandits with adversarial corruptions, where the reward for each round can be corrupted arbitrarily. They assumed

Algorithm	Misspecified MDP?	Result
LSVI-UCB (He et al., 2021a)	×	$\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1}\log(K))$
LSVI-UCB (Papini et al., 2021a)	×	$\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1}\log(1/\lambda))$
Cert-LSVI-UCB (ours, Theorem 4.4.1)	✓	$\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1})$

Table 4.1: Instance-dependent regret bounds for different algorithms under the linear MDP setting. Here d is the dimension of the linear function $\boldsymbol{\phi}(s,a)$, H is the horizon length, Δ is the minimal suboptimality gap. All results in the table represent high probability regret bounds. The regret bound depends the number of episodes K in He et al. (2021a) and the minimum positive eigenvalue λ of features mapping in Papini et al. (2021b). **Misspecified MDP?** indicates if the algorithm can (✓) handle the misspecified linear MDP or not (×).

that the summation of the corruption up to K rounds is bounded by $C > 0$ and proposed an algorithm achieving $\widetilde{\mathcal{O}}(d\sqrt{K} + dC)$ regret bound with the known C. Since the corruption level $C = K\zeta$ in the misspecification setting, their result directly implied an $\mathcal{O}(d\sqrt{K} + dK\zeta)$ linear regret, which differs from the optimal guarantee with a extra $O(\sqrt{d})$ factor. Besides these series of work, Camilleri et al. (2021) also studied the robustness of kernel bandits with misspecification.

4.3 Preliminaries

We consider a linear contextual bandit problem. In round $k \in [K]$, the agent receives a decision set $\mathcal{D}_k \subset \mathbb{R}^d$ and selects an arm $\mathbf{x}_k \in \mathcal{D}_k$ then observes the reward $r_k = r(\mathbf{x}_k) + \varepsilon_k$, where $r(\cdot) : \mathbb{R}^d \mapsto [0,1]$ is a deterministic expected reward function and ε_k is a zero-mean R-sub-Gaussian random noise. i.e., $\mathbb{E}[e^{\lambda\varepsilon_k}|\mathbf{x}_{1:k}, \varepsilon_{1:k-1}] \leqslant \exp(\lambda^2R^2/2), \forall k \in [K], \lambda \in \mathbb{R}$.

In this work, we assume that all contextual vector $\mathbf{x} \in \mathcal{D}_k$ satisfies $\|\mathbf{x}\|_2 \leqslant L$ and the reward function $r(\cdot) : \mathbb{R}^d \to [0,1]$ can be approximated by a linear function $r(\mathbf{x}) = \mathbf{x}^\top\boldsymbol{\theta}^* +$

$\eta(\mathbf{x})$, where $\eta(\cdot) : \mathbb{R}^d \mapsto [-\zeta, \zeta]$ is an unknown misspecification error function. We further assume $\|\boldsymbol{\theta}^*\|_2 \leqslant B$ and for simplicity, we assume $B, L \geqslant 1$. We denote the optimal reward at round k as $r_k^* = \max_{\mathbf{x}\in\mathcal{D}_k} r(\mathbf{x})$ and the optimal arm $\mathbf{x}_k^* = \text{argmax}_{\mathbf{x}\in\mathcal{D}_k} r(\mathbf{x})$. Our goal is to minimize the regret defined by $\text{Regret}(K) := \sum_{k=1}^K r_k^* - r(\mathbf{x}_k)$.

We focus on the minimal sub-optimality gap condition.

Definition 4.3.1 (Minimal sub-optimality gap)**.** For each $\mathbf{x} \in \mathcal{D}_k$, the sub-optimality gap $\Delta_k(\mathbf{x})$ is defined by $\Delta_k(\mathbf{x}) := r_k^* - r(\mathbf{x})$ and the minimal sub-optimality gap Δ is defined by $\Delta := \min_{k\in[K],\mathbf{x}\in\mathcal{D}_k}\{\Delta_k(\mathbf{x}) : \Delta_k(\mathbf{x}) > 0\}$.

Then we further assume this minimal sub-optimality gap is strictly positive, i.e., $\Delta > 0$.

4.4 Constant Regret Bound with Known Sub-Optimality Gap

4.4.1 Proposed Algorithm

In this subsection, we propose our algorithm, DS-OFUL, in Algorithm 10. The algorithm runs for K rounds. At each round, the algorithm first estimates the underlying parameter $\boldsymbol{\theta}^*$ by solving the following ridge regression problem in Line 4

$$\boldsymbol{\theta}_k = \text{argmin}_{\boldsymbol{\theta}} \sum_{i\in\mathcal{C}_{k-1}} \left(r_i - \mathbf{x}_i^\top \boldsymbol{\theta}\right)^2 + \lambda\|\boldsymbol{\theta}\|_2^2,$$

where $\mathcal{C}_{k-1}$ is the index set of the selected contextual vectors for regression and is initialized as an empty set at the beginning. After receiving the contextual vectors set $\mathcal{D}_k$, the algorithm selects an arm from the optimistic estimation powered by the Upper Confidence Bound (UCB) bonus in Line 6. In line 8, the algorithm adds the index of current round into $\mathcal{C}_k$ if the UCB bonus of the chosen arm $\mathbf{x}_k$, denoted by $\|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}}$, is greater than the threshold Γ. Intuitively speaking, since the UCB bonus reflects the uncertainty of the model about the given arm $\mathbf{x}$, Line 8 discards the data that brings little uncertainty ($\|\mathbf{x}\|_{\mathbf{U}_k^{-1}}$) to the model. Finally, we denote the total number of selected data in Line 8 by $|\mathcal{C}_K|$. We will declare the choices of the parameter Γ, β and λ in the next section.

Algorithm 10 Data Selection OFUL (DS-OFUL)

Input: Threshold Γ, radius β and regularizer λ

Initialize $\mathcal{C}_0 = \varnothing, \mathbf{U}_0 = \lambda\mathbf{I}, \boldsymbol{\theta}_0 = \mathbf{0}$

for $k = 1, \ldots, K$ **do**

  Set $\mathbf{U}_k = \lambda\mathbf{I} + \sum_{i\in\mathcal{C}_{k-1}} \mathbf{x}_i\mathbf{x}_i^\top$.

  Set $\boldsymbol{\theta}_k = \mathbf{U}_k^{-1}\sum_{i\in\mathcal{C}_{k-1}} r_i\mathbf{x}_i$.

  Receive the decision set $\mathcal{D}_k$.

  Select $\mathbf{x}_k = \operatorname{argmax}_{\mathbf{x}\in\mathcal{D}_k}\left\{\mathbf{x}^\top\boldsymbol{\theta}_k + \beta\|\mathbf{x}\|_{\mathbf{U}_k^{-1}}\right\}$.

  Receive reward r_k

  if $\|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}} \geqslant \Gamma$ **then** $\mathcal{C}_k = \mathcal{C}_{k-1}\cup\{k\}$ **else** $\mathcal{C}_k = \mathcal{C}_{k-1}$

end for

4.4.2 Regret Bound

In this subsection, we provide the regret upper bound of Algorithm 10 and the regret lower bound for learning the misspecified linear contextual bandit.

Theorem 4.4.1 (Upper Bound)**.** For any $0 < \delta < 1$, let $\lambda = B^{-2}$ and $\Gamma = \Delta/(2\sqrt{d}\iota_1)$ where $\iota_1 = (24+18R)\log((72+54R)LB\sqrt{d}\Delta^{-1}) + \sqrt{8R^2\log(1/\delta)}$. Set $\beta = 1 + 4\sqrt{d\iota_2} + R\sqrt{2d\iota_3}$ where $\iota_2 = \log(3LB\Gamma^{-1})$, $\iota_3 = \log((1+16L^2B^2\Gamma^{-2}\iota_2)/\delta)$. If the misspecification level is bounded by $2\sqrt{d}\zeta\iota_1 \leqslant \Delta$, then with probability at least $1-\delta$, the cumulative regret of Algorithm 10 is bounded by

$$\text{Regret}(K) \leqslant \frac{32\beta\sqrt{2d^3\iota_2\log(1+16d\Gamma^{-2}\iota_2)}\iota_1}{\Delta}.$$

Remark 4.4.2. Since $\beta = \widetilde{\mathcal{O}}(\sqrt{d})$, Theorem 4.4.1 suggests an $\widetilde{\mathcal{O}}(d^2\Delta^{-1})$ constant regret bound independent of the total number of rounds K when $\zeta \leqslant \widetilde{\mathcal{O}}(\Delta/\sqrt{d})$, which improves the logarithmic regret $\widetilde{\mathcal{O}}(d^2\Delta^{-1}\log K)$ in Abbasi-Yadkori et al. (2011) to a constant regret[2]. Note that our constant regret bound relies on the knowledge of the minimal sub-optimality

[2]When we say constant regret, we ignore the $\log(1/\delta)$ factor in the regret as we choose δ to be a constant.

gap Δ, while the OFUL algorithm in Abbasi-Yadkori et al. (2011) does not need prior knowledge about the minimal sub-optimality gap Δ.

Remark 4.4.3. Our *high probability* constant regret bound does not violate the lower bound proved in Hao et al. (2020), which says that certain diversity condition on the contexts is necessary to achieve an *expected* constant regret bound (Papini et al., 2021b). Here we only provide a high-probability constant regret bound. When extending this high probability constant regret bound to expected regret bound, we have

$$\mathbb{E}[\text{Regret}(K)] \leqslant \widetilde{\mathcal{O}}(d^2\Delta^{-1}\log(1/\delta))(1-\delta) + \delta K,$$

which depends on K. To obtain a sub-linear expected regret, we can choose $\delta = 1/K$, which yields a logarithmic regret $\widetilde{\mathcal{O}}(d^2\Delta^{-1}\log(K))$ and does not violate the lower bound in Hao et al. (2020).

Remark 4.4.4. Notably, Papini et al. (2021b) can achieve a constant expected regret bound under certain diversity condition, which requires the contexts of arms span the whole $\mathbb{R}^d$ space. In contrast, our constant regret bound does not need such an assumption and is a high-probability constant regret bound.

4.4.3 Key Proof Techniques

Here we present the key proof techniques for achieving the constant regret with the knowledge of sub-optimality gap Δ. The detailed proof is deferred to Section 4.9.1.

4.4.3.1 Regret decomposition

The total regret over all K rounds can be decomposed as follows

$$\text{Regret}(K) = \sum_{k\in\mathcal{C}_K} \big(r_k^* - r(\mathbf{x}_k)\big) + \sum_{k\notin\mathcal{C}_K} \big(r_k^* - r(\mathbf{x}_k)\big). \tag{4.4.1}$$

4.4.3.2 Finite samples collected in $\mathcal{C}_k$

Since we only adding the contextual arm with large uncertainty (i.e., $\|\mathbf{x}\|_{\mathbf{U}_k^{-1}} \geqslant \Gamma$) into the set $\mathcal{C}_k$, we can bound the number of samples in $\mathcal{C}_k$ as $\mathcal{C}_k = \widetilde{\mathcal{O}}(d\Gamma^{-2})$ which is claimed in the following lemma.

Lemma 4.4.5. Given $0 < \Gamma \leqslant 1$, set $\lambda = B^{-2}$. For any $k \in [K]$, $|\mathcal{C}_k| \leqslant 16d\Gamma^{-2}\log(3LB\Gamma^{-1})$.

Then the following lemma suggests that a finite regression set $\mathcal{C}_k$ can lead to a small confidence set with misspecification.

Lemma 4.4.6. Let $\lambda = B^{-2}$. For all $\delta > 0$, with probability at least $1-\delta$, for all $\mathbf{x} \in \mathbb{R}^d, k \in [K]$, the prediction error is bounded by:

$$|\mathbf{x}^\top(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)| \leqslant \left(1 + R\sqrt{2d\iota} + \zeta\sqrt{|\mathcal{C}_k|}\right)\|\mathbf{x}\|_{\mathbf{U}_k^{-1}},$$

where $\iota = \log((d+|\mathcal{C}_k|L^2B^2)/(d\delta))$ and $|\mathcal{C}_k|$ is the total number of data used in regression at the k-th round.

Comparing the confidence radius $\widetilde{\mathcal{O}}(R\sqrt{d} + \zeta\sqrt{|\mathcal{C}_k|})$ here with the conventional radius $\widetilde{\mathcal{O}}(R\sqrt{d})$ in OFUL, one can find that the misspecification error will affect the radius by an $\sqrt{|\mathcal{C}_K|}$ factor. If we use all the data to do regression, the confidence radius will be in the order of $\widetilde{\mathcal{O}}(\sqrt{K})$ and therefore will lead to a $\mathcal{O}(K\sqrt{\log K})$ regret bound (see Lemma 11 in Abbasi-Yadkori et al. (2011)). This makes the regret bound vacuous. In contrast, in our algorithm, the confidence radius is only $\sqrt{|\mathcal{C}_K|}$ where $|\mathcal{C}_K|$ is finite given Lemma 4.4.5. As a result, our regret bound will not grow with K as in OFUL and will be smaller.

4.4.3.3 Skipped rounds are optimal

Given the fact that the selected arm set $\mathcal{C}_k$ is finite, the rest of the proof is simply showing that the skipped rounds $k \notin \mathcal{C}_k$ are optimal and will not incur regret. Since we have $\|\mathbf{x}\|_{\mathbf{U}_k^{-1}} \leqslant \Gamma$ for those skipped rounds, the sub-optimality is bounded by the following (informal) lemma.

Lemma 4.4.7. The instantaneous regret for round $k \notin \mathcal{C}_k$ is bounded by

$$\Delta_k(\mathbf{x}_k) \leqslant 2\zeta + 2\beta\|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}} \leqslant \widetilde{\Theta}(\zeta + \Delta + \sqrt{d}\Gamma),$$

Setting $\Gamma = \widetilde{\Theta}(\Delta/\sqrt{d})$ suggests that the instantaneous regret $\Delta_k(\mathbf{x}_k) \leqslant \Delta$, which means no instantaneous regret occurs on round k.

4.4.3.4 Achieving the constant regret

To wrap up, as (4.4.1) suggests, for rounds $k \in \mathcal{C}_K$, we can follow the gap-dependent regret analysis in Abbasi-Yadkori et al. (2011) and obtain an $\widetilde{\mathcal{O}}(d^2 \log(|\mathcal{C}_K|)/\Delta)$ gap-dependent regret bound, which is independent of K according to Lemma 4.4.5. For rounds $k \notin \mathcal{C}_K$, Lemma 4.4.7 guarantees a zero instantaneous regret. Putting them together yields the claimed constant regret bound.

4.5 Constant Regret Bound with Unknown Sub-Optimality Gap

4.5.1 Proposed Algorithm

Although Algorithm 10 can achieve a constant regret, it requires the knowledge of sub-optimality gap Δ. To tackle this problem, we propose a new algorithm that does not require the knowledge of sub-optimality gap Δ.

The algorithm is described in Algorithm 11. It inherits the arm elimination method from SupLinUCB (Chu et al., 2011). A similar algorithm is also presented for misspecified linear bandits in Takemura et al. (2021).

Algorithm 11 works as follows. At each round $k \in [K]$, the algorithm maintains l levels of ridge regression with different set $\mathcal{C}_{k-1}^l$, where the estimation error for the l-th level is about $\beta(l)2^{-l}$ (we will prove this in the latter analysis). Then starting from the first level $l = 1$ and the received decision set $\mathcal{D}_k$, if there exists an arm in the decision set with a

large uncertainty (i.e., $\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}} \geqslant 2^{-l}$), the algorithm directly selects that arm (Line 8). According to Lemma 4.4.5 in the analysis of DS-OFUL, the number of selected contexts at each level should be bounded. If the uncertainty for all arms is smaller than the threshold 2^{-l}, the algorithm follows the arm elimination rule, which reduces the decision set into

$$\mathcal{D}_k^{l+1} = \left\{\mathbf{x} : \mathbf{x} \in \mathcal{D}_k^l, r_k^l(\mathbf{x}_k^l) - r_k^l(\mathbf{x}) \leqslant 3\beta(l)2^{-l}\right\}. \tag{4.5.1}$$

Then the algorithm enters the next level $l+1$ until it reaches $\log(k)$-th level as Line 10 suggests. For the level $l \geqslant \log(k)$, the algorithm directly selects the arm with highest optimistic reward on Line 11 and does not add the index k to the regression set $\mathcal{C}_k^l$ as on Line 12 since the uncertainty is small enough.

Algorithm 11 can be viewed as the multi-level version of Algorithm 10 boosted by the peeling technique. Algorithm 11 does not require the knowledge of the sub-optimality gap Δ: if Δ is known, one can directly jump to a specific level $l_\Delta = \widetilde{\mathcal{O}}(\log(d/\Delta))$, where the prediction error is bounded by $2\beta(l_\Delta)2^{-l_\Delta} = \widetilde{\mathcal{O}}(\Delta)$ and is sufficient to achieve zero-instantaneous regret. However, when the Δ is unknown, Algorithm 11 has to do a grid search over $2^{-1}, 2^{-2}, \cdots 2^{-l_\Delta}, \cdots$ and waste some of the samples to learn the first $l_\Delta - 1$ levels. We will revisit and compare the difference between these two algorithms in the later regret analysis.

4.5.2 Regret Bound

This subsection provides the regret upper bound for Algorithm 11.

Theorem 4.5.1 (Upper Bound)**.** For any $0 < \delta < 1$, let $\lambda = B^{-2}$. For every integer $l > 0$, set $\beta(l) = 1 + R\sqrt{2d\iota_2(l)}$ where $\iota_2(l) = \log((d2^l + 16L^2B^2 8^l \iota_1(l))/(d\delta))$ and $\iota_1(l) = \log\left(3LB2^l\right)$. If the misspecification level is bounded by $4l_\Delta\zeta\left(1 + 4\sqrt{d\iota_1(l_\Delta)}\right) < \Delta$ where l_Δ is the minimal solution to $l_\Delta > \log(8\beta(l_\Delta)/\Delta)$, then with probability at least $1-\delta$, the cumulative regret

Algorithm 11 SupLinUCB

Input: Regularization λ, confidence radius $\beta(\cdot)$

Initialize $\mathcal{C}_0^l = \varnothing$ for all $l \in [\lceil \log(K) \rceil]$

for $k = 1, 2, \cdots K$ **do**

  Set $\mathcal{D}_k^1 = \mathcal{D}_k$ and $l = 1$

  repeat

    Set $\mathbf{U}_k^l = \lambda \mathbf{I} + \sum_{i \in \mathcal{C}_{k-1}^l} \mathbf{x}_i \mathbf{x}_i^\top$, $\boldsymbol{\theta}_k^l = (\mathbf{U}_k^l)^{-1} \sum_{i \in \mathcal{C}_{k-1}^l} r_i \mathbf{x}_i$

    Set $r_k^l(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}_k^l + \beta(l) \|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}}$, action $\mathbf{x}_k^l = \operatorname{argmax}_{\mathbf{x} \in \mathcal{D}_k^l} r_k^l(\mathbf{x})$

    if $\max_{\mathbf{x} \in \mathcal{D}_k^l} \|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}} \geqslant 2^{-l}$ **then**

      Choose $\mathbf{x}_k = \operatorname{argmax}_{\mathbf{x} \in \mathcal{D}_k^l} \|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}}$

      Update $\mathcal{C}_k^l = \mathcal{C}_{k-1}^l \cup \{k\}$ and keep $\mathcal{C}_k^{l'} = \mathcal{C}_{k-1}^{l'}$ for all $l' \neq l$

    else if $k \leqslant 4^l d$ **then**

      Choose $\mathbf{x}_k = \mathbf{x}_k^l$

      Keep $\mathcal{C}_k^{l'} = \mathcal{C}_{k-1}^{l'}$ for all $l' \geqslant 1$

    else

      Set $\mathcal{D}_k^{l+1}$ according to (4.5.1) and increase $l = l + 1$

    end if

  until $\mathbf{x}_k$ is chosen and then receive reward r_k

end for

of Algorithm 10 is bounded by

$$\operatorname{Regret}(K) \leqslant \frac{2^{14} d \beta^2(l_\Delta) \iota_1(l_\Delta)}{\Delta}.$$

Remark 4.5.2. Since $\beta(l) = \widetilde{\mathcal{O}}(\sqrt{dl})$ and $l_\Delta = \widetilde{\mathcal{O}}(\log(d/\Delta))$, Theorem 4.5.1 suggests that SupLinUCB enjoys a constant regret bound $\widetilde{\mathcal{O}}(d^2 \Delta^{-1})$ when $\zeta \leqslant \widetilde{\mathcal{O}}(\Delta/\sqrt{d})$, which is independent of the total number of rounds K. Note that in Algorithm 11, the choices of λ and β_l do not depend on the sub-optimality gaps Δ and misspecification level ζ.

Remark 4.5.3. When $\zeta \geqslant \Delta/\sqrt{d}$, it is hard to provide a gap-dependent regret bound

due to the large misspecification level ζ. However, a gap-independent regret bound of $\widetilde{\mathcal{O}}(\sqrt{dK} + \sqrt{d}\zeta K \log(K))$ is proved in Takemura et al. (2021), which suggests the performance of SupLinUCB algorithm will not significantly decrease when the condition on misspecification does not hold.

Remark 4.5.4. Comparing the constant factors of DS-OFUL (Algorithm 10) and SupLinUCB (Algorithm 11) on the dominating terms $\widetilde{\mathcal{O}}(\beta^2 d/\Delta)$, one can find that the constant factors of SupLinUCB is significantly larger than DS-OFUL. This is because it takes more samples to learn the first $l_\Delta - 1$ levels in SupLinUCB while DS-OFUL directly learns the l_Δ-th level. Therefore, despite having the same order of constant regret bound (in big-O notation), one can expect that SupLinUCB has a worse performance than DS-OFUL (when Δ is known or can be estimated by grid search).

4.5.3 Key Proof Techniques

Here we provide additional proof techniques besides the techniques discussed in Section 4.4.3. First of all, Lemmas 4.4.5 and 4.4.6, which are built on a single level selected by $\|\mathbf{x}\|_{\mathbf{U}_k^{-1}} \geqslant \Gamma$, can be generalized to the following lemmas for all levels l. The detailed proof are deferred to Section 4.9.3.

Lemma 4.5.5. Set $\lambda = B^{-2}$, for any $k \in [K]$ and $l > 0$, $|\mathcal{C}_k^l| \leqslant 16d4^l\iota_1(l)$, where $\iota_1(l) = \log\left(3LB2^l\right)$.

Lemma 4.5.6. Set $\lambda = B^{-2}$. For any level $l > 0$, for any $\delta > 0$, with probability at least $1-\delta$, for all $k \in [K]$, the prediction error is bounded by

$$\left|\mathbf{x}^\top(\boldsymbol{\theta}_k^l - \boldsymbol{\theta}^*)\right| \leqslant \left(1 + R\sqrt{2d\iota_2(l)} + \zeta\sqrt{\left|\mathcal{C}_k^l\right|}\right)\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}},$$

for all $\mathbf{x}$ such that $\|\mathbf{x}\|_2 \leqslant L$, where $\iota_2(l) = \log((d + |\mathcal{C}_k^l|L^2B^2)/(d\delta))$.

The following two proof techniques are crucial to prove constant regret bound of Algorithm 11.

Optimal arm is never eliminated Considering the optimal arm in the eliminated set, which is defined by $\mathbf{x}_k^{l,*} = \text{argmax}_{\mathbf{x}\in\mathcal{D}_l} r(\mathbf{x})$. Obviously $\mathbf{x}_k^{1,*} = \mathbf{x}_k^*$. The following (informal) lemma says that the decision set always contains a nearly optimal action $\mathbf{x}_k^{l,*}$:

Lemma 4.5.7 (informal)**.** For any level $l > 0$, assume some good events hold, then there exists $\mathbf{x}_k^{l,*} \in \mathcal{D}_k^l$, such that $r(\mathbf{x}_k^*) - r(\mathbf{x}_k^{l,*}) \leqslant 2(l-1)\zeta \left(1 + 4\sqrt{d\iota_1(l)}\right)$ where $\iota_1(l) = \log\left(3LB2^l\right)$.

Given the result of Lemma 4.5.7 and the existence of the sub-optimality gap Δ, we have $\mathbf{x}_k^{l,*} = \mathbf{x}_k^*$ when l is not too large. This means that the optimal arm is never eliminated from the decision set $\mathcal{D}^l$.

Sub-optimal arms are all eliminated Intuitively speaking, at level l, the prediction error is bounded by $\widetilde{\mathcal{O}}(\beta(l) \cdot 2^{-l})$ with some additional misspecification term ζ. Therefore, when we eliminate the arms at level l, the sub-optimality of the arms in $\mathcal{D}^l$ is bounded by the following (informal) lemma:

Lemma 4.5.8 (informal)**.** For any level $l > 0$, for any arm $\mathbf{x} \in \mathcal{D}_k^l$, $r(\mathbf{x}_k^*) - r(\mathbf{x}) \leqslant 6\beta(l)2^{-l} + 2l\zeta \left(1 + 4\sqrt{d\iota_1(l)}\right)$ where $\iota_1(l) = \log\left(3LB2^l\right)$.

Given Lemma 4.5.8, we know that when l is sufficiently large (e.g., larger than l_Δ), all $\mathbf{x} \in \mathcal{D}_k^l$ enjoys a sub-optimality less than Δ. Combining with the existence of sub-optimality gap Δ, we know that all of the sub-optimal arms are eliminated after level l_Δ.

Regret decomposition Given Lemma 4.5.5 and Lemma 4.5.8, the regret over all K rounds can be decomposed into

$$\text{Regret}(K) = \sum_{k=1}^{K} \left(r(\mathbf{x}_k^*) - r(\mathbf{x}_k)\right) = \sum_{l\geqslant 1} \sum_{k\in\mathcal{C}_K^l} \left(r(\mathbf{x}_k^*) - r(\mathbf{x}_k)\right) = \sum_{l=1}^{l_\Delta} \sum_{k\in\mathcal{C}_K^l} \left(r(\mathbf{x}_k^*) - r(\mathbf{x}_k)\right),$$

where the last equality is due to the fact that no regret occurs after $l > l_\Delta$. For each level $l \leqslant l_\Delta$, the summation of the instantaneous regret within $k \in \mathcal{C}_K^l$ can be bounded following

the gap-dependent regret bound of Abbasi-Yadkori et al. (2011) to obtain a $\widetilde{\mathcal{O}}(d^2 \log |\mathcal{C}_K^l|/\Delta)$ regret bound which is independent from K. Then taking the summation over $l \leqslant l_\Delta$ yields the claimed constant regret bound.

4.6 Lower Bound

Following a similar idea in Lattimore et al. (2020), we prove a gap-dependent lower bound for misspecified stochastic linear bandits. Note that stochastic linear bandit can be seen as a special case of linear contextual bandits with a fixed decision set $\mathcal{D}_k = \mathcal{D}$ across all round $k \in [K]$. Similar results and proof can be found in Du et al. (2019) for episodic reinforcement learning.

Theorem 4.6.1 (Lower Bound)**.** Given the dimension d and the number of arms $|\mathcal{D}|$, for any $\Delta \leqslant 1$ and $\zeta \geqslant 3\Delta\sqrt{8\log(|\mathcal{D}|)/(d-1)}$, there exists a set of stochastic linear bandit problems $\boldsymbol{\Theta}$ with minimal sub-optimality gap Δ and misspecification error level ζ, such that for any algorithm that has a sublinear expected regret bound for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, i.e., $\mathbb{E}[\text{Regret}_{\boldsymbol{\theta}}(K)] \leqslant CK^\alpha$ with $C > 0$ and $0 \leqslant \alpha < 1$, we have

- When $K \leqslant \mathcal{O}(|\mathcal{D}|)$, the expected regret is lower bounded by $\mathbb{E}_{\boldsymbol{\theta}\sim\text{Unif.}(\boldsymbol{\Theta})}[\text{Regret}_{\boldsymbol{\theta}}(K)] \geqslant K\Delta$.
- When $K \geqslant \Omega(|\mathcal{D}|)$, the expected regret is lower bounded by $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbb{E}[\text{Regret}_{\boldsymbol{\theta}}(K)] \geqslant \widetilde{\Omega}(|\mathcal{D}| \log(K)\Delta^{-1})$.

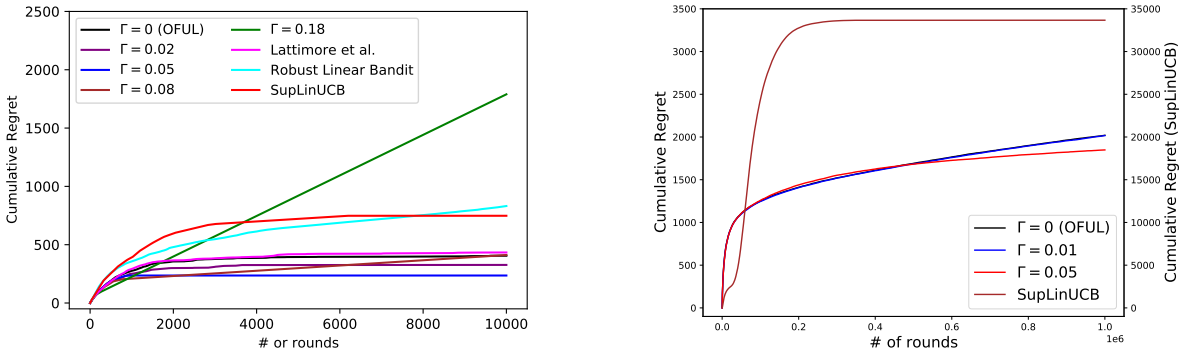
Remark 4.6.2. Theorem 4.6.1 shows two regimes under the case $\zeta \geqslant \widetilde{\Omega}(\Delta/\sqrt{d})$. In the first regime $K \leqslant \mathcal{O}(|\mathcal{D}|)$ where the decision set is large (e.g., $|\mathcal{D}| = d^{100}$), any algorithm will suffer from a linear regret $\widetilde{\mathcal{O}}(\Delta K)$, which suggests that the regime cannot be efficiently learnable. In the second regime $K \geqslant \Omega(|\mathcal{D}|)$, Theorem 4.6.1 suggests an $\widetilde{\Omega}(|\mathcal{D}|\Delta^{-1}\log(K))$ regret lower bound, which is matched by the multi-armed bandit algorithm with an upper bound $\widetilde{\mathcal{O}}(|\mathcal{D}|\Delta^{-1}\log(K))$ (Lattimore and Szepesvári, 2020). Therefore, in this easier regime, linear

function approximation cannot provide any performance improvement and one can simply adopt the multi-armed bandit algorithm to learn the bandit model.

Remark 4.6.3. Theorems 4.4.1 and 4.6.1 provide a holistic picture about the role of misspecification in linear contextual bandits. Here we focus on the more difficult regime $K \leqslant |\mathcal{D}|$. In the regime $K \leqslant |\mathcal{D}|$, when $\zeta \leqslant \widetilde{\mathcal{O}}(\Delta/\sqrt{d})$, Theorem 4.4.1 suggests that the bandit problem is efficiently learnable, and our algorithm DS-OFUL can achieve a constant regret, which improves upon the logarithmic regret bound in the well-specified setting (Abbasi-Yadkori et al., 2011). On the other hand, when $\zeta \geqslant \widetilde{\Omega}(\Delta/\sqrt{d})$, Theorem 4.6.1 provides a linear regret lower bound suggesting that the bandit model can not be efficiently learned.

4.7 Numerical Experiments

To verify the performance improvement by data selection using the UCB bonus in Algorithm 10 and the effectiveness of the parameter-free algorithm Algorithm 11, we conduct experiments for bandit tasks on both synthetic and real-world datasets, which we will de-



(a) On synthetic dataset over 10K rounds

(b) On Asirra dataset over 1M rounds, $\zeta = 0.01$.

Figure 4.2: Cumulative regret of DS-OFUL with different Γ. Results are averaged over 8 runs. In Figure 4.2b for Asirra dataset, the cumulative regret of DS-OFUL (as well as OFUL) can be read from the y-axis on the left. The cumulative regret of SupLinUCB algorithm can be read from the y-axis on the right.

Table 4.2: Averaged cumulative regret and elapsed time of DS-OFUL over 8 runs. The **bold face** value indicates the best (low regret or low elapsed time) for all the algorithm configurations

Algorithm Configuration, (Γ)	Regret (mean±std.)	Regret in last 1k steps	Elapsed Time(sec)
OFUL (Abbasi-Yadkori et al., 2011), $\Gamma = 0$	405.4 ± 76.5	4.94	15.06
DS-OFUL (Algorithm 10), $\Gamma = 0.02$	326.5 ± 68.0	**0.0**	8.59
DS-OFUL (Algorithm 10), $\Gamma = 0.05$	$\mathbf{235.75 \pm 40.3}$	**0.0**	6.30
DS-OFUL (Algorithm 10), $\Gamma = 0.08$	411.6 ± 566.7	22.44	5.97
DS-OFUL (Algorithm 10), $\Gamma = 0.13$	1789.5 ± 1918.8	173.67	**5.56**
Eq. (6) in Lattimore et al. (2020)	433.36 ± 64	1.79	$\geqslant 7$ hrs.
Robust Linear Bandit (Ghosh et al., 2017)	831.5 ± 880.4	42.58	12.85
SupLinUCB (Algorithm 11)	747.9 ± 329.5	**0.0**	31.86

scribe in detail below.

4.7.1 Synthetic Dataset

The synthetic dataset is composed as follows: we set $d = 16$ and generate parameter $\boldsymbol{\theta}^* \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and contextual vectors $\{\mathbf{x}_i\}_{i=1}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ where $N = 100$. The generated parameter and vectors are later normalized to be $\|\boldsymbol{\theta}^*\|_2 = \|\mathbf{x}_i\|_2 = 1$. The reward function is calculated by $r_i = \langle \boldsymbol{\theta}^*, \mathbf{x}_i \rangle + \eta_i$ where $\eta_i \sim \text{Unif}\{-\zeta, \zeta\}$. The contextual vectors and reward function is fixed after generated. The random noise on the receiving rewards ε_t are sampled from the standard normal distribution.

We set the misspecification level $\zeta = 0.02$ and verified that the sub-optimality gap

over the N contextual vectors $\Delta \approx 0.18$. We do a grid search for $\beta = \{1, 3, 10\}$, $\lambda = \{1, 3, 10\}$ [3] and report the cumulative regret of Algorithm 10 with different parameter $\Gamma = \{0, 0.02, 0.05, 0.08, 0.18\}$ over 8 independent trials with total rounds $K = 10000$. It is obvious that when $\Gamma = 0$, our algorithm degrades to the standard OFUL algorithm (Abbasi-Yadkori et al., 2011) which uses data from all rounds into regression.

Besides the OFUL algorithm, we also compare with the algorithm (LSW) in Equation (6) of Lattimore et al. (2020) and the RLB in Ghosh et al. (2017) in Figure 4.2a and Table 4.2. For Lattimore et al. (2020), the estimated reward is updated by $r(\mathbf{x}) = \mathbf{x}^\top \boldsymbol{\theta}_k + \beta \|\mathbf{x}\|_{\mathbf{U}_k^{-1}} + \varepsilon \sum_{s=1}^{k} |\mathbf{x}^\top \mathbf{U}_k^{-1} \mathbf{x}_s^{-1}|$. However, since the time complexity of the LSW algorithm is $\widetilde{\mathcal{O}}(K^2)$ due to the hardness of calculating $\varepsilon \sum_{s=1}^{k} |\mathbf{x}^\top \mathbf{U}_k^{-1} \mathbf{x}_s^{-1}|$ incrementally w.r.t. k. In our setting it takes more than 7 hours for 10000 rounds.

For the RLB algorithm in Ghosh et al. (2017), we did the hypothesis test for $k = 10$ rounds and then decided whether to use OFUL or multi-armed UCB. The results show that both LSW and RLB achieve a worse regret than OFUL since in our setting ζ is relatively small.

The result is shown in Figure 4.2a and the average cumulative regret on the last round is reported in Table 4.2 with its variance over 8 trials. We can see that by setting $\Gamma \approx \Delta/\sqrt{d} \approx 0.18/\sqrt{16} \approx 0.05$, Algorithm 10 can achieve less cumulative regret compared with OFUL ($\Gamma = 0$). The algorithm with a proper choice of Γ also convergences to zero instantaneous regret faster than OFUL. It is also evident that a too large $\Gamma = 0.18 \approx \Delta$ will cause the algorithm to fail to learn the contextual vectors and induce a linear regret. Also, our algorithm shows that using a larger Γ can significantly boost the speed of the algorithm by reducing the number of regressions needed in the algorithm.

Besides the performance improvement achieved by Algorithm 10, the experiments also demonstrates the effectiveness of Algorithm 11. As Table 4.2 suggests, SupLinUCB achieves

[3] By "grid search", we tune the parameter $(\beta, \lambda) = (1, 1), (1, 3), \cdots, (10, 3), (10, 10)$ and see their results.

a zero cumulative regret over the last 1000 steps. However, as discussed in Remark 4.5.4, the total regret of SupLinUCB is much higher than the DS-OFUL and OFUL since it takes more samples to learn the first $l_\Delta - 1$ levels which is not used by DS-OFUL. This constant larger sample complexity could also be verified by a longer elapsed time for executing the SubLinUCB comparing to DS-OFUL.

4.7.2 Real-world Dataset

To demonstrate that the proposed algorithm can be easily applied to modern machine learning tasks, we carried out experiments on the Asirra dataset (Elson et al., 2007). The task of agent is to distinguish the image of cats from the image of dogs. At each round k, the agent receives the feature vector $\boldsymbol{\phi}_{1,k} \in \mathbb{R}^{512}$ of a cat image and another feature vector $\boldsymbol{\phi}_{2,k} \in \mathbb{R}^{512}$ of a dog image. Both feature vectors are generated using ResNet-18 (He et al., 2016) pretrained on ImageNet (Deng et al., 2009). We normalize $\|\boldsymbol{\phi}_{1,k}\|_2 = \|\boldsymbol{\phi}_{2,k}\|_2 = 1$. The agent is required to select the cat from these two vectors. It receives reward $r_t = 1$ if it selects the correct feature vector, and receives $r_t = 0$ otherwise. It is trivial that the sub-optimality gap of this task is $\Delta = 1$. To better demonstrate the influence of misspecification on the performance of the algorithm, we only select the data with $|\boldsymbol{\phi}_i^\top \boldsymbol{\theta}^* - r_i| \leqslant \zeta$ with $r_i = 1$ if it is a cat and $r_i = 0$ otherwise. $\boldsymbol{\theta}^*$ is a pretrained parameter on the whole dataset using linear regression $\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} \sum_{i=1}^N (\boldsymbol{\phi}_i^\top \boldsymbol{\theta} - r_i)^2$, which the agent does not know.

For hyper-parameter tuning, we select $\beta = \{0.1, 0.3, 1\}$ and $\lambda = \{1, 3, 10\}$ by doing a grid search [4] and repeat the experiments for 8 times over 1M rounds for each parameter configuration. As shown in Figure 4.2b, when $\zeta = 0.01$, setting $\Gamma = 0.05 \approx \Delta/\sqrt{d}$ will eventually have a better performance comapred with OFUL algorithm (setting $\Gamma = 0$). On the other hand, the SupLinUCB algorithm (Algorithm 11) will suffer from a much higher, but constant regret bound, which is well aligned with our theoretical result especially Remark 4.5.4. We

[4] By "grid search", we tune the parameter $(\beta, \lambda) = (0.1, 1), (0.1, 3), \cdots, (1, 3), (1, 10)$ and see their results.

Table 4.3: The number of remaining data samples after data processing with expected misspecification level

ζ	# of cats	# of dogs
∞ (without preprocessing)	12500	12500
0.5 (linear separable)	10316	10511
0.1	3182	3248
0.05	2408	2442
0.01	1886	1905

skip the Robust Linear Bandit (Ghosh et al., 2017) algorithm since it is for stochastic linear bandit with fixed contextual features for each arm while here the contextual features are sampled and not fixed. The LSW (Equation (6) in Lattimore et al. (2020) is skipped due to the infeasible executing time.

As a sensitivity analysis, we also set $\zeta = \{0.5, 0.1, 0.05\}$ to test the impact of misspecification on the performance of algorithm choices of Γ. More experiment configurations and results are deferred to Section 4.7.3.

4.7.3 Experiment Details and Additional Results

4.7.3.1 Experiment Configuration

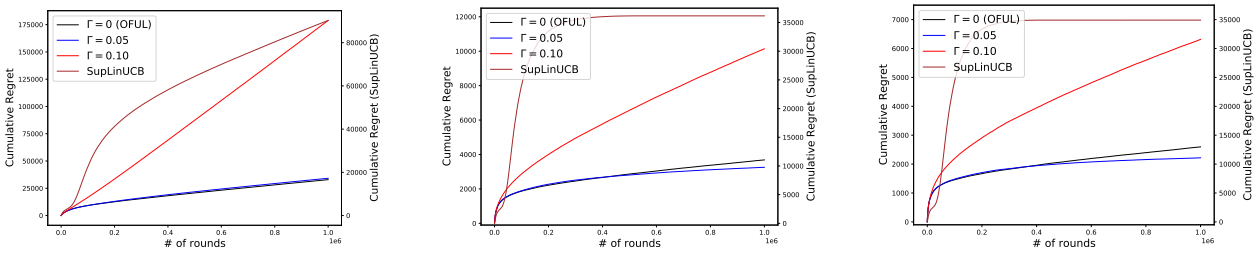
The experiment on synthetic dataset is conducted on Google Colab with a 2-core Intel® Xeon® CPU @ 2.20GHz. The experiment on the real-world Asirra dataset (Elson et al., 2007) is conducted on an AWS p2-xlarge instance.

4.7.3.2 Data Preprocessing for the Asirra Dataset

To demonstrate how our algorithm can deal with different levels of misspecification, we do data preprocessing before feeding the data into the agent. As described in Section 4.7.2, the remaining data with expected misspecification level ζ are shown in Table 4.3. It can be verified that even with the smallest misspecification level, there are still more than 10% of the data is selected.

4.7.3.3 Additional Result on the Asirra Dataset

As a sensitivity analysis, we change the misspecification level in the preprocessing part in the Asirra dataset. The result is shown in Figure 4.3. This result suggests that when the misspecification is small enough, setting $\Gamma = \Delta/\sqrt{d}$ can deliver a reasonable result and SupLinUCB (Chu et al., 2011) can achieve a constant regret bound when $\zeta \leqslant 0.1$. It is aligned with the parameter setting in our Theorem 4.4.1 and the result in our Theorem 4.5.1. Meanwhile, we found that when $\zeta = 0.5$, which means it is strictly larger than the threshold $\Delta/\sqrt{d}$, the algorithm cannot achieve a similar performance with of $\zeta < 0.1$, regardless of the setting of parameter Γ. This also verifies the theoretical understanding of how a large misspecification level will harm the performance of the algorithm.



(a) $\zeta = 0.5$ (b) $\zeta = 0.1$ (c) $\zeta = 0.05$

Figure 4.3: The performance of DS-OFUL under different misspecification levels ζ. Results are averaged over 8 runs, with standard errors shown as shaded areas.

4.8 Conclusion

We study the misspecified linear contextual bandit from a gap-dependent perspective. We propose an algorithm and show that if the misspecification level $\zeta \leqslant \widetilde{\mathcal{O}}(\Delta/\sqrt{d})$, the proposed algorithm, DS-OFUL, can achieve the same gap-dependent regret bound as in the well-specified case. Along with Lattimore et al. (2020); Du et al. (2019), we provide a complete picture on the interplay between misspecification and sub-optimality gap, in which $\Delta/\sqrt{d}$ plays an important role on the phase transition of ζ to decide if the bandit model can be efficiently learned.

Besides the aforementioned constant regret result, DS-OFUL algorithm requires the knowledge of sub-optimality ap Δ. We prove that the SupLinUCB algorithm (Chu et al., 2011) can be viewed as a multi-level version of our algorithm and can also achieve a constant regret with our fine-grained analysis without the knowledge of Δ. Experiments are conducted to demonstrate the performance of the DS-OFUL algorithm and verify the effectiveness of SupLinUCB algorithm.

The promising result suggests a few interesting directions for future research. For example, it would be interesting to incorporate the Lipschitz continuity or smoothness properties of the reward function to derive fine-grained results.

4.9 Proofs

4.9.1 Detailed Proof of Theorem 4.4.1

In this section, we provide detailed proof for Theorem 4.4.1. First, we present a technical lemma to bound the total number of data used in the online linear regression in Algorithm 10.

Lemma 4.9.1 (Restatement of Lemma 4.4.5)**.** Given $0 < \Gamma \leqslant 1$, set $\lambda = B^{-2}$. For any $k \in [K]$, $|\mathcal{C}_k| \leqslant 16d\Gamma^{-2}\log(3LB\Gamma^{-1})$.

Lemma 4.9.1 suggests that up to $\widetilde{O}(d\Gamma^{-2})$ contextual vectors have a UCB bonus greater than Γ. A similar result is also provided in He et al. (2021b), suggesting an $\widetilde{\mathcal{O}}(\Gamma^{-2})$ Uniform-PAC sample complexity. Lemma 4.9.1 also suggests that the numbers of data points added into the regression set $\mathcal{C}$ is finite. Thus, the impact of the noise and the misspecification on the linear regression estimator can be well-controlled.

For a linear regression with up to $|\mathcal{C}_k|$ data points, the next lemma controls the prediction error under misspecification.

Lemma 4.9.2 (Formal statement of Lemma 4.4.6)**.** Let $\lambda = B^{-2}$. For all $\delta > 0$, with probability at least $1-\delta$, for all $\mathbf{x} \in \mathbb{R}^d, k \in [K]$, the prediction error is bounded by:

$$|\mathbf{x}^\top(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)| \leqslant \left(1 + R\sqrt{2d\iota} + \zeta\sqrt{|\mathcal{C}_k|}\right) \|\mathbf{x}\|_{\mathbf{U}_k^{-1}},$$

where $\iota = \log((d+|\mathcal{C}_k|L^2B^2)/(d\delta))$ and $|\mathcal{C}_k|$ is the total number of data used in regression at the k-th round.

Lemma 4.9.2 provides a similar confidence bound as the well-specified linear contextual bandits algorithms like OFUL (Abbasi-Yadkori et al., 2011). Comparing the confidence radius here $\widetilde{\mathcal{O}}(R\sqrt{d} + \zeta\sqrt{|\mathcal{C}_{k-1}|})$ with the conventional radius in OFUL $\widetilde{\mathcal{O}}(R\sqrt{d})$, one can find that there is an additional term $\zeta\sqrt{|\mathcal{C}_k|}$ that is caused by the misspecification. If we directly use all data to do the regression, the resulting confidence radius will be in the order of $\widetilde{\mathcal{O}}(\sqrt{K})$ and therefore will lead to a $\mathcal{O}(K\sqrt{\log K})$ regret bound (see Lemma 11 in Abbasi-Yadkori et al. (2011)). This makes the regret bound vacuous. In our algorithm, however, the confidence radius is only $\sqrt{|\mathcal{C}_k|}$ where $|\mathcal{C}_k|$ is bounded by Lemma 4.9.1. As a result, our regret bound will not be vacuous (i.e., superlinear in K).

When the misspecification level is well bounded by $\zeta = \widetilde{\mathcal{O}}(\Delta/\sqrt{d})$, the following corollary is a direct result of Lemmas 4.9.2 by replacing the term $|\mathcal{C}_k|$ with its upper bound provided in Lemma 4.9.1.

Corollary 4.9.3. Suppose $2\sqrt{d}\zeta\iota_1 \leqslant \Delta$, let $\lambda = B^{-2}$ and $0 < \Gamma \leqslant 1$. Let $\beta = 1 + 2\Delta\Gamma^{-1}\sqrt{\iota_2}/\iota_1 + R\sqrt{2d\iota_3}$ where $\iota_2 = \log(3LB\Gamma^{-1})$, $\iota_3 = \log((1+16L^2B^2\Gamma^{-2}\iota_2)/\delta)$, then with

probability at least $1-\delta$, for all $\mathbf{x} \in \mathbb{R}^d, k \in [K]$, the estimation error for all $k \in [K]$ is bounded by: $|\mathbf{x}^\top(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)| \leqslant \beta \|\mathbf{x}\|_{\mathbf{U}_k^{-1}}$.

Proof. By Lemma 4.9.1, replacing $|\mathcal{C}_k|$ with its upper bound yields

$$|\mathbf{x}^\top(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)| \leqslant (1 + 4\sqrt{d}\zeta\Gamma^{-1}\sqrt{\iota_2} + R\sqrt{2d\iota_3})\|\mathbf{x}\|_{\mathbf{U}_k^{-1}} \leqslant \beta\|\mathbf{x}\|_{\mathbf{U}_k^{-1}},$$

where the second inequality is due to the condition $2\sqrt{d}\zeta \leqslant \Delta/\iota_1$. □

Next we introduce an auxiliary lemma controlling the instantaneous regret bound using the UCB bonus and the misspecification level.

Lemma 4.9.4 (Formal statement of Lemma 4.4.7)**.** Suppose Corollary 4.9.3 holds, for all $k \in [K]$, the instantaneous regret at round k is bounded by

$$\Delta_k(\mathbf{x}_k) = r_k^* - r(\mathbf{x}_k) \leqslant 2\zeta + 2\beta\|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}}.$$

The next technical lemma from He et al. (2021a) bounds the summation of a subset of the bonuses.

Lemma 4.9.5 (Lemma 6.6, He et al. 2021a)**.** For any subset $\mathcal{G} = \{c_1, \cdots, c_i\} \subseteq \mathcal{C}_K$, we have

$$\sum_{k \in \mathcal{G}} \|\mathbf{x}_k\|^2_{\mathbf{U}_k^{-1}} \leqslant 2d\log(1 + |\mathcal{G}|L^2/\lambda).$$

The next auxiliary lemma is used to control the dominating terms.

Lemma 4.9.6. Let $\iota_1 = (24 + 18R)\log((72 + 54R)LB\sqrt{d}\Delta^{-1}) + \sqrt{8R^2\log(1/\delta)}$, $\Gamma = \Delta/(2\sqrt{d}\iota_1)$, $\iota_2 = \log(3LB\Gamma^{-1})$, $\iota_3 = \log((1+16L^2B^2\Gamma^{-2}\iota_2)/\delta)$, we have $\iota_1 > 2+4\sqrt{\iota_2}+R\sqrt{2\iota_3}$.

Equipped with these lemmas, we can start the proof of Theorem 4.4.1.

Proof of Theorem 4.4.1. First, note that by setting $\Gamma = \Delta/(2\sqrt{d}\iota_1)$, the confidence radius β becomes $1 + 4\sqrt{d\iota_2} + R\sqrt{2d\iota_3}$. Then our proof starts by assuming that Corollary 4.9.3 holds

with probability at least $1-\delta$. We decompose the index set $[K]$ into two subsets. The first set is the set of not selected data $[K]\backslash\mathcal{C}_K$, and the second set is the set of selected data $\mathcal{C}_K$. We will bound the cumulative regret within these two sets separately.

First, for those non-selected data $k \notin \mathcal{C}_k$, i.e. $\|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}} < \Gamma$, combining Lemma 4.9.4 with Corollary 4.9.3 yields

$$r_k^* - r(\mathbf{x}_k) < 2\zeta + 2\beta\Gamma = 2\zeta + \frac{\Delta}{\sqrt{d}\iota_1} + \frac{\sqrt{2\iota_3}R\Delta}{\iota_1} + \frac{4\Delta\sqrt{\iota_2}}{\iota_1}, \tag{4.9.1}$$

where $\iota_1, \iota_2, \iota_3$ are the same as Theorem 4.4.1, and the equality is due to $\Gamma = \Delta/(2\sqrt{d}\iota_1)$. When misspecification condition $2\sqrt{d}\zeta \leqslant \Delta/\iota_1$ holds, (4.9.1) suggests that

$$r_k^* - r(\mathbf{x}_k) < \frac{2\Delta}{\sqrt{d}\iota_1} + \frac{4\Delta\sqrt{\iota_2}}{\iota_1} + \frac{\sqrt{2\iota_3}R\Delta}{\iota_1}. \tag{4.9.2}$$

Lemma 4.9.6 suggests that when $\iota_1 = (24+18R)\log((72+54R)LB\sqrt{d}\Delta^{-1}) + \sqrt{8R^2\log(1/\delta)}$ $\iota_1 > 2+4\sqrt{\iota_2}+R\sqrt{2\iota_3}$, (4.9.2) yields that the instantaneous regret $r_k^* - r(\mathbf{x}_k) < \Delta$ at round k. By Definition 4.3.1, the instantaneous regret is zero for all $k \notin \mathcal{C}_k$, indicating the non-selected data incur zero instantaneous regret.

In addition, Lemma 4.9.4 suggests that the instantaneous regret for those $k \in \mathcal{C}_K$ is bounded by

$$\begin{aligned}
\sum_{k\in\mathcal{C}_K} r_k^* - r(\mathbf{x}_k) &\leqslant \sum_{k\in\mathcal{C}_K} \Big(2\beta\|\boldsymbol{\phi}_k\|_{\mathbf{U}_k^{-1}} + 2\zeta\Big) \\
&\leqslant 2\beta\sqrt{|\mathcal{C}_K|}\sqrt{\sum_{k\in\mathcal{C}_K}\|\boldsymbol{\phi}_k\|^2_{\mathbf{U}_k^{-1}}} + 2|\mathcal{C}_K|\zeta \\
&\leqslant 8\beta\Gamma^{-1}\sqrt{d\iota_2}\sqrt{2d\log(1+16d\Gamma^{-2}\iota_2)} + 32\zeta d\Gamma^{-2}\iota_2 \\
&\leqslant 16\beta\sqrt{2d^3\iota_2\log(1+16d\Gamma^{-2}\iota_2)}\iota_1/\Delta + 64\sqrt{d^3}\iota_1\iota_2/\Delta \\
&\leqslant 32\beta\sqrt{2d^3\iota_2\log(1+16d\Gamma^{-2}\iota_2)}\iota_1/\Delta,
\end{aligned} \tag{4.9.3}$$

where the second inequality follows the Cauchy-Schwarz inequality, the third one yields from Lemma 4.9.5 while the fourth utilizes the fact that $\Gamma = \Delta/(2\sqrt{d}\iota_1)$ and $\zeta \leqslant \Delta/(2\sqrt{d}\iota_1)$. The

last one is due to the fact that the second term in the fourth inequality is dominated by the first one.

To warp up, the cumulative regret can be decomposed by

$$\text{Regret}(K) = \sum_{k \notin \mathcal{C}_K} (r_k^* - r(\mathbf{x}_k)) + \sum_{k \in \mathcal{C}_K} (r_k^* - r(\mathbf{x}_k)) \leqslant 0 + \frac{32\beta\sqrt{2d^3\iota_2 \log(1 + 16d\Gamma^{-2}\iota_2)}\iota_1}{\Delta},$$

where the first two zeros are given by the fact that for $k \notin \mathcal{C}_K$, we have $r_k^* - r(\mathbf{x}_k) = 0$. the regret bound for $k \in \mathcal{G}$ is given by (4.9.3). □

4.9.2 Proof of Technical Lemmas in Section 4.9.1

4.9.2.1 Proof of Lemma 4.9.1

The following auxiliary lemma and its corollary are useful

Lemma 4.9.7 (Lemma A.2, Shalev-Shwartz and Ben-David 2014)**.** Let $a \geqslant 1$ and $b > 0$. Then $x \geqslant 4a \log(2a) + 2b$ yields $x \geqslant a \log(x) + b$.

Lemma 4.9.7 can easily indicate the following lemma.

Lemma 4.9.8. Let $a \geqslant 1$. Then $x \geqslant 4 \log(2a) + a^{-1}$ yields $x \geqslant \log(1 + ax)$.

Proof. Let $y = 1 + ax, x = (y - 1)/a$. Then $x \geqslant 4 \log(2a) + a^{-1}$ is equivalent with $y \geqslant 4a \log(2a) + 2$. By Lemma 4.9.7, this implies $y \geqslant a \log(y) + 1$ which is exactly $x \geqslant \log(1 + ax)$. □

Equipped with these technical lemmas, we can start our proof.

Proof of Lemma 4.9.1. Since the cardinality of set $\mathcal{C}_k$ is monotonically increasing w.r.t. k, we fix k to be K in the proof and only provide the bound of $\mathcal{C}_K$. For all selected data $k \in \mathcal{C}_K$, we have $\|\boldsymbol{\phi}_k\|_{\mathbf{U}_k^{-1}} \geqslant \Gamma$. Therefore, when $\Gamma \leqslant 1$, the summation of the bonuses over

data $k \in \mathcal{C}_K$ is lower bounded by

$$\sum_{k \in \mathcal{C}_K} \min\left\{1, \|\boldsymbol{\phi}_k\|^2_{\mathbf{U}_k^{-1}}\right\} \geqslant |\mathcal{C}_K| \min\{1, \Gamma^2\} = |\mathcal{C}_K|\Gamma^2. \tag{4.9.4}$$

On the other hand, Lemma 2.8.15 implies

$$\sum_{k \in \mathcal{C}_K} \min\left\{1, \|\boldsymbol{\phi}_k\|^2_{\mathbf{U}_k^{-1}}\right\} \leqslant 2d \log\left(\frac{\lambda d + |\mathcal{C}_K| L^2}{\lambda d}\right). \tag{4.9.5}$$

Combining (4.9.5) and (4.9.4), the total number of the selected data points $|\mathcal{C}_K|$ is bounded by

$$\Gamma^2 |\mathcal{C}_K| \leqslant 2d \log\left(\frac{\lambda d + |\mathcal{C}_K| L^2}{\lambda d}\right).$$

This result can be re-organized as

$$\frac{\Gamma^2 |\mathcal{C}_K|}{2d} \leqslant \log\left(1 + \frac{2L^2}{\Gamma^2 \lambda} \frac{\Gamma^2 |\mathcal{C}_K|}{2d}\right). \tag{4.9.6}$$

Let $\lambda = B^{-2}$ and since $2L^2B^2 \geqslant 2 \geqslant \Gamma^2$, by Lemma 4.9.8, if

$$\frac{\Gamma^2 |\mathcal{C}_K|}{2d} > 4 \log\left(\frac{4L^2B^2}{\Gamma^2}\right) + 1 \geqslant 4 \log\left(\frac{4L^2B^2}{\Gamma^2}\right) + \frac{\Gamma^2}{2L^2B^2},$$

then (4.9.6) will not hold. Thus the necessary condition for (4.9.6) to hold is

$$\frac{\Gamma^2 |\mathcal{C}_K|}{2d} \leqslant 4 \log\left(\frac{4L^2B^2}{\Gamma^2}\right) + 1 = 8 \log\left(\frac{2LB}{\Gamma}\right) + \log(e) = 8 \log\left(\frac{2LBe^{\frac{1}{8}}}{\Gamma}\right) < 8 \log\left(\frac{3LB}{\Gamma}\right).$$

By basic calculus we get the claimed bound for $|\mathcal{C}_K|$ and complete the proof. □

4.9.2.2 Proof of Lemma 4.9.2

The proof follows the standard technique for linear bandits, we first introduce the self-normalized bound for vector-valued martingales from Abbasi-Yadkori et al. (2011).

Lemma 4.9.9 (Theorem 1, Abbasi-Yadkori et al. 2011)**.** Let $\{\mathcal{F}_t\}_{t=0}^{\infty}$ be a filtration. Let $\{\varepsilon_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that ε_t is $\mathcal{F}_t$-measurable and ε_t is conditionally

R-sub-Gaussian for some $R \geqslant 0$. Let $\{\boldsymbol{\phi}_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$-valued stochastic process such that $\boldsymbol{\phi}_t$ is $\mathcal{F}_{t-1}$ measurable and $\|\boldsymbol{\phi}\|_2 \leqslant L$ for all t. For any $t \geqslant 0$, define $\mathbf{U}_t = \lambda \mathbf{I} + \sum_{k=1}^{t} \boldsymbol{\phi}_k \boldsymbol{\phi}_k$. Then for any $\delta > 0$, with probability at least $1 - \delta$, for all $t \geqslant 0$

$$\left\| \sum_{k=1}^{t} \boldsymbol{\phi}_k \varepsilon_k \right\|^2_{\mathbf{U}_t^{-1}} \leqslant 2R^2 \log \left(\frac{\sqrt{\det(\mathbf{U}_t)}}{\sqrt{\det(\mathbf{U}_0)}\delta} \right).$$

Lemma 4.9.10 (Lemma 8, Zanette et al. 2020c)**.** Let $\{\mathbf{a}_i\}_{i=1}^{d}$ be any sequence of vectors in $\mathbb{R}^d$ and $\{b_i\}_{i=1}^{d}$ be any sequence of scalars such that $|b_i| \leqslant \zeta$. For any $\lambda > 0$:

$$\left\| \sum_{i=1}^{n} \mathbf{a}_i b_i \right\|^2_{\left[\sum_{i=1}^{n} \mathbf{a}_i \mathbf{a}_i^\top + \lambda \mathbf{I}\right]^{-1}} \leqslant n\zeta^2.$$

The next lemma is to bound the perturbation of the misspecification

Lemma 4.9.11. Let $\{\eta_k\}_k$ be any sequence of scalars such that $|\eta_k| \leqslant \zeta$ for any $k \in [K]$. For any index subset $\mathcal{C} \subseteq [K]$, define $\mathbf{U} = \lambda \mathbf{I} + \sum_{k \in \mathcal{C}} \mathbf{x}_k \mathbf{x}_k^\top$, then for any $\mathbf{x} \in \mathbb{R}^d$, we have

$$\left| \mathbf{x}^\top \mathbf{U}^{-1} \sum_{k \in \mathcal{C}} \mathbf{x}_k \eta_k \right| \leqslant \zeta \sqrt{|\mathcal{C}|} \|\mathbf{x}\|_{\mathbf{U}^{-1}}.$$

Proof. By Cauchy-Schwartz inequality we have

$$\left| \mathbf{x}^\top \mathbf{U}^{-1} \sum_{k \in \mathcal{C}} \mathbf{x}_k \eta_k \right| \leqslant \|\mathbf{x}\|_{\mathbf{U}^{-1}} \left\| \sum_{k \in \mathcal{C}} \mathbf{x}_k \eta_k \right\|_{\mathbf{U}^{-1}} \leqslant \zeta \sqrt{|\mathcal{C}|} \|\mathbf{x}\|_{\mathbf{U}^{-1}},$$

where the second inequality dues to lemma 4.9.10. □

The next lemma is the Determinant-Trace inequality.

Lemma 4.9.12. Suppose sequence $\{\mathbf{x}_k\}_{k=1}^{K} \subset \mathbb{R}^d$ and for any $k \in [K]$, $\|\mathbf{x}_k\|_2 \leqslant L$. For any index subset $\mathcal{C} \subseteq [K]$, define $\mathbf{U} = \lambda \mathbf{I} + \sum_{k \in \mathcal{C}} \mathbf{x}_k \mathbf{x}_k^\top$ for some $\lambda > 0$, then $\det(\mathbf{U}) \leqslant (\lambda + |\mathcal{C}|L^2/d)^d$.

Proof. The proof of this lemma is almost the same as Lemma 10 in Abbasi-Yadkori et al. (2011) by replacing the index set $[K]$ with any subset $\mathcal{C}$. We refer the readers to Abbasi-Yadkori et al. (2011) for details. □

Equipped with these lemmas, we can start our proof.

Proof of Lemma 4.9.2. For any $k \in [K]$, considering the data samples $k' \in \mathcal{C}_{k-1}$ used for regression at round k. Following the update rule of $\mathbf{U}_k$ and $\boldsymbol{\theta}_k$ yields

$$\begin{aligned}
\mathbf{U}_k(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*) &= \mathbf{U}_k\mathbf{U}_k^{-1}\bigg(\sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'} r_{k'}\bigg) - \bigg(\lambda\mathbf{I} + \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'}\mathbf{x}_{k'}^\top\bigg)\boldsymbol{\theta}^* \\
&= \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'} r_{k'} - \lambda\boldsymbol{\theta}^* - \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'}\mathbf{x}_{k'}^\top\boldsymbol{\theta}^* \\
&= -\lambda\boldsymbol{\theta}^* + \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'}(r_{k'} - \mathbf{x}_{k'}^\top\boldsymbol{\theta}^*) \\
&= -\lambda\boldsymbol{\theta}^* + \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'}\varepsilon_{k'} + \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'}\eta_{k'},
\end{aligned}$$

where the first equation is due to $\mathbf{U}_k = \lambda\mathbf{I} + \sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_k\mathbf{x}_k^\top$ and $\boldsymbol{\theta}_k = \mathbf{U}_k^{-1}\sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'} r_{k'}$. The last equation follows the fact that $r_{k'}$ is generated from $r_{k'} = r(\mathbf{x}_{k'}) + \varepsilon_{k'} = \mathbf{x}_{k'}^\top\boldsymbol{\theta}^* + \eta(\mathbf{x}_{k'}) + \varepsilon_{k'}$, where we denote $\eta(\mathbf{x}_{k'})$ as $\eta_{k'}$ for the model misspecification error and $\varepsilon_{k'}$ is the random noise. Therefore, consider any contextual vector $\mathbf{x} \in \mathbb{R}^d$, we have

$$\begin{aligned}
\left|\mathbf{x}^\top(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)\right| &= \left|\mathbf{x}^\top\mathbf{U}_k^{-1}\mathbf{U}_k(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)\right| \\
&\leqslant \lambda\underbrace{\left|\mathbf{x}^\top\mathbf{U}_k^{-1}\boldsymbol{\theta}^*\right|}_{q_1} + \underbrace{\left|\mathbf{x}^\top\mathbf{U}_k^{-1}\sum_{k'\in\mathcal{C}_{k-1}} \boldsymbol{\phi}_{k'}\varepsilon_{k'}\right|}_{q_2} + \underbrace{\left|\mathbf{x}^\top\mathbf{U}_k^{-1}\sum_{k'\in\mathcal{C}_{k-1}} \boldsymbol{\phi}_{k'}\eta_{k'}\right|}_{q_3},
\end{aligned}$$

where the inequality is due to the triangle inequality. Lemma 4.9.11 yields that $q_3 \leqslant \zeta\sqrt{|\mathcal{C}_{k-1}|}\|\mathbf{x}\|_{\mathbf{U}_k^{-1}}$. From the fact that $|\mathbf{x}^\top\mathbf{A}\mathbf{y}| \leqslant \|\mathbf{x}\|_{\mathbf{A}}\|\mathbf{y}\|_{\mathbf{A}}$, we can bound term q_1 by

$$q_1 \leqslant \|\mathbf{x}\|_{\mathbf{U}_k^{-1}}\|\boldsymbol{\theta}^*\|_{\mathbf{U}_k^{-1}} \leqslant \lambda^{-1/2}B\|\mathbf{x}\|_{\mathbf{U}_k^{-1}}. \tag{4.9.7}$$

where the last inequality is due to the fact that $\mathbf{U}_k^{-1} \preceq \lambda^{-1}\mathbf{I}$. Term q_2 is also bounded as

$$q_2 \leqslant \|\mathbf{x}\|_{\mathbf{U}_k^{-1}}\bigg\|\sum_{k'\in\mathcal{C}_{k-1}} \mathbf{x}_{k'}\varepsilon_{k'}\bigg\|_{\mathbf{U}_k^{-1}} = \|\mathbf{x}\|_{\mathbf{U}_k^{-1}}\underbrace{\bigg\|\sum_{k'=1}^{K} \mathbb{1}\left[k' \in \mathcal{C}_{k-1}\right]\mathbf{x}_{k'}\varepsilon_{k'}\bigg\|_{\mathbf{U}_k^{-1}}}_{I_1}, \tag{4.9.8}$$

where the second equation uses the indicator function to rewrite the summation over subset $\mathcal{C}_{k-1}$. Denoting $\mathbf{y}_{k'} = \mathbb{1}\left[k' \in \mathcal{C}_{k-1}\right]\mathbf{x}_{k'}$, noticing that $\|\mathbf{y}_{k'}\|_2 \leqslant \|\mathbf{x}_{k'}\|_2 \leqslant L$ and

$$\mathbf{U}_k = \sum_{k' \in \mathcal{C}_{k-1}} \mathbf{x}_{k'}\mathbf{x}_{k'}^\top = \sum_{k'=1}^{K} \mathbb{1}\left[k' \in \mathcal{C}_{k-1}\right]\mathbf{x}_{k'}\mathbf{x}_{k'}^\top = \sum_{k'=1}^{K} \mathbf{y}_{k'}\mathbf{y}_{k'}^\top,$$

by Lemma 4.9.9, I_1 can be further bounded by

$$I_1 \leqslant \sqrt{2R^2 \log\left(\frac{\sqrt{\det(\mathbf{U}_k)}}{\sqrt{\det(\mathbf{U}_0)}\delta}\right)} \leqslant R\sqrt{2\log\left(\frac{\det(\mathbf{U}_k)}{\det(\mathbf{U}_0)\delta}\right)} = R\sqrt{2\log\left(\frac{\det(\mathbf{U}_k)}{\lambda^d\delta}\right)}, \quad (4.9.9)$$

where the second inequality follows the fact that $\det(\mathbf{U}_k) \geqslant \det(\mathbf{U}_0) = \lambda^d$. Notice that $\mathbf{U}_k = \lambda\mathbf{I} + \sum_{k' \in \mathcal{C}_{k-1}} \mathbf{x}_{k'}\mathbf{x}_{k'}^\top$. Lemma 4.9.12 suggests that $\det(\mathbf{U}_k) \leqslant (\lambda + |\mathcal{C}_{k-1}|L^2/d)^d$, plugging this into (4.9.9), we obtain

$$I_1 \leqslant R\sqrt{2\log\left(\frac{(\lambda + |\mathcal{C}_{k-1}|L^2/d)^d}{\lambda^d\delta}\right)} \leqslant R\sqrt{2d\log\left(\frac{d\lambda + |\mathcal{C}_{k-1}|L^2}{d\lambda\delta}\right)}.$$

Plugging the bound of I_1 into (4.9.8) and combining with (4.9.7) and Lemma 4.9.11 together, replacing $|\mathcal{C}_{k-1}|$ with its upper bound $|\mathcal{C}_K|$ we have with probability at least $1-\delta$, for all $k \in [K], \mathbf{x} \in \mathbb{R}^d$,

$$|\mathbf{x}^\top(\boldsymbol{\theta}_k - \boldsymbol{\theta}^*)| \leqslant \left(R\sqrt{2d\log\left(\frac{d\lambda + |\mathcal{C}_K|L^2}{d\lambda\delta}\right)} + B\lambda^{-1/2} + \zeta\sqrt{|\mathcal{C}_K|}\right)\|\boldsymbol{\phi}\|_{\mathbf{U}_k^{-1}}.$$

Letting $\lambda = B^{-2}$ we get the claimed results. □

4.9.2.3 Proof of Lemma 4.9.4

Proof. According to the definition of expected reward function $r(\mathbf{x})$, we have for all $k \in [K]$, suppose the condition in Lemma 4.9.2 holds, then

$$\begin{aligned}
r_k^* - r_k &= \eta(\mathbf{x}_k^*) - \eta(\mathbf{x}_k) + (\mathbf{x}_k^*)^\top \boldsymbol{\theta}^* - \mathbf{x}_k^\top \boldsymbol{\theta}^* \\
&\leqslant 2\zeta + (\mathbf{x}_k^*)^\top \boldsymbol{\theta}^* - \mathbf{x}_k^\top \boldsymbol{\theta}^* \\
&= 2\zeta + (\mathbf{x}_k^*)^\top \boldsymbol{\theta}_k + (\mathbf{x}_k^*)^\top (\boldsymbol{\theta}^* - \boldsymbol{\theta}_k) - \mathbf{x}_k^\top \boldsymbol{\theta}_k + \mathbf{x}_k^\top (\boldsymbol{\theta}_k - \boldsymbol{\theta}^*) \\
&\leqslant 2\zeta + (\mathbf{x}_k^*)^\top \boldsymbol{\theta}_k + \beta \|\mathbf{x}_k^*\|_{\mathbf{U}_k^{-1}} - \mathbf{x}_k^\top \boldsymbol{\theta}_k + \beta \|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}} \\
&\leqslant 2\zeta + \mathbf{x}_k^\top \boldsymbol{\theta}_k + \beta \|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}} - \mathbf{x}_k^\top \boldsymbol{\theta}_k + \beta \|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}} \\
&\leqslant 2\zeta + 2\beta \|\mathbf{x}_k\|_{\mathbf{U}_k^{-1}},
\end{aligned}$$

where the first inequality utilize the fact that $|\eta(\mathbf{x})| \leqslant \zeta$ for all $\mathbf{x} \in \mathcal{D}_k$, the second inequality follows from Corollary 4.9.3, the third inequality is due to the fact that $\mathbf{x}_k = \operatorname{argmax}_{\mathbf{x} \in \mathcal{D}_k} \mathbf{x}^\top \boldsymbol{\theta}_k + \beta \|\mathbf{x}\|_{\mathbf{U}_k^{-1}}$, which is executed in Line 6 of Algorithm 10. □

4.9.2.4 Proof of Lemma 4.9.6

Proof. First it is clear to see that $\sqrt{2\iota_3} = \sqrt{2\log(1 + 16L^2B^2\Gamma^{-2}\iota_2) + 2\log(1/\delta)}$. Using the fact that $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$, it can be further bounded by

$$\sqrt{2\iota_3} \leqslant \sqrt{2\log(1 + 16L^2B^2\Gamma^{-2}\iota_2)} + \sqrt{2\log(1/\delta)}.$$

Assuming $L \geqslant 1, B \geqslant 1, \Gamma = \Delta/(2\sqrt{d\iota_1}) \leqslant 1$ yields $LB\Gamma^{-1} \geqslant 1$, then by basic calculus one can verify that

$$2 + 4\sqrt{\iota_2} \leqslant 6\log(3LB\Gamma^{-1}), \quad \sqrt{2\log(1 + 16L^2B^2\Gamma^{-2}\iota_2)} \leqslant 3\log(3LB\Gamma^{-1}),$$

therefore we have that

$$\begin{aligned}
2 + 4\sqrt{\iota_2} + R\sqrt{2\iota_3} &\leqslant (6 + 3R)\log(3LB\Gamma^{-1}) + \sqrt{2\log(1/\delta)}R \\
&= (6 + 3R)\log(6LB\sqrt{d}\Delta^{-1}\iota_1) + \sqrt{2\log(1/\delta)}R,
\end{aligned}$$

where the last equality is from the fact that $\Gamma = \Delta/(2\sqrt{d}\iota_1)$. Lemma 4.9.7 suggests that the necessary condition for

$$\underbrace{(6LB\sqrt{d}\Delta^{-1})\iota_1}_{x} \geqslant \underbrace{(6LB\sqrt{d}\Delta^{-1})(6+3R)}_{a}\log(6LB\sqrt{d}\Delta^{-1}\iota_1) + \underbrace{(6LB\sqrt{d}\Delta^{-1})\sqrt{2\log(1/\delta)}R}_{b} \tag{4.9.10}$$

is that

$$\begin{aligned}(6LB\sqrt{d}\Delta^{-1})\iota_1 \geqslant 4(6LB\sqrt{d}\Delta^{-1})(6+3R)\log(2(6LB\sqrt{d}\Delta^{-1})(6+3R)) \\ + 2(6LB\sqrt{d}\Delta^{-1})\sqrt{2\log(1/\delta)}R,\end{aligned}$$

which suggests that setting

$$\iota_1 = (24+18R)\log((72+54R)LB\sqrt{d}\Delta^{-1}) + \sqrt{8R^2\log(1/\delta)}$$

implies the fact that $\iota_1 \geqslant 2 + 4\sqrt{\iota_2} + R\sqrt{2\iota_3}$ □

4.9.3 Detailed Proof of Theorem 4.5.1

The first lemma shows that the contexts selected to l-th level are bounded independent from K

Lemma 4.9.13 (Restatement of Lemma 4.5.5)**.** Set $\lambda = B^{-2}$. For any $k \in [K]$ and $l > 0$, $|\mathcal{C}_k^l| \leqslant 16d4^l\iota_1(l)$ where $\iota_1(l) = \log\left(3LB2^l\right)$.

Proof. The proof is similar to the proof of Lemma 4.9.1 by repalcing $\Gamma = 2^{-l}$. □

The next lemma provides a fluctuation control as well as the concentration in the ridge regression

Lemma 4.9.14 (Restatement of Lemma 4.5.6)**.** Set $\lambda = B^{-2}$. For any level $l > 0$, for any $\delta > 0$, with probability at least $1-\delta$, for all $k \in [K]$, the estimation error is bounded by

$$\left|\mathbf{x}^\top(\boldsymbol{\theta}_k^l - \boldsymbol{\theta}^*)\right| \leqslant \left(1 + R\sqrt{2d\iota_2(l)} + \zeta\sqrt{|\mathcal{C}_k^l|}\right)\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}},$$

for all $\mathbf{x}$ such that $\|\mathbf{x}\|_2 \leqslant L$, where $\iota_2(l) = \log((d + |\mathcal{C}_k^l|L^2B^2)/(d\delta))$.

Proof. The proof is similar to the proof of Lemma 4.9.2 □

Combining Lemma 4.9.13 and Lemma 4.9.14, we have the following corollary.

Corollary 4.9.15. Set $\lambda = B^{-2}$. For any $\delta > 0$, with probability at least $1-\delta$, for all round $k \in [K]$ and any level $l > 0$, for all $\mathbf{x}$ such that $\|\mathbf{x}\|_2 \leqslant L$, the prediction error is bounded by

$$\left|\mathbf{x}^\top(\boldsymbol{\theta}_k^l - \boldsymbol{\theta}^*)\right| \leqslant \left(\beta(l) + 4\zeta 2^l\sqrt{d\iota_1(l)}\right)\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}},$$

where $\beta(l) = 1+R\sqrt{2d\iota_2(l)}$, $\iota_2(l) = \log((d2^l+16L^2B^2 8^l\iota_1(l))/(d\delta))$, and $\iota_1(l) = \log\left(3LB2^l\right)$.

Proof. The proof is simply by plugging the result in Lemma 4.9.13 into Lemma 4.9.14 and replacing the δ with $\delta/2^l$. By the union bound over $l \in \mathbb{N}^+$ and the fact that $\sum_{l=1}^{\infty}\delta/2^l = \delta$ yields the claimed result. □

Now, we are about to control $\mathcal{D}_k^l$, which means here we only consider the case where $\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}} \leqslant 2^{-l}$ for all $\mathbf{x} \in \mathcal{D}_k^l$ and assuming the high-probability event in previous subsection always holds. The following lemma suggests that the decision set always keeps a nearly optimal action $\mathbf{x}_k^{l,*}$. Let $\mathcal{G}_K$ be the event that the high probability statement in Corollary 4.9.15 holds.

Lemma 4.9.16 (Formal statement of Lemma 4.5.7)**.** For any level $l > 0$, assume event $\mathcal{G}_K$ holds, then there exists $\mathbf{x}_k^{l,*} \in \mathcal{D}_k^l$, $r(\mathbf{x}_k^*) - r(\mathbf{x}_k^{l,*}) \leqslant 2(l-1)\zeta\left(1+4\sqrt{d\iota_1(l)}\right)$ where $\iota_1(l) = \log\left(3LB2^l\right)$.

Proof. We would prove the statement by induction. Since $\mathcal{D}_k^1 = \mathcal{D}_k$, we have $\mathbf{x}_k^* \in \mathcal{D}_k^1$ and thus the induction basis holds according to $r(\mathbf{x}_k^*) - r(\mathbf{x}_k^{l,*}) = 0$. Now we assume the statement holds for level l, that is, there exists $\mathbf{x}_k^{l,*} \in \mathcal{D}_k^l$ such that $\mathbf{x}_k^{l,*} \in \mathcal{D}_k^l$, $r(\mathbf{x}_k^*) - r(\mathbf{x}_k^{l,*}) \leqslant 2(l-1)\zeta\left(1+4\sqrt{d\iota_1(l)}\right)$.

If $\mathbf{x}_k^{l,*} \in \mathcal{D}_k^{l+1}$, then the desired statement directly holds by choosing $\mathbf{x}_k^{l,*} = \mathbf{x}_k^{l-1,*}$. Otherwise $\mathbf{x}_k^{l,*}$ is eliminated by some action $\mathbf{x}_k^{l+1,*} \in \mathcal{D}_k^l$ that $r_k^l(\mathbf{x}_k^{l+1,*}) \geqslant r_k^l(\mathbf{x}_k^{l,*}) + 2\beta(l)2^{-l}$.

Moreover, from the definition of estimator $r_k^l(\cdot)$, we have

$$r_k^l(\mathbf{x}_k^{l+1,*}) - r(\mathbf{x}_k^{l+1,*}) \leqslant \zeta + \left\langle \mathbf{x}_k^{l+1,*}, \theta_k^l - \theta^* \right\rangle + \beta(l) \left\| \mathbf{x}_k^{l+1,*} \right\|_{(\mathbf{U}_k^l)^{-1}} \tag{4.9.11}$$

and

$$r(\mathbf{x}_k^{l,*}) - r_k^l(\mathbf{x}_k^{l,*}) \leqslant \zeta - \left\langle \mathbf{x}_k^{l,*}, \theta_k^l - \theta^* \right\rangle - \beta(l) \left\| \mathbf{x}_k^{l,*} \right\|_{(\mathbf{U}_k^l)^{-1}}. \tag{4.9.12}$$

Combining (4.9.11) and (4.9.12) and the fact that $r_k^l(\mathbf{x}_k^{l+1,*}) \geqslant r_k^l(\mathbf{x}_k^{l,*}) + 3\beta(l)2^{-l}$ gives that

$$\begin{aligned}
r(\mathbf{x}_k^{l,*}) - r(\mathbf{x}_k^{l+1,*}) &\leqslant -3\beta(l)2^{-l} + 2\zeta + \left\langle \mathbf{x}_k^{l+1,*} - \mathbf{x}_k^{l,*}, \theta_k^l - \theta^* \right\rangle - \beta(l) \left\| \mathbf{x}_k^{l+1,*} \right\|_{(\mathbf{U}_k^l)^{-1}} \\
&\quad + \beta(l) \left\| \mathbf{x}_k^{l,*} \right\|_{(\mathbf{U}_k^l)^{-1}} \\
&\leqslant -3\beta(l)2^{-l} + 2\zeta + 2 \cdot 2^{-l} \left(\beta(l) + 4\zeta 2^l \sqrt{d\iota_1(l)} \right) + \beta(l)2^{-l} \\
&\leqslant 2\zeta \left(1 + 4\sqrt{d\iota_1(l)} \right),
\end{aligned}$$

where the second inequality is suggested by Corollary 4.9.15 and $\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}} \leqslant 2^{-l}$ for all $\mathbf{x} \in \mathcal{D}_k^l$. The desired statement can then be reached using the induction hypothesis. □

Then, the following lemma suggests that the performance of the actions in the decision set is guaranteed.

Lemma 4.9.17 (Formal statement of Lemma 4.5.8)**.** For any level $l > 0$, assume event $\mathcal{G}_K$ holds, then for any action $\mathbf{x} \in \mathcal{D}_k^l$, $r(\mathbf{x}_k^*) - r(\mathbf{x}) \leqslant 6\beta(l)2^{-l} + 2l\zeta \left(1 + 4\sqrt{d\iota_1(l)} \right)$ where $\iota_1(l) = \log\left(3LB2^l\right)$.

Proof. Let $\mathbf{x}_k^{l,*} \in \mathcal{D}_k^l$ be the optimal action given in Lemma 4.9.16. According to the elimination process, for any action $\mathbf{x} \in \mathcal{D}_k^l$, it holds that $r_k^l(\mathbf{x}) \geqslant r_k^l(\mathbf{x}_k^{l,*}) - 3\beta(l)2^{-l}$. Moreover, from the definition of estimator $r_k^l(\cdot)$, we have

$$r_k^l(\mathbf{x}) - r(\mathbf{x}) \leqslant \zeta + \left\langle \mathbf{x}, \theta_k^l - \theta^* \right\rangle + \beta(l) \|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}}$$

and

$$r(\mathbf{x}_k^{l,*}) - r_k^l(\mathbf{x}_k^{l,*}) \leqslant \zeta - \left\langle \mathbf{x}_k^{l,*}, \theta_k^l - \theta^* \right\rangle - \beta(l) \left\| \mathbf{x}_k^{l,*} \right\|_{(\mathbf{U}_k^l)^{-1}}.$$

Combining the above three inequalities give

$$\begin{aligned} r(\mathbf{x}_k^{l,*}) - r(\mathbf{x}) &\leqslant 3\beta(l)2^{-l} + 2\zeta + 2^{-l} + \left\langle \mathbf{x} - \mathbf{x}_k^{l,*}, \theta_k^l - \theta^* \right\rangle - \beta(l) \left\| \mathbf{x}_k^{l,*} \right\|_{(\mathbf{U}_k^l)^{-1}} \\ &\quad + \beta(l) \left\| \mathbf{x}_k^{l-1,*} \right\|_{(\mathbf{U}_k^l)^{-1}} \\ &\leqslant 3\beta(l)2^{-l} + 2\zeta + 2 \cdot 2^{-l} \left(\beta(l) + 4\zeta 2^l \sqrt{d\iota_1(l)} \right) + \beta(l)2^{-l} \\ &\leqslant 6\beta(l)2^{-l} + 2\zeta \left(1 + 4\sqrt{d\iota_1(l)} \right), \end{aligned}$$

where the second inequality is suggested by Corollary 4.9.15 and $\|\mathbf{x}\|_{(\mathbf{U}_k^l)^{-1}} \leqslant 2^{-l}$ for all $\mathbf{x} \in \mathcal{D}_k^l$. The desired statement can then be reached by combining Lemma 4.9.16. □

Proof of Theorem 4.5.1. Consider the case that event $\mathcal{G}_K$ holds. Let l_Δ be the smallest integer solution to $l_\Delta > \log(8\beta(l_\Delta)\Delta^{-1})$. Note this relation ensures $4\beta(l_\Delta)2^{-l_\Delta} < \Delta/2$. In case that the misspecification level is bounded by $2l_\Delta\zeta \left(1 + 4\sqrt{d\iota_1(l_\Delta)} \right) < \Delta/2$, it holds that $6\beta(l_\Delta)2^{-l_\Delta} + 2l_\Delta\zeta \left(1 + 4\sqrt{d\iota_1(l_\Delta)} \right) < \Delta$. According to Lemma 4.9.17, it satisfies that

$$r(\mathbf{x}_k^*) - r(\mathbf{x}) \leqslant 6\beta(l_\Delta)2^{-l_\Delta} + 2l_\Delta\zeta \left(1 + 4\sqrt{d\iota_1(l_\Delta)} \right)$$

for any $\mathbf{x} \in \mathcal{D}_k^{l_\Delta}$. According to the process of arm elimination, we have $\mathcal{D}_k^l \subseteq \mathcal{D}_k^{l_\Delta}$ for any $l \geqslant l_\Delta$. Thus, it holds that $r(\mathbf{x}_k^*) - r(\mathbf{x}) < \Delta$ for any $\mathbf{x} \in \mathcal{D}_k^l, l \geqslant l_\Delta$. Note that according to the definition of Δ, we have $r(\mathbf{x}_k^*) - r(\mathbf{x}) > \Delta$ for all $\mathbf{x} \in \mathcal{D}_k^l$ that $r(\mathbf{x}_k^*) \neq r(\mathbf{x})$. These two statements together restrict $r(\mathbf{x}_k^*) = r(\mathbf{x})$ for any $\mathbf{x} \in \mathcal{D}_k^l$ on every $l > l_\Delta$, that is, any action that remains in the decision sets on higher levels are optimal. Let $\mathcal{U}_K^l$ be the set of index k that action $\mathbf{x}_k$ is chosen from layer l. We have $|\mathcal{U}_K^l| \leqslant |\mathcal{C}_K^l| + 4^l d$. Thus, we could decompose

the total regret by

$$\begin{aligned}
\mathrm{Regret}(K) &= \sum_{l\geqslant 1}\sum_{k\in\mathcal{U}_K^l}(r(\mathbf{x}_k^*)-r(\mathbf{x})) = \sum_{l=1}^{l_\Delta-1}\sum_{k\in\mathcal{U}_K^l}(r(\mathbf{x}_k^*)-r(\mathbf{x}))\\
&\leqslant \sum_{l=1}^{l_\Delta-1}(|\mathcal{C}_K^l|+4^l d)\cdot\left(6\beta(l)2^{-l}+2l\zeta\left(1+4\sqrt{d\iota_1(l)}\right)\right)\\
&\leqslant \sum_{l=1}^{l_\Delta-1}16d4^l\iota_1(l)\cdot\left(6\beta(l)2^{-l}+2l\zeta\left(1+4\sqrt{d\iota_1(l)}\right)\right)\\
&\leqslant 96d\sum_{l=1}^{l_\Delta-1}\beta(l)2^l\iota_1(l)+32d\zeta\sum_{l=1}^{l_\Delta-1}l4^l\iota_1(l)\left(1+4\sqrt{d\iota_1(l)}\right)\\
&\leqslant 96d\beta(l_\Delta)2^{l_\Delta}\iota_1(l_\Delta)+32dl_\Delta 4^{l_\Delta}\iota_1(l_\Delta)\zeta\left(1+4\sqrt{d\iota_1(l_\Delta)}\right)\\
&\leqslant 1536d\beta^2(l_\Delta)\iota_1(l_\Delta)/\Delta+8192d\beta^2(l_\Delta)\iota_1(l_\Delta)/\Delta\\
&\leqslant 2^{14}d\beta^2(l_\Delta)\iota_1(l_\Delta)/\Delta
\end{aligned}$$

where the second equality is given by Lemma 4.9.17, the second inequality is given by Lemma 4.9.13, the third last inequality holds since $\beta(\cdot)$ and $\iota_1(\cdot)$ are monotone increase and the second inequality since $2^{l_\Delta-1}\leqslant 8\beta(l_\Delta-1)\Delta^{-1}\leqslant 8\beta(l_\Delta)\Delta^{-1}$ and $2l_\Delta\zeta\left(1+4\sqrt{d\iota_1(l_\Delta)}\right)<\Delta/2$.

□

4.9.4 Proof of Theorem 4.6.1

To begin with, we introduce the lemma providing a sparse vector set in $\mathbb{R}^d$.

Lemma 4.9.18 (Lemma 3.1, Lattimore et al. 2020)**.** For any $\varepsilon>0$ and $d<[|\mathcal{D}|]$ such that $d\geqslant\lceil 8\log(|\mathcal{D}|)\varepsilon^{-2}\rceil$, there exists a vector set $\mathcal{D}\subset\mathbb{R}^d$ such that $\|\mathbf{x}\|_2=1$ for all $\mathbf{x}\in\mathcal{D}$ and $|\langle\mathbf{x},\mathbf{y}\rangle|\leqslant\varepsilon$ for all $\mathbf{x},\mathbf{y}\in\mathcal{D}$ and $\mathbf{x}\neq\mathbf{y}$.

Next, we present the Bretagnolle–Huber inequality providing the lower bound to distinguish a system.

Lemma 4.9.19 (Bretagnolle–Huber inequality)**.** Let P and Q be probability measures on the same measurable space $(\Omega, \mathcal{F})$, let $\mathcal{A} \in \mathcal{F}$ be an arbitary event. Then

$$P(\mathcal{A}) + Q(\mathcal{A}^c) \geqslant \frac{1}{2}\exp(-\mathrm{KL}(P, Q)).$$

For stochastic linear bandit problem with finite arm, we can denote $T_i(k)$ as the number of rounds the algorithm visit the i-th arm over total k rounds. Then We have the KL-divergence decomposition lemma.

Lemma 4.9.20 (Lemma 15.1, Lattimore and Szepesvári (2020))**.** Let $\nu = (P_1, \cdots, P_n)$ be the reward distributions associated with one n-armed bandit and let $\nu' = (P_1', \cdots, P_n')$ be another n-armed bandit. Fix some algorithm π and let $\mathbb{P}_\nu = \mathbb{P}_{\nu\pi}, \mathbb{P}_{\nu'} = \mathbb{P}_{\nu',\pi}$ be the probability measures on the canonical bandit model induced by the k-round interconnection of π and ν (respectively, π and ν'). Then $\mathrm{KL}(\mathbb{P}_\nu, \mathbb{P}_{\nu'}) = \sum_{i=1}^n \mathbb{E}_\nu[T_i(n)]\mathrm{KL}(P_i, P_i')$

Proof of Theorem 4.6.1. The proof starts from inheriting the idea from Lattimore et al. (2020). Given dimension d and the number of arms $|\mathcal{D}|$, setting $\varepsilon = \sqrt{8\log(|\mathcal{D}|)/(d-1)}$, we can provide the contextual vector set $\mathcal{D}$ such that

$$\|\mathbf{x}\|_2 = 1, \forall \mathbf{x} \in \mathcal{D}, |\langle \mathbf{x}, \mathbf{y}\rangle| \leqslant \sqrt{\frac{8\log(|\mathcal{D}|)}{d-1}}, \forall \mathbf{x}, \mathbf{y} \in \mathcal{D}, \mathbf{x} \neq \mathbf{y},$$

For simplicity, we index the decision set as $\mathbf{x}_1, \cdots, \mathbf{x}_{|\mathcal{D}|}$. Given the minimal sub-optimality gap Δ, we provide the parameter set $\boldsymbol{\Theta}$ as follows:

$$\boldsymbol{\Theta} = \left\{\boldsymbol{\theta}_{(i,j)} = \Delta\mathbf{x}_i + 2\Delta\mathbf{x}_j, \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}, i \neq j\right\} \bigcup \{\boldsymbol{\theta}_i = \Delta\mathbf{x}_i, \mathbf{x}_i \in \mathcal{D}\}.$$

It can be verified that $\boldsymbol{\Theta}$ contains two kinds of $\boldsymbol{\theta}$. The first one $\boldsymbol{\theta}_{(i,j)}$ is a mixture of two different contexts $\mathbf{x}_i, \mathbf{x}_j$ with different strength Δ and 2Δ. The second one is $\boldsymbol{\theta}_i$ which only contains features from one context $\mathbf{x}_i$. We can further verify that the size of $|\boldsymbol{\Theta}| = |\mathcal{D}|^2$ and $\|\boldsymbol{\theta}\|_2 \leqslant \sqrt{5}\Delta$ for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. For different parameter $\boldsymbol{\theta}$, the reward function is sampled from a

Gaussian distribution $\mathcal{N}(r_{\boldsymbol{\theta}}(\mathbf{x}), 1)$, where the expected reward function is defined as

$$r_{\boldsymbol{\theta}_{(i,j)}}(\mathbf{x}) = \begin{cases} 2\Delta \text{ if } \mathbf{x} = \mathbf{x}_j \\ \Delta \text{ if } \mathbf{x} = \mathbf{x}_i \\ 0 \text{ otherwise} \end{cases}, r_{\boldsymbol{\theta}_i}(\mathbf{x}) = \begin{cases} \Delta \text{ if } \mathbf{x} = \mathbf{x}_i \\ 0 \text{ otherwise} \end{cases}.$$

We can verify that the minimal sub-optimality of all these bandit problem is Δ. For different parameter $\boldsymbol{\theta}$ and input $\mathbf{x}$, by utilizing the sparsity of the set $\mathcal{D}$ (i.e. $|\mathbf{x}^\top y| \leqslant \varepsilon$ if $\mathbf{x} \neq \mathbf{y}$), we can verify the misspecification level as

$$|r_{\boldsymbol{\theta}_{(i,j)}}(\mathbf{x}) - \boldsymbol{\theta}_{(i,j)}^\top \mathbf{x}| = \begin{cases} |2\Delta - 2\Delta \mathbf{x}_j^\top \mathbf{x} - \Delta \mathbf{x}_i^\top \mathbf{x}| \leqslant \Delta\varepsilon \text{ if } \mathbf{x} = \mathbf{x}_j \\ |\Delta - 2\Delta \mathbf{x}_j^\top \mathbf{x} - \Delta \mathbf{x}_i^\top \mathbf{x}| \leqslant 2\Delta\epsilon \text{ if } \mathbf{x} = \mathbf{x}_i \\ |0 - 2\Delta \mathbf{x}_j^\top \mathbf{x} - \Delta \mathbf{x}_i^\top \mathbf{x}| \leqslant 3\Delta\varepsilon \text{ otherwise} \end{cases}$$

$$|r_{\boldsymbol{\theta}_i}(\mathbf{x}) - \boldsymbol{\theta}_i^\top(\mathbf{x})| = \begin{cases} |\Delta - \Delta \mathbf{x}_i^\top \mathbf{x}| = 0 \text{ if } \mathbf{x} = \mathbf{x}_i \\ |0 - \Delta \mathbf{x}_i^\top \mathbf{x}| \leqslant \Delta\varepsilon \text{ otherwise.} \end{cases}$$

Therefore we have verified that the misspecification level is bounded by $\zeta = 3\Delta\varepsilon$.

The provided bandit structure is hard for any linear algorithm to learn since any algorithm cannot get any information before it encounters non-zero expected rewards, even regardless of the noise of the rewards. We following the same method in Lattimore and Szepesvári (2020). If the algorithm choose arm i at the first round, there would be $|\mathcal{D}|$ parameters (i.e. $\boldsymbol{\theta}_i, \boldsymbol{\theta}_{(i,\cdot)}$ receiving a non-zero expected reward. On the second round if the algorithm choose a different arm j, there would be $|\mathcal{D}|$ parameters (i.e. $\boldsymbol{\theta}_j, \boldsymbol{\theta}_{(j,k:k\neq i)}$ receiving a non-zero

expected reward. Therefore the average time of receiving zero expected reward should be

$$\begin{aligned}
|\mathcal{D}|^{-2}\sum_{i=1}^{|\mathcal{D}|}(i-1)(|\mathcal{D}|-i+1) &= |\mathcal{D}|^{-2}\sum_{i=0}^{|\mathcal{D}|-1} i(|\mathcal{D}|-i) \\
&= |\mathcal{D}|^{-2}\left(|\mathcal{D}|\sum_{i=0}^{|\mathcal{D}|-1} i - \sum_{i=0}^{|\mathcal{D}|-1} i^2\right) \\
&= |\mathcal{D}|^{-2}\left(\frac{|\mathcal{D}|^2(|\mathcal{D}|-1)}{2} - \frac{|\mathcal{D}|(|\mathcal{D}|-1)(2|\mathcal{D}|-1)}{6}\right) \\
&= \frac{|\mathcal{D}|-1}{2}\left(1-\frac{2|\mathcal{D}|-1}{3|\mathcal{D}|}\right) \\
&\geqslant \frac{|\mathcal{D}|-1}{6},
\end{aligned}$$

where the third equation is from the fact that $\sum_{i=1}^{n} i = n(n+1)/2$ and $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$. The last inequality is from the fact that $2|\mathcal{D}|-1)/(3|\mathcal{D}|) \leqslant 2/3$. Therefore, even without of the random noise, any algorithm is expected to receive $\min\{K, (|\mathcal{D}|-1)/6\}$ uninformative data with expected reward to be zero. Therefore any algorithm will receive a $\Delta\min\{K, (|\mathcal{D}|-1)/6\}$ regret considers the suboptimality as Δ.

Next, we consider the effect of random noise. For any algorithm running on this parameter set $\boldsymbol{\Theta}$, we find two parameter $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_{i,j}$ where $j \neq i$. Define the event as $\mathcal{A} = \{T_j(k) \geqslant k/2\}$ and $\mathcal{A}^c = \{T_j(k) < k/2\}$. By Lemma 4.9.19 and Lemma 4.9.20,

$$\begin{aligned}
\mathbb{P}_{\boldsymbol{\theta}_i}\left(T_j(k) \geqslant \frac{k}{2}\right) + \mathbb{P}_{\boldsymbol{\theta}_{(i,j)}}\left(T_j(k) < \frac{k}{2}\right) &\geqslant \frac{1}{2}\exp(-\mathrm{KL}(\mathbb{P}_{\boldsymbol{\theta}_i}, \mathbb{P}_{\boldsymbol{\theta}_{(i,j)}})) \\
&\geqslant \frac{1}{2}\exp\left(-\sum_{n\in\mathcal{D}}\mathbb{E}_{\boldsymbol{\theta}_i}[T_n(k)]\mathrm{KL}\left(\mathbb{P}_{\boldsymbol{\theta}_{(i,j)},n}, \mathbb{P}_{\boldsymbol{\theta}_j,n}\right)\right).
\end{aligned} \tag{4.9.13}$$

Noticing the minimal sub-optimality gap is Δ. Also the j-th arm is the sub-optimal arm for parameter $\boldsymbol{\theta}_i$. Therefore, once $T_j(k) \geqslant k/2$, the algorithm will at least suffer from $\Delta k/2$ regret for parameter $\boldsymbol{\theta}_i$. Also, since the j-th arm is the optimal arm for bandit $\boldsymbol{\theta}_{(i,j)}$. If $T_j(k) < k/2$, the algorithm will also at least suffer from $\Delta k/2$ regret for $\boldsymbol{\theta}_{(i,j)}$. Denoting

$\mathcal{R}_{\boldsymbol{\theta}}(k)$ as the expected cumulative regret over k rounds, that is to say

$$\mathcal{R}_{\boldsymbol{\theta}_i}(k) \geqslant \frac{\Delta k}{2}\mathbb{P}_{\boldsymbol{\theta}_i}(T_j(k) \geqslant k/2) \quad \mathcal{R}_{\boldsymbol{\theta}_j}(k) \geqslant \frac{\Delta k}{2}\mathbb{P}_{\boldsymbol{\theta}_i}(T_j(k) < k/2). \tag{4.9.14}$$

On the other hand since the bandit using $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ only differ in the j-th arm. Since standard Gaussian noise is adapted, $\mathrm{KL}(\mathbb{P}_{\boldsymbol{\theta}_i,n}, \mathbb{P}_{\boldsymbol{\theta}_{(i,j)},n}) = \Delta^2\, \mathbb{1}[n = j]/2$. Combining this with (4.9.14), (4.9.13) suggests that

$$\mathcal{R}_{\boldsymbol{\theta}_i}(k) + \mathcal{R}_{\boldsymbol{\theta}_j}(k) \geqslant \frac{\Delta k}{2} \exp\left(-\frac{\Delta^2}{2}\mathbb{E}_{\boldsymbol{\theta}_i}\left[T_j(k)\right]\right),$$

which suggests that

$$\mathbb{E}_{\boldsymbol{\theta}_i}\left[T_j(k)\right] \geqslant \frac{\log(\Delta k) - \log 2 - \log(\mathcal{R}_{\boldsymbol{\theta}_i}(k) + \mathcal{R}_{\boldsymbol{\theta}_j}(k))}{\Delta^2/2}, \tag{4.9.15}$$

For any algorithm seeking to get a sublinear expected regret bound of $\mathcal{R}_{\boldsymbol{\theta}}(k) \leqslant Ck^{\alpha}$ with $C > 0, 0 \leqslant \alpha < 1$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, (4.9.15) becomes

$$\mathbb{E}_{\boldsymbol{\theta}_i}\left[T_j(k)\right] \geqslant \frac{\log(\Delta k) - \log 2 - \log(2Ck^{\alpha})}{\Delta^2/2} = \frac{\log(\Delta k) - \log(4C) - \alpha \log k}{\Delta^2/2}. \tag{4.9.16}$$

Since that the regret on $\boldsymbol{\theta}_i$ can be decomposed by

$$\mathcal{R}_{\boldsymbol{\theta}_i}(k) = \Delta \sum_{n=1, n\neq i}^{|\mathcal{D}|} T_n(k), \tag{4.9.17}$$

combining (4.9.17) with (4.9.16) yields

$$\mathcal{R}_{\boldsymbol{\theta}_i}(k) \geqslant \frac{2(|\mathcal{D}| - 1)}{\Delta} \max\left\{\log(\Delta k) - \log(4C) - \alpha \log k, 0\right\},$$

where the max operator is trivially taken for $\mathcal{R}_{\boldsymbol{\theta}}(k) \geqslant 0$.

□

CHAPTER 5

Uncertainty-Aware Robust Reinforcement Learning via Certified Estimator

5.1 Introduction

In Chapter 4, we discussed a data selection method and a phased algorithm that can handle the misspecified bandit tasks and deliver a constant regret bound. In this chapter, we move on to a more general reinforcement learning setting.

Reinforcement learning (RL) has been a popular approach for teaching agents to make decisions based on feedback from the environment. RL has shown great success in a variety of applications, including robotics (Kober et al., 2013), gaming (Mnih et al., 2013), and autonomous driving. In the most of these applications, there is a common expectation that RL agents will master tasks while making only a bounded number of mistakes, even over indefinite runs. However, theoretical support of this expectation is limited in the RL theory literature: in the instance-independent case, Jin et al. (2020b); Ayoub et al. (2020); Wang et al. (2019), provided only $\widetilde{\mathcal{O}}(\sqrt{K})$ regret upper bounds; in the instance-dependent setting, Simchowitz and Jamieson (2019); Yang et al. (2021); He et al. (2021a) provided logarithmic $\widetilde{\mathcal{O}}(\Delta^{-1} \log K)$ high-probability regret upper bounds for both tabular MDPs and MDPs with linear function approximations, given a suboptimality gap Δ. However, these findings suggest that an agent's regret increases with the number of episodes K, contradicting the practical expectation of finite mistakes. Conversely, recent years have witnessed a series of work providing a constant regret bound for RL and bandits, suggesting that an RL

agent's regret may remain bounded even when it faces an indefinite number of episodes. Papini et al. (2021a); Zhang et al. (2021a) have provided instance-dependent constant regret bounds under the assumption of prior data distribution. However, verifying these data distribution assumptions can be difficult or infeasible. On the other hand, it is known that high-probability constant regret bounds can be achieved unconditionally in multi-armed bandits (Abbasi-Yadkori et al., 2011) and in contextual linear bandits if and only if the misspecification is sufficiently small with respect to the minimal sub-optimality gap (Zhang et al., 2023c). This raises a critical question:

Is it possible to design a reinforcement learning algorithm that incurs only constant regret under minimal assumptions?

To answer this question, we introduce a algorithm, which we refer to as Cert-LSVI-UCB, for reinforcement learning with linear function approximation. To encompass a broader range of real-world scenarios characterized by large state-action spaces and the need for function approximations, we adapt the *misspecified linear MDP* (Jin et al., 2020b) setting, where both the transition kernel and reward function can be approximated by a linear function with approximation error ζ. We show that, with our innovative design of certified estimator and novel analytical techniques, Cert-LSVI-UCB achieves constant regret without relying on any prior assumptions on data distributions.

5.1.1 Organization of this Chapter

This chapter is organized as follows: we discuss the related work in Section 5.2 and the preliminaries in Section 5.3. In Section 5.4, we present Cert-LSVI-UCB which leverages a certified estimator to guarantee the robustness of the estimation of the value function. In Section 5.5, we present the regret analysis for Cert-LSVI-UCB. We highlight several key techniques in Section 5.6 and draw the conclusion in Section 5.7. The detailed proof is deferred to Section 5.8.

Algorithm	Misspecified MDP?	Result
`LSVI-UCB` (He et al., 2021a)	×	$\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1}\log(K))$
`LSVI-UCB` (Papini et al., 2021a)	×	$\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1}\log(1/\lambda))$
Cert-LSVI-UCB (ours, Theorem 5.5.1)	✓	$\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1})$

Table 5.1: Instance-dependent regret bounds for different algorithms under the linear MDP setting. Here d is the dimension of the linear function $\boldsymbol{\phi}(s,a)$, H is the horizon length, Δ is the minimal suboptimality gap. All results in the table represent high probability regret bounds. The regret bound depends the number of episodes K in He et al. (2021a) and the minimum positive eigenvalue λ of features mapping in Papini et al. (2021b). **Misspecified MDP?** indicates if the algorithm can (✓) handle the misspecified linear MDP or not (×).

5.2 Related Work

Instance-dependent regret bound in RL. Although most of the theoretical RL works focus on worst-case regret bounds, instance-dependent (a.k.a., problem-dependent, gap-dependent) regret bound is another important bound to understanding how the hardness of different instance can affect the sample complexity of the algorithm. For tabular MDPs, Jaksch et al. (2010) proved a $\widetilde{\mathcal{O}}(D^2S^2A\Delta^{-1}\log K)$ instance-dependent regret bound for average-reward MDP where D is the diameter of the MDP and Δ is the policy suboptimal gap. Simchowitz and Jamieson (2019) provided a lower bound for episodic MDP which suggests that the any algorithm will suffer from $\Omega(\Delta^{-1})$ regret bound. Yang et al. (2021) analyzed the optimistic Q-learning and proved a $\mathcal{O}(SAH^6\Delta^{-1}\log K)$ logarithmic instance-dependent regret bound. In the domain of linear function approximation, He et al. (2021a) provided instance-dependent regret bounds for both linear MDPs (i.e., $\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1}\log K)$) and linear mixture MDPs (i.e., $\widetilde{\mathcal{O}}(d^2H^5\Delta^{-1}\log K)$). Furthermore, Dann et al. (2021) provided an improved analysis for this instance-dependent result with a redefined suboptimal gap. Zhang et al. (2023b) proved a similar logarithmic instance-dependent bound with He

et al. (2021a) in misspecified linear MDPs, showing the relationship between misspecification level and suboptimality bound. Despite all these bounds are logarithmic depended on the number of episode K, many recent works are trying to remove this logarithmic dependence. Papini et al. (2021a) showed that under the linear MDP assumption, when the distribution of contexts $\boldsymbol{\phi}(s, a)$ satisfies the 'diversity assumption' (Hao et al., 2020) called 'UniSOFT', then LSVI-UCB algorithm may achieve an expected constant regret w.r.t. K. Zhang et al. (2021a) showed a similar result on bilinear MDP (Yang and Wang, 2020b), and extended this result to offline setting, indicating that the algorithm only need a finite offline dataset to learn the optimal policy. Table 5.1 summarizes the most relevant results mentioned above for the ease of comparison with our results.

RL with model misspecification. All of the aforementioned works consider the well-specified setting and ignore the approximation error in the MDP model. To better understand this misspecification issue, Du et al. (2019) showed that having a good representation is insufficient for efficient RL unless the approximation error (i.e., misspecification level) by the representation is small enough. In particular, Du et al. (2019) showed that an $\widetilde{\Omega}(\sqrt{H/d})$ misspecification will lead to $\Omega(2^H)$ sample complexity for RL to identify the optimal policy, even with a generative model. On the other hand, a series of work (Jin et al., 2020b; Zanette et al., 2020b,a) provided $\widetilde{\mathcal{O}}(\sqrt{K} + \zeta K)$-type regret bound for RL in various settings, where ζ is the misspecification level[1] and we ignore the dependence on the dimension of the feature mapping d and the planing horizon H for simplicity. These algorithms, however, require the knowledge of misspecification level ζ, thus are not *parameter-free*. Another concern for these algorithms is that some of the algorithms (Jin et al., 2020b) would possibly suffer from a *trivial asymptotic regret*, i.e., $\text{Regret}(k) > \omega(k\zeta \cdot \text{poly}(d, H, \log(1/\delta)))$, as suggested by Vial et al. (2022). This means the performance of the RL algorithm will possibly degenerate as

[1]The misspecification level for these upper bounds is measured in the total variation distance between the ground truth transition kernel and approximated transition kernel, which is strictly stronger than the infinite-norm misspecification used in Du et al. (2019).

the number of episodes k grows. To tackle these two issues, Vial et al. (2022) propose the `Sup-LSVI-UCB` algorithm which requires a parameter ε_{tol}. When $\varepsilon_{\text{tol}} = d/\sqrt{K}$, the proposed algorithm is *parameter-free* but will have a trivial *asymptotic regret bound*. When $\varepsilon_{\text{tol}} = \zeta$, the algorithm will have a non-trivial *asymptotic regret bound* but is not *parameter-free* since it requires knowledge of the misspecification level. Another series of works (He et al., 2022b; Lykouris et al., 2021; Wei et al., 2022) are working on the *corruption robust* setting. In particular, Lykouris et al. (2021); Wei et al. (2022) are using the *model-selection* technique to ensure the robustness of RL algorithms under adversarial MDPs.

5.3 Preliminaries

We consider episodic Markov Decision Processes denoted by $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, \{r_h\}, \{\mathbb{P}_h\})$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the finite action space, H is the length of each episode, $r_h : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is the reward function at stage h and $\mathbb{P}_h(\cdot|s, a)$ is the transition probability function at stage h. The policy $\pi = \{\pi_h\}_{h=1}^H$ denotes a set of policy functions $\pi_h : \mathcal{S} \mapsto \mathcal{A}$ for each stage h. For given policy π, we define the state-action value function $Q_h^\pi(s, a)$ and the state value function $V_h^\pi(s)$ as

$$Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}\Big[\sum_{h'=h+1}^H r_{h'}\big(s_{h'}, \pi_{h'}(s_{h'})\big) \;\Big|\; s_h = s, a_h = a\Big], V_h^\pi(s) = Q_h^\pi\big(s, \pi_h(s)\big),$$

where $s_{h'+1} \sim \mathbb{P}_h(\cdot|s_{h'}, a_{h'})$. The optimal state-action value function Q_h^* and the optimal state value function V_h^* are defined by $Q_h^*(s, a) = \max_\pi Q_h^\pi(s, a), V_h^*(s) = \max_\pi V_h^\pi(s)$.

By definition, both the state-action value function $Q_h^\pi(s, a)$ and the state value function $V_h^\pi(s)$ are bounded by $[0, H]$ for any state s, action a and stage h. For any function $V : \mathcal{S} \mapsto \mathbb{R}$, we denote by $[\mathbb{P}_h V](s, a) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} V(s')$ the expected value of V after transitioning from state s given action a at stage h and $[\mathbb{B}_h V](s, a) = r_h(s, a) + [\mathbb{P}_h V](s, a)$ where $\mathbb{B}$ is referred to as the *Bellman operator*. For each stage $h \in [H]$ and policy π, the Bellman

equation, as well as the Bellman optimality equation, are presented as follows

$$Q_h^\pi(s,a) = r_h(s,a) + [\mathbb{P}_h V_{h+1}^\pi](s,a) := [\mathbb{B}_h V_{h+1}^\pi](s,a),$$
$$Q_h^*(s,a) = r_h(s,a) + [\mathbb{P}_h V_{h+1}^*](s,a) := [\mathbb{B}_h V_{h+1}^*](s,a).$$

We use regret to measure the performance of RL algorithms. It is defined as $\text{Regret}(K) = \sum_{k=1}^K \left(V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)\right)$, where π^k represents the agent's policy at episode k. This definition quantifies the cumulative difference between the expected rewards that could have been obtained by following the optimal policy and those achieved under the agent's policy across the first K episodes, measuring the total loss in performance due to suboptimal decisions.

We consider linear function approximation in this work, where we adopt the *misspecified linear MDP* assumption, which is firstly proposed in Jin et al. (2020b).

Assumption 5.3.1 (ζ-Approximate Linear MDP, Jin et al. 2020b)**.** For any $\zeta \leqslant 1$, we say a MDP $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, \{r_h\}, \{\mathbb{P}_h\})$ is a ζ-*approximate linear MDP* with a feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}^d$, if for any $h \in [H]$, there exist d *unknown* (signed) measures $\boldsymbol{\mu}_h = \left(\mu_h^{(1)}, \cdots, \mu_h^{(d)}\right)$ over $\mathcal{S}$ and an unknown vector $\boldsymbol{\theta}_h \in \mathbb{R}^d$ such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\left\|\mathbb{P}_h(\cdot|s,a) - \langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}_h(\cdot)\rangle\right\|_{\text{TV}} \leqslant \zeta, \quad \left|r_h(s,a) - \langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h\rangle\right| \leqslant \zeta,$$

w.l.o.g. we assume $\forall (s,a) \in \mathcal{S} \times \mathcal{A} : \|\boldsymbol{\phi}(s,a)\| \leqslant 1$ and $\forall h \in [H] : \|\boldsymbol{\mu}_h(\mathcal{S})\| \leqslant \sqrt{d}, \|\boldsymbol{\theta}_h\| \leqslant \sqrt{d}$.

The ζ-*approximate linear MDP* suggests that for any policy π, the state-action value function Q_h^π can be approximated by a linear function of the given feature mapping $\boldsymbol{\phi}$ up to some misspecification level, which is summarized in the following proposition.

Proposition 5.3.2 (Lemma C.1, Jin et al. 2020b)**.** For a ζ-*approximate linear MDP*, for any policy π, there exist corresponding weights $\{\mathbf{w}_h^\pi\}_{h\in[H]}$ where $\mathbf{w}_h^\pi = \boldsymbol{\theta}_h + \int V_{h+1}^\pi(s')\mathrm{d}\boldsymbol{\mu}_h(s')$ such that for any $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$, $\left|Q_h^\pi(s,a) - \langle \boldsymbol{\phi}(s,a), \mathbf{w}_h^\pi\rangle\right| \leqslant 2H\zeta$. We have $\|wb_h^\pi\|_2 \leqslant 2H\sqrt{d}$.

Next, we introduce the definition of the suboptimal gap as follows.

Definition 5.3.3 (Minimal suboptimality gap)**.** For each $s \in \mathcal{S}, a \in \mathcal{A}$ and step $h \in [H]$, the suboptimality gap $\text{gap}_h(s,a)$ is defined by $\Delta_h(s,a) = V_h^*(s) - Q_h^*(s,a)$ and the minimal suboptimality gap Δ is defined by $\Delta = \min_{h,s,a}\big\{\Delta_h(s,a) : \Delta_h(s,a) \neq 0\big\}$.

Notably, a task with a larger Δ means it is easier to distinguish the optimal action $\pi_h^*(s)$ from other actions $a \in \mathcal{A}$, while a task with lower gap Δ means it is more difficult to distinguish the optimal action.

5.4 Proposed Algorithms

5.4.1 Main algorithm: Cert-LSVI-UCB

We begin by introducing our main algorithm Cert-LSVI-UCB, which is a modification of the `Sup-LSVI-UCB` (Vial et al., 2022). As presented in Algorithm 12, for each episode k, our algorithm maintains a series of index sets $\mathcal{C}_{k,h}^l$ for each stage $h \in [H]$ and phase l. The algorithm design ensures that for any episode k, the maximum number of phases l is bounded by $L_k \leqslant \max\{\lceil \log_4(k/d) \rceil, 0\}$. During the exploitation step, for each phase l associated with the index set $\mathcal{C}_{k-1,h}^l$, the algorithm constructs the estimator vector $\mathbf{w}_{h,l}^k$ by solving the following ridge regression problem in Line 6 and Line 7:

$$\mathbf{w}_{h,l}^k \leftarrow \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d} \lambda\|\mathbf{w}\|_2^2 + \sum_{\tau\in\mathcal{C}_{h,l}^{k-1}} \big(\mathbf{w}^\top \boldsymbol{\phi}_h^\tau - r_h^\tau - \widehat{V}_{h+1}^k(s_{h+1}^\tau)\big)^2.$$

After calculating the estimator vector $\mathbf{w}_{h,l}^k$ in Line 8, the algorithm quantilizes $\mathbf{w}_{h,l}^k$ and $(\mathbf{U}_{h,l}^k)^{-1}$ to the precision of κ_l. Similar to `Sup-LSVI-UCB` (Vial et al., 2022), we note $\widetilde{\mathbf{U}}_{h,l}^{k,-1}$ is the quantized version of inverse covariance matrix $(\mathbf{U}_{h,l}^k)^{-1}$ rather than the inverse of quantized covariance matrix $(\widetilde{\mathbf{U}}_{h,l}^k)^{-1}$. The main difference between our implementation and that in Vial et al. (2022) is that we use a layer-dependent quantification precision κ_l instead of the global quantification precision $\kappa = 2^{-4L}/d$, which enables our algorithm get rid of the dependence on $\mathcal{O}(\log K)$ in the maximum number of phases L_k.

Algorithm 12 Cert-LSVI-UCB

1: Set $V_{H+1}^k(s) = 0$ for all $(s,k) \in \mathcal{S} \times [K]$, $\mathcal{C}_{h,l}^k = \varnothing$ for all $(h,l) \in [H] \times \mathbb{N}^+$, $\lambda = 16$

2: **for** episode $k = 1, \cdots, K$ **do**

3: Set $L_k = \max\{\lceil \log_4(k/d) \rceil, 0\}$

4: **for** step $h = H, \cdots, 1$ **do**

5: **for** phase $l = 1, \cdots, L_k + 1$ **do**

6: $\mathbf{U}_{h,l}^k = \lambda \mathbf{I} + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau (\boldsymbol{\phi}_h^\tau)^\top$

7: $\mathbf{w}_{h,l}^k = (\mathbf{U}_{h,l}^k)^{-1} \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big(r_h^\tau + \widehat{V}_{h+1}^k(s_{h+1}^\tau)\big)$

8: $\widetilde{\mathbf{U}}_{h,l}^{k,-1} = \kappa_l \big\lceil (\mathbf{U}_{h,l}^k)^{-1} / \kappa_l \big\rfloor$, $\widetilde{\mathbf{w}}_{h,l}^k = \kappa_l \big\lceil \mathbf{w}_{h,l}^k / \kappa_l \big\rfloor$ where $\kappa_l = 0.01 \cdot 2^{-4l} d^{-1}$

9: **end for**

10: $\widehat{V}_h^k(s_h^\tau), \cdot, \cdot, \cdot = \text{Cert-LinUCB}(s_h^\tau; \{\widetilde{\mathbf{w}}_{h,l}^k\}_l, \{\widetilde{\mathbf{U}}_{h,l}^{k,-1}\}_l, L_k)$ for all $\tau \in [k-1]$

11: **end for**

12: Observe $s_1^k \in \mathcal{S}$

13: **for** step $h = 1, \cdots, H$ **do**

14: $\cdot, \pi_h^k(s_h^k), l_h^k(s_h^k), f_h^k(s_h^k) = \text{Cert-LinUCB}(s_h^k; \{\widetilde{\mathbf{w}}_{h,l}^k\}_l, \{\widetilde{\mathbf{U}}_{h,l}^{k,-1}\}_l, L_k)$

15: $\mathcal{C}_{h,l_h^k(s_h^k)}^k = \mathcal{C}_{h,l_h^k(s_h^k)}^{k-1} \cup \{k\}$ **if** $f_h^k(s_h^k) = 1$ **else** $\mathcal{C}_{h,l_h^k(s_h^k)}^{k-1}$

16: $\mathcal{C}_{h,l}^k = \mathcal{C}_{h,l}^{k-1}$ for all $l \neq l_h^k(s_h^k)$

17: Play $\pi_h^k(s_h^k)$, set $\boldsymbol{\phi}_h^k = \boldsymbol{\phi}\big(s_h^k, \pi_h^k(s_h^k)\big)$, receive r_h^k and observe $s_{h+1}^k \in \mathcal{S}$

18: **end for**

19: **end for**

After obtaining $\widetilde{\mathbf{w}}_{h,l}^k$ and $\widetilde{\mathbf{U}}_{h,l}^{k,-1}$, a subroutine, Cert-LinUCB, is called to calculate an optimistic value function $\widehat{V}_h^k(s_h^\tau)$ for all historical states s_h^τ in Line 10. Then the algorithm transits to stage $h-1$ and iteratively computes $\widetilde{\mathbf{w}}_{h,l}^k$ and $\widetilde{\mathbf{U}}_{h,l}^{k,-1}$ for all phase l and stage $h \in [H]$.

In the exploration step, the algorithm starts to do planning from the initial state s_1^k. For each observed state s_h^k, the same subroutine, Cert-LinUCB, will be called in Line 14 for the policy $\pi_h^k(s_h^k)$, the corresponding phase $l_h^k(s_h^k)$, and a flag $f_h^k(s_h^k)$. If the flag $f_h^k(s_h^k) = 1$, the

Algorithm 13 Cert-LinUCB : $\big(s; \{\widetilde{\mathbf{w}}_{h,l}^k\}_l, \{\widetilde{\mathbf{U}}_{h,l}^{k,-1}\}_l, L\big) \mapsto \big(\widehat{V}_h^k(s), \pi_h^k(s), l_h^k(s), f_h^k(s)\big)$

1: **input:** $s \in \mathcal{S}, \forall l : \widetilde{\mathbf{w}}_{h,l}^k \in \mathbb{R}^d, \widetilde{\mathbf{U}}_{h,l}^{k,-1} \in \mathbb{R}^{d\times d}, L \in \mathbb{N}^+$
2: **output:** $\widehat{V}_h^k(s) \in \mathbb{R}, \pi_h^k(s) \in \mathcal{A}, l_h^k(s) \in \mathbb{N}^+, f_h^k(s) \in \{0,1\}$
3: $\mathcal{A}_{h,1}^k(s) = \mathcal{A}, \check{V}_{h,0}^k(s) = 0, \widehat{V}_{h,0}^k(s) = H$
4: **for** phase $l = 1, \cdots, L+1$ **do**
5: Set $Q_{h,l}^k(s,a) = \big\langle \boldsymbol{\phi}(s,a), \widetilde{\mathbf{w}}_{h,l}^k \big\rangle$
6: Set $\pi_{h,l}^k(s) = \mathrm{argmax}_{a\in\mathcal{A}_{h,l}^k} Q_{h,l}^k(s,a), V_{h,l}^k(s) = Q_{h,l}^k\big(s, \pi_{h,l}^k(s)\big)$
7: **if** $l > L$ **then**
8: **return** $\big(\widehat{V}_h^k(s), \pi_h^k(s), l_h^k(s), f_h^k(s)\big) = \big(\widehat{V}_{h,l-1}^k(s), \pi_{h,l-1}^k(s), l, 1\big)$
9: **else if** $\gamma_l \cdot \max_{a\in\mathcal{A}_{h,l}^k(s)} \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}_{h,l}^{k,-1}} \geqslant 2^{-l}$ **then**
10: **return** $\big(\widehat{V}_h^k(s), \pi_h^k(s), l_h^k(s), f_h^k(s)\big) = \big(\widehat{V}_{h,l-1}^k(s), \mathrm{argmax}_{a\in\mathcal{A}_{h,l}^k(s)} \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}_{h,l}^{k,-1}}, l, 1\big)$
11: **else if** $\max\big\{V_{h,l}^k(s) - 3\cdot 2^{-l}, \check{V}_{h,l-1}^k(s)\big\} > \min\big\{V_{h,l}^k(s) + 3\cdot 2^{-l}, \widehat{V}_{h,l-1}^k(s)\big\}$ **then**
12: **return** $\big(\widehat{V}_h^k(s), \pi_h^k(s), l_h^k(s), f_h^k(s)\big) = \big(\widehat{V}_{h,l-1}^k(s), \pi_{h,l-1}^k(s), l, 0\big)$
13: **else**
14: $\widehat{V}_{h,l}^k(s) = \min\big\{V_{h,l}^k(s) + 3\cdot 2^{-l}, \widehat{V}_{h,l-1}^k(s)\big\}$
15: $\check{V}_{h,l}^k(s) = \max\big\{V_{h,l}^k(s) - 3\cdot 2^{-l}, \check{V}_{h,l-1}^k(s)\big\}$
16: $\mathcal{A}_{h,l+1}^k(s) = \Big\{a \in \mathcal{A}_{h,l}^k(s) : Q_{h,l}^k(s,a) \geqslant V_{h,l}^k(s) - 4\cdot 2^{-l}\Big\}$
17: **end if**
18: **end for**

algorithm adds the index k to the index set $\mathcal{C}_{h,l_h^k(s_h^k)}^k$ in Line 15. Otherwise, the algorithm skips the current index k and all index sets remain unchanged. Finally, the algorithm executes policy $\pi_h^k(s_h^k)$, receives reward r_h^k and observes the next state s_{h+1}^k in Line 17.

5.4.2 Subroutine: Cert-LinUCB

Next we introduce subroutine Cert-LinUCB, improved from `Sup-Lin-UCB-Var` (Vial et al., 2022) that computes the optimistic value function $\widehat{V}_h^k$. The algorithm is described as

follows. Starting from phase $l = 1$, the algorithm first calculates the estimated state-action function $Q_{h,l}^k(s,a)$ as a linear function over the quantified parameter $\widetilde{\mathbf{w}}_{h,l}^k$ and feature mapping $\boldsymbol{\phi}(s,a)$, following Proposition 5.3.2. After calculating the estimated state-action value function $Q_{h,l}^k(s)$, the algorithm computes the greedy policy $\pi_{h,l}^k(s)$ and its corresponding value function $V_{h,l}^k(s)$.

Similar to `Sup-Lin-UCB-Var` (Vial et al., 2022), our algorithm has several conditions starting from Line 7 to determine whether to stop at the current phase or to eliminate the actions and proceed to the next phase $l+1$, which are listed in the following conditions.

- **Condition 1**: In Line 7, if the current phase l is greater than the maximum phase L, we directly stop at that phase and take the greedy policy on previous phase $\pi_h^k(s) = \pi_{h,l-1}^k(s)$.
- **Condition 2**: In Line 9, if there exists an action whose uncertainty $\|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}_{h,l}^{k,-1}}$ is greater than the threshold $2^{-l}\gamma_l^{-1}$, our algorithm will perform exploration by selecting that action.
- **Condition 3**: In Line 11, we compare the value of the pessimistic value function $\check{V}_{h,l}^k(s)$ and the optimistic value function $\widehat{V}_{h,l}^k(s)$ which will be assigned in Line 14 and Line 15, if the pessimistic estimation will be greater than the optimistic estimation, we will stop at that phase and take the greedy policy on previous phase $\pi_h^k(s) = \pi_{h,l-1}^k(s)$. Only in this case, the Algorithm 13 outputs flag $f_h^k(s) = 0$, which means this observation will not be used in Line 15 in Algorithm 12.
- **Condition 4**: In the default case in Line 16, the algorithm proceeds to the next phase after eliminating actions.

Notably, in **Condition 4**, since the expected estimation precision in the l-th phase is about $\widetilde{\mathcal{O}}(2^{-l})$, our algorithm can eliminate the actions whose state-action value is significantly

less than others, i.e., less than $\widetilde{\mathcal{O}}(2^{-l})$, while retaining the remaining actions for the next phase.

Specially, our algorithm differs from that in Vial et al. (2022) in terms of **Condition 3** to certify the performance of the estimation. In particular, a well-behaved estimation should always guarantee that the optimistic estimation is greater than the pessimistic estimation. According to Line 14 and Line 15, this is equivalent to the confidence region for l-th phase has intersection of the previous confidence region $[\check{V}^k_{h,l-1}(s), \widehat{V}^k_{h,l-1}(s)]$. Otherwise, we hypothesis the estimation on l-th phase is corrupted by either misspecification or bad concentration event, thus will stop the algorithm. We will revisit the detail of this design later.

It's important to highlight that our algorithms provide unique approaches when compared with previous works. In particular, He et al. (2021b) does not eliminate actions and combines estimations from all layers by considering the minimum estimated optimistic value function. This characteristic prevents their algorithm from achieving a uniform PAC guarantee in the presence of misspecification. For a more detailed comparison with He et al. (2021b), please refer to Section 5.8.1. Additionally, Lykouris et al. (2021); Wei et al. (2022) focus on a model-selection regime where a set of base learners are employed in the algorithms, whereas we adopt a multi-phase approach similar with SupLinUCB rather than conducting model selection over base learners.

5.5 Constant Regret Guarantee

We present the regret analysis in this section.

Theorem 5.5.1. Under Assumption 5.3.1, let $\gamma_l = 5(l + 20 + \lceil \log(ld) \rceil) dH\sqrt{\log(16ldH/\delta)}$ for some fixed $0 < \delta < 1/4$. With probability at least $1 - 4\delta$, if the minimal suboptimality gap Δ satisfies $\Delta > \widetilde{\Omega}\big(\sqrt{d}H^2\zeta\big)$, then for all $K \in \mathbb{N}^+$, the regret of Algorithm 12 is upper

bounded by

$$\text{Regret}(K) \leqslant \widetilde{\mathcal{O}}\big(d^3H^5\Delta^{-1}\log(1/\delta)\big).$$

This regret bound is constant w.r.t. the episode K.

Theorem 5.5.1 demonstrates a constant regret bound with respect to number of episodes K. Compared with Papini et al. (2021a), our regret bound does not require any prior assumption on the feature mapping $\boldsymbol{\phi}$, such as the *UniSOFT* assumption made in Papini et al. (2021a). In addition, compared with the previous logarithmic regret bound He et al. (2021a) in the well-specified setting, our constant regret bound removes the $\log K$ factor, indicating the cumulative regret no longer grows w.r.t. the number of episode K, with high probability.

Remark 5.5.2. As discussed in Zhang et al. (2023c) in the misspecified linear bandits, Our *high probability* constant regret bound does not violate the lower bound proved in Papini et al. (2021a), which says that certain diversity condition on the contexts is necessary to achieve an *expected* constant regret bound. When extending this high probability constant regret bound to the expected regret bound, we have

$$\mathbb{E}[\text{Regret}(K)] \leqslant \widetilde{\mathcal{O}}\big(d^3H^5\Delta^{-1}\log(1/\delta)\big)\cdot(1-\delta)+\delta K,$$

which depends on the number of episodes k. To obtain a sub-linear expected regret, we can choose $\delta = 1/K$, which yields a logarithmic expected regret $\widetilde{\mathcal{O}}(d^3H^5\Delta^{-1}\log K)$ and does not violate the lower bound in Papini et al. (2021a).

Remark 5.5.3. Du et al. (2019) provide a lower bound showing the interplay between the misspecification level ζ and suboptimality gap Δ in a weaker setting, which we discuss in detail in Section 5.8.2. Along with the result from Du et al. (2019), our results suggests that ignoring the dependence on H, $\zeta = \widetilde{\mathcal{O}}(\Delta/\sqrt{d})$ plays an important seperation for if a misspeficied model can be efficiently learned. This result is also aligned with the positive result and negative result for linear bandits (Lattimore et al., 2020; Zhang et al., 2023c).

5.6 Highlight of Proof Techniques

In this section, we highlight several major challenges in obtaining the constant regret under misspecified linear MDP assumption and how our method, especially the certified estimator, tackles these challenges.

5.6.1 Technical challenges

Challenge 1. Achieving layer-wise local estimation error. In the analysis of the value function under misspecified linear MDPs, we need to follow the multi-phase estimation strategy (Vial et al., 2022) to eliminate suboptimal actions and improve the robustness of the next phase estimation. Similar approaches have been observed in Zhang et al. (2023c); Chu et al. (2011) within the framework of (misspecified) linear bandits. However, unlike linear bandits, when constructing the empirical value function $\widehat{V}_h$ for stage h in linear MDPs, Jin et al. (2020b) requires a covering statement on value functions to ensure the convergence of the regression, which is written by: (see Lemma D.4 in Jin et al. (2020b) for details)

$$\left\|\sum_{\tau\in\mathcal{C}}\boldsymbol{\phi}_h^\tau\big[\widehat{V}_{h+1}^k(s^\tau)-\mathbb{E}[\widehat{V}_{h+1}^k(s^\tau)]\big]\right\|_{\mathbf{U}_h^{-1}}\leqslant\widetilde{\mathcal{O}}_H\Big(\sqrt{d\log(|\mathcal{C}|)+\log(|\mathcal{V}_{h+1}^k|/\delta)}+\sqrt{d}\kappa\Big),\quad(5.6.1)$$

where we employ notation $\widetilde{\mathcal{O}}_H$ to obscure the dependence on H to simplify the presentation. We use the notation $\mathcal{V}_{h+1}^k$ to denote as an κ-covering for the value functions $\widehat{V}_{h+1}^k$. Takemura et al. (2021); Vial et al. (2022) used quantification instead of the covering number, but this approach still encounters the issue of taking the union bound across the set of value functions, thereby incorporating the dependency on the cardinality of this set.

In the multi-phase algorithm, the regression employs a distinct set $\mathcal{C}=\mathcal{C}_{h,l}^k$ for each phase l. However, all these regressions use the overall empirical value function $\widehat{V}_{h+1}^k$ from the subsequent stage $h+1$, which is formulated using all pairs of parameters $\big\{\mathbf{w}_{h,\ell}^k,\mathbf{U}_{h,\ell}^k\big\}_\ell$ in L phases. Consequently, the covering number $\log|\mathcal{V}_{h+1}^k|$ is directly proportional to the number of phases $L=\mathcal{O}(\log K)$.

Therefore, when analyzing any single phase l, prior analysis cannot eliminate the $\log K$ term from (5.6.1) to achieve a *local* estimation error independent that is independent of the logarithmic number of global episodes $\log K$. Furthermore, due to the algorithm design of previous methods (Vial et al., 2022), additional $\log K$ terms may be introduced, induced by global quantification (i.e., $\varepsilon_{tol} = d/\sqrt{K}$).

Challenge 2. Achieving constant regret from local estimation error In misspecified linear bandits, Zhang et al. (2023c) concludes their proof by controlling $\sum_{k=1}^{\infty} \mathbb{1}[V_1^*(s_1^k) - V_1^\pi(s_1^k) \geqslant \Delta]$[2]. Although it is trivial showing that rounds with instantaneous regret $V_1^*(s_1^k) - V_1^\pi(s_1^k) < \Delta$ is optimal in bandits (i.e., $V_1^*(s_1^k) = V_1^\pi(s_1^k)$), previous works fail to reach a similar result for RL settings. This difficulty arises from the randomness inherent in MDPs: Consider a policy π that is optimal at the initial stage $h = 1$. After the initial state and action, the MDP may transition to a state s_2' with a small probability p where the policy π is no longer optimal, or to another state s_2 where π remains optimal until the end. In this context, the gap between $V_1^*(s_1)$ and $V_1^\pi(s_1)$ can be arbitrarily small, given a sufficiently small $p > 0$:

$$V_1^*(s_1) - V_1^\pi(s_1) = p\big(V_2^*(s_2') - V_2^\pi(s_2')\big) + (1-p)\big(V_2^*(s_2) - V_2^\pi(s_2)\big) = p\big(V_2^*(s_2') - V_2^\pi(s_2')\big).$$

Therefore, one cannot easily draw a constant regret conclusion simply by controlling the summation $\sum_{k=1}^{\infty} \mathbb{1}[V_1^*(s_1^k) - V_1^\pi(s_1^k) \geqslant \Delta]$ since the gap between $V_1^*(s_1^k) - V_1^\pi(s_1^k)$ needs to be further fine-grained controlled. In short, the existence of Δ describing the minimal gap between $V^*(s) - Q^*(s,a)$ cannot be easily applied to controlling regret $V^*(s) - V^\pi(s)$.

5.6.2 A novel approach: Cert-LinUCB

We introduce the technical details in designing our new subroutine, Cert-LinUCB, to tackle **Challenge 1**. In the addition of using 'local quantification' which ensure the quantification

[2]We employ the RL notations and set $h = 1$ for the ease of comparison.

error of each phase l depend on the local phase $\widetilde{\mathcal{O}}(l)$ instead of the global parameter $\log K$, Algorithm 13 eliminates the $\log K$ dependence from $\log|\mathcal{V}|$ by employing certified estimator.

Considering the concentration term we need to control for each phase l:

$$\left\|\sum_{\tau\in\mathcal{C}_{h,l}^k}\boldsymbol{\phi}_h^\tau\big[\widehat{V}_{h+1}^k(s^\tau)-\mathbb{E}[\widehat{V}_{h+1}^k(s^\tau)]\big]\right\|_{(\mathbf{U}_{h,l}^k)^{-1}},$$

as discussed in **Challenge 1**, the function class $\widehat{\mathcal{V}}_{h+1}^k \ni \widehat{V}_{h+1}^k$ involves $L=\mathcal{O}(\log K)$ parameters, leading to a $\log K$ dependence in the results when using traditional routines. The idea of certified estimator is to get rid of this by not directly controlling $\log|\mathcal{V}_{h+1}^k|$. Instead, certified estimator establishes a covering statement for the value function class $\mathcal{V}_{h+1,l_+}^k \ni \widehat{V}_{h+1,l_+}^k$, where $\widehat{V}_{h+1,l_+}^k$ is the value function that only incorporates the first l_+ phases of parameters $\big\{\mathbf{w}_{h,\ell}^k,\mathbf{U}_{h,\ell}^k\big\}_\ell$. Under this framework, the covering statement becomes:

Lemma 5.6.1 (Lemma 5.9.4, informal)**.** Let $\widehat{V}_{h+1,l_+}^k$ be the output of Algorithm 13 terminated at phase $L=l_+$, then with probability is at least $1-2\delta$,

$$\left\|\sum_{\tau\in\mathcal{C}_{h,l}^k}\boldsymbol{\phi}_h^\tau\big[\widehat{V}_{h+1,l_+}^k(s^\tau)-\mathbb{E}[\widehat{V}_{h+1,l_+}^k(s^\tau)]\big]\right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant \gamma_{l,l_+}=5l_+dH\sqrt{\log(16ldH/\delta)}.$$

To apply Lemma 5.6.1, it is essential to bound the distance between $\widehat{V}_{h+1,l_+}^k$ and $\widehat{V}_{h+1}^k$. For this purpose, we maintain a monotonic sequence of the optimistic value function $\widehat{V}_{h+1,l}^k$ and the pessimistic value function $\check{V}_{h+1,l}^k$, ensuring that

$$\begin{aligned}&\check{V}_{h+1,1}^k(s)\leqslant\check{V}_{h+1,2}^k(s)\leqslant\cdots\leqslant\check{V}_{h+1,l_h^k(s)-1}^k(s)\\&\leqslant\widehat{V}_{h+1}^k(s)=\widehat{V}_{h+1,l_h^k(s)-1}^k(s)\leqslant\cdots\leqslant\widehat{V}_{h+1,2}^k(s)\leqslant\widehat{V}_{h+1,1}^k(s).\end{aligned}\tag{5.6.2}$$

This monotonicity is guaranteed according to Line 11 in Cert-LinUCB, where the process is terminated once (5.6.2) is violated. As a result, we can control the distance between $\widehat{V}_{h+1,l_+}^k$ and $\widehat{V}_{h+1}^k$ as the following lemma.

Lemma 5.6.2 (Lemma 5.9.2, informal)**.** There is a faithful extension of $\widehat{V}_{h+1,l_+}^k$ to every $l_+\in\mathbb{N}^+$ that $|\widehat{V}_h^k(s)-\widehat{V}_{h,l_+}^k(s)|\leqslant 6\cdot 2^{-l_+}$ always holds.

Following these results, combining Lemma 5.6.2 and Lemma 5.6.1 together obtains a local concentration bound for each phase l that is independent of $\log K$ when choosing $l_+ = l + \widetilde{\mathcal{O}}(\log(ld))$.

Lemma 5.6.3 (Lemma 5.9.5, informal)**.** With probability at least $1-2\delta$, for any $(k,h,l) \in [K] \times [H] \times \mathbb{N}^+$:

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau) - \widehat{V}_{h+1}^k(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 1.1\gamma_l.$$

As a result of our improved concentration analysis, we can achieve a local estimation error for the estimated value function in each phase l:

Lemma 5.6.4 (Lemma 5.9.6, informal)**.** With high probability, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [l_h^k(s) - f_h^k(s)], a_l \in \mathcal{A}_{h,l}^k(s)$, we have $\left| Q_{h,l}^k(s,a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a) \right| \leqslant 2 \cdot 2^{-l} + \mathcal{O}(\sqrt{d} H \zeta)$.

Lemma 5.6.4 bounds the estimation error for any state s by the phase $l_h^k(s)$ where $l_h^k(s)$ indicates the layer at which Algorithm 13 terminates. As the early phases cannot provide sufficient accuracy due to a lack of data, we also need to analyze the conditions under which Line 11 is triggered in Algorithm 13.

Lemma 5.6.5 (Lemma 5.9.8, informal)**.** With probability at least $1-2\delta$, for any $(k,h) \in [K] \times [H]$, Line 11 in Algorithm 13 can only be triggered on phase $l \geqslant \widetilde{\Omega}\big(\log(1/\zeta)\big)$.

Lemma 5.6.5 delivers a clear message: the trigger of Line 11 is related to the misspecification level ζ. In the well-specified setting, Line 11 will never be triggered ($l \geqslant \infty$). When the misspecification level is large, then Line 11 will be more likely triggered, indicating it's harder to get higher precise estimation via higher $l_h^k(s)$, according to Lemma 5.6.4.

In order to bound the number of suboptimality gap taken by Cert-LSVI-UCB, we start with the following standard decomposition

$$\begin{aligned} V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) &= \big(V_h^*(s_h^k) - \widehat{V}_h^k(s_h^k) \big) + \big(\widehat{V}_h^k(s_h^k) - V_h^{\pi^k}(s_h^k) \big) \\ &= \big(V_h^*(s_h^k) - \widehat{V}_h^k(s_h^k) \big) + \textstyle\sum_{h'=h}^H \Big(\widehat{V}_{h'}^k(s_{h'}^k) - [\mathbb{B}_h \widehat{V}_{h'+1}^k]\big(s_{h'}^k, \pi_{h'}^k(s_{h'}^k)\big) \Big) + \sum_{h'=h}^H \eta_{h'}^k. \end{aligned}$$

where $\eta_h^k = [\mathbb{P}_h(\widehat{V}_{h+1}^k - V_{h+1}^{\pi^k})]\big(s_h^k, \pi_h^k(s_h^k)\big) - \big(\widehat{V}_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi^k}(s_{h+1}^k)\big)$ is a zero-mean random variable induced by the transition kernel. We can bound each of the factors using standard regret analysis on the basis created by combining Lemma 5.6.4 and Lemma 5.6.5:

1. Global underestimation error $V_h^*(s_h^k) - \widehat{V}_h^k(s_h^k)$. (see Lemma 5.9.25)
2. Local overestimation error $\widehat{V}_h^k(s_h^k) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s_{h'}^k, \pi_h^k\big(s_h^k\big))$. (see Lemma 5.9.26)
3. Transition noise η_h^k. (see Lemma 5.9.28)

In general, we can reach the following results, which provide an local regret upper bound for arbitrary index subsets which is independent from the number of total episodes.

Lemma 5.6.6 (Lemma 5.9.29, Informal)**.** With high probability, for any index set $\mathcal{K}$ and any ε that is comparably large against ζ, it satisfies that

$$\textstyle\sum_{k\in\mathcal{K}}\big(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\big) \leqslant 0.49|\mathcal{K}|\varepsilon + \widetilde{\mathcal{O}}\big(d^3H^4\varepsilon^{-1} + \sqrt{H^3|\mathcal{K}|}\big).$$

Note that for index set $\mathcal{K} = \big[k : V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \geqslant \varepsilon\big]$, the regret enjoys a trivial lower bound that $\sum_{k\in\mathcal{K}}\big(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\big) \geqslant |\mathcal{K}|\varepsilon$. We thus can reach the following result.

Lemma 5.6.7 (Lemma 5.9.12, Informal)**.** With high probability, for any $\varepsilon > \widetilde{\Omega}(\sqrt{d}H^2\zeta)$ and $h \in [H]$, Cert-LSVI-UCB ensures $\sum_{k=1}^{\infty} \mathbb{1}\big[V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \geqslant \varepsilon\big] \leqslant \widetilde{\mathcal{O}}\big(d^3H^4\varepsilon^{-2}\big)$.

Remark 5.6.8. He et al. (2021b) achieved a *uniform-PAC* bound for (well-specified) linear MDP, which states as

$$w.h.p., \forall\epsilon > 0, \textstyle\sum_{k=1}^{\infty} \mathbb{1}[V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k) \geqslant \epsilon] \leqslant \widetilde{\mathcal{O}}(d^3H^5\epsilon^{-2}), \tag{5.6.3}$$

comparing (5.6.3) with Lemma 5.6.7 on well-specified setting where $\zeta = 0$, one can find that our result is better than He et al. (2021b) under stronger condition: Lemma 5.6.7 ensures this *uniform-PAC* result under all stage $h \in [H]$ while He et al. (2021b) only ensure the initial statement. Due to the randomness of MDP discussed in **Challenge 2**, the guarantee for

$h = 1$ cannot be easily generated to any $h \in [H]$. Second, our result $\widetilde{\mathcal{O}}(d^3H^4\epsilon^{-2})$ improves He et al. (2021b) by a factor H. This is because a more efficient data selection strategy which we will discuss in detail in Section 5.8.1.

5.6.3 Settling the gap between $V^* - V^\pi$ and $V^* - Q^*$

According to **Challenge 2**, the regret in episodes where $V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k) \leqslant \Delta$ is non-zero since the minimal suboptimality gap assumption Δ only guarantees $\Delta_h^k = V_h^*(s_h^k) - Q_h^*\big(s_h^k, \pi_h^k(s_h^k)\big) \notin (0, \Delta)$ but put no restrictions on $V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)$.

Notice that the regret $V_h^*(s_h) - V_h^{\pi^k}(s_h)$ in episode k is the expectation of cumulative suboptimality gap $\sum_{h=1}^H \Delta_h^k$ taking over trajectory $\{s_h^k\}_{h=1}^H$. In addition, the variance of the random variable can be self-bounded according to

$$\mathrm{Var}\Big[\sum_{h=1}^H \Delta_h^k\Big] \leqslant \mathbb{E}\Big[\Big(\sum_{h=1}^H \Delta_h^k\Big)^2\Big] \leqslant H^2\mathbb{E}\Big[\sum_{h=1}^H \Delta_h^k\Big] = H^2\big(V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k)\big).$$

As Freedman inequality (Lemma 5.9.30) implies $\sum_{t=1}^T \mathrm{Var}[\eta^t] \leqslant aC$ and $\sum_{t=1}^T \mathrm{Var}[\eta^t] \leqslant vC$ only happens with small probability for every C with proper constant a and v, together with a union bound statement C, we can reach the following statement indicates the cumulative regret can be upper bounded using the cumulative suboptimality gap:

Lemma 5.6.9 (Lemma 5.9.14, Informal)**.** The following statement holds with high probability:

$$\sum_{k=1}^K \big(V_h^*(s_h) - V_h^{\pi^k}(s_h)\big) \leqslant \widetilde{\mathcal{O}}\Big(\sum_{k=1}^K \sum_{h=1}^H \Delta_h^k + H^2\Big).$$

In addition, since the loss on the state-value function $V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)$ is a upper bound for the sub-optimality gap Δ_h^k, we are able to show that the cumulative suboptimality gap is constantly bounded when the minimal suboptimality gap is sufficiently large as in Lemma 5.9.13:

$$\begin{aligned}
\sum_{k=1}^K \sum_{h=1}^H \Delta_h^k &= \sum_{k=1}^K \sum_{h=1}^H \Big(\Delta \cdot \mathbb{1}\Big[\Delta_h^k \geqslant \Delta\Big] + \int_\Delta^H \mathbb{1}\Big[\Delta_h^k \geqslant \varepsilon\Big] \mathrm{d}\varepsilon\Big) \\
&\leqslant \sum_{k=1}^K \sum_{h=1}^H \Big(\Delta \cdot \mathbb{1}\Big[V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \geqslant \Delta\Big] + \int_\Delta^H \mathbb{1}\Big[V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \geqslant \varepsilon\Big] \mathrm{d}\varepsilon\Big).
\end{aligned}$$

Note that the final summation can be upper bounded by $\widetilde{\mathcal{O}}\big(d^3H^5\Delta^{-2}\big)$ using Lemma 5.6.7. Together with Lemma 5.6.9, we reach the desired statement that Cert-LSVI-UCB achieves a constant regret bound when the misspecification is sufficiently small against the minimal suboptimality gap.

5.7 Conclusion

In this chapter, we proposed a new algorithm, called certified estimator, for reinforcement learning with a misspecified linear function approximation. Our algorithm is parameter-free and does not require prior knowledge of misspecification level ζ or the suboptimality Δ. Our algorithm is based on a novel certified estimator and provides the first constant regret guarantee for misspecified linear MDPs and (well-specified) linear MDPs. There are still some future works resulting from current algorithm. First, it is still an open question that whether the dependence on the planning horizon and dimension d, H is optimal for instance-dependent regret bound, as in the well-specified case, the regret lower bound is $\Omega(d\sqrt{H^3K})$ (Zhou et al., 2021b), which has been recently attained by He et al. (2022a); Agarwal et al. (2022). Second, our work propose a new open question that if it possible to fill the gap between the positive result in our work and the negative result in Du et al. (2019). We believe it's important in both theory and practice to understand how much misspecification level can be tolerated to efficiently learn the algorithm.

5.8 Additional Discussions

5.8.1 Comparison with He et al. (2021b)

It is worth comparing our algorithm with He et al. (2021b), which also provides a uniform PAC bound for linear MDPs. Both our algorithm and theirs utilize a multi-phase structure that maintains multiple regression-based value function estimators at different phases. De-

spite this similarity, there are several major differences between our algorithm and that in He et al. (2021b), which are highlighted as follows:

1. In Line 7 of Algorithm 12, when calculating the regression-based estimator, for different phase l, we use the same regression target $\widehat{V}_{h+1}^k$, while their algorithm uses different $V_{h+1,l}^k$ for different phase l.

2. When aggregating the regression estimators over all different L_k phases, we follow the arm elimination method as in Chu et al. (2011), while He et al. (2021b) simply take the point-wise minimum of all estimated state-action functions, i.e., $Q(s,a) = \min_{l\in[L_k]} Q_{k,h}^l(s,a)$.

3. When calculating the phase $l_h^k(s_h^k)$ for a trajectory $s_1^k, s_2^k, \cdots, s_H^k$, He et al. (2021b) require that the phase $l_h^k(s_h^k)$ to be monotonically decreasing with respect to the stage h, i.e., $l_h^k(s_h^k) \leqslant l_{h-1}^k(s_{h-1}^k)$ (see line 19 in Algorithm 2 in He et al. (2021b)). Such a requirement will lead to a poor estimation for later stages and thus increase the sample complexity. In contrast, we do not have this requirement or any other requirements related to $l_h^k(s_h^k)$ and $l_{h-1}^k(s_{h-1}^k)$.

As a result, by 3, He et al. (2021b) have to sacrifice some sample complexity to make their algorithm work for different target value functions $V_{h+1,l}^k$. As a comparison, since we use the same regression target for different phase l, we do not have to make such a sacrifice in 3. Moreover, by 2, He et al. (2021b) cannot deal with linear MDPs with misspecification, while our algorithm can handle misspecification as in Vial et al. (2022).

5.8.2 Discussion on Lower Bounds of Sample Complexity

We present a lower bound from Du et al. (2019) to better illustrate the interplay between the misspecification level ζ and the suboptimality gap Δ.

Assumption 5.8.1 (Assumption 4.3, Du et al. 2019, ζ-Approximate Linear MDP)**.** There exists $\zeta > 0$, $\boldsymbol{\theta}_h \in \mathbb{R}^d$ and $\boldsymbol{\mu}_h : \mathcal{S} \mapsto \mathbb{R}^d$ for each stage $h \in [H]$ such that for any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we have $\left|\mathbb{P}_h(s'|s, a) - \langle \boldsymbol{\phi}(s, a), \boldsymbol{\mu}_h(s')\rangle\right| \leqslant \zeta$ and $\left|r(s, a) - \langle \boldsymbol{\phi}(s, a), \boldsymbol{\theta}_h\rangle\right| \leqslant \zeta$.

Theorem 5.8.2 (Theorem 4.2, Du et al. 2019)**.** There exists a family of hard-to-learn linear MDPs with action space $|\mathcal{A}| = 2$ and a feature mapping $\boldsymbol{\phi}(s, a)$ satisfying Assumption 5.8.1, such that for any algorithm that returns a 1/2-optimal policy with probability 0.9 needs to sample at least $\Omega(\min\{|\mathcal{S}|, 2^H, \exp(d\zeta^2/16)\})$ episodes.

Remark 5.8.3. As claimed in Du et al. (2019), Theorem 5.8.2 suggests that when misspecification in the ℓ_∞ norm satisfies $\zeta = \Omega(\Delta\sqrt{H/d})$, the agent needs an exponential number of episodes to find a near-optimal policy, where $\Delta = 1/2$ in their setting. It is worth noting that Assumption 5.8.1 is a ℓ_∞ approximation for the transition matrix. Such a ℓ_∞ guarantee ($\|\cdot\|_\infty \leqslant \zeta$) is weaker than the ℓ_1 guarantee ($\|\cdot\|_1 \leqslant \zeta$) provided in Assumption 5.3.1. So it's natural to observe a positive result when making a stronger assumption and a negative result when making a weaker assumption. In addition, despite of this difference, one could find that $\zeta \sim \Delta/\sqrt{d}$ plays a vital role in determining if the task can be efficiently learned. Similar positive and negative results are also provided in Lattimore et al. (2020); Zhang et al. (2023c) in the linear contextual bandit setting (a special case of linear MDP with $H = 1$).

5.9 Proofs

5.9.1 Constant Regret Guarantees for Cert-LSVI-UCB

In this section, we present the proof of Theorem 5.5.1. To begin with, we recap the notations used in the algorithm and introduce several shorthand notations that would be employed for the simplicity of latter proof. The notation table is presented in Table 5.2.Any proofs not included in this section are deferred to Section 5.9.2.

Notation	Meaning
ζ	Misspecification level of feature map ϕ_h. (see Definition 5.3.1)
Δ	Minimal suboptimality gap among Δ_h. (see Definition 5.3.3)
s_h^k, a_h^k	States and actions introduced in the episode k by the policy π_k.
$Q_h^\pi(s,a), V_h^\pi(s)$	Ground-truth state-action value function and state value function of policy π.
$Q_h^*(s,a), V_h^*(s)$	The optimal ground-truth state-action value function and state value function.
$\Delta_h(s,a)$	Suboptimal gap with respect to the optimal policy π^*. (see Definition 5.3.3)
$\mathbb{P}_h, \mathbb{B}_h$	The ground-truth transition kernel and the Bellman operator.
κ_l	The quantification precision in the phase l. (see Algorithm 12)
γ_l	The confidence radius in the phase l. (see Theorem 5.5.1)
$\mathcal{C}_{h,l}^k$	Index sets during phase l in the episode k. (see Algorithm 12)
$\mathbf{w}_{h,l}^k, \mathbf{U}_{h,l}^k$	Empirical weights and covariance matrix in the phase l. (see Algorithm 12)
$\widetilde{\mathbf{w}}_{h,l}^k, \widetilde{\mathbf{U}}_{h,l}^k$	Quantified version of $\mathbf{w}_{h,l}^k$ and $\mathbf{U}_{h,l}^k$. (see Algorithm 12)
$\widehat{V}_h^k(s)$	The overall optimistic state value function. (see Algorithm 13)
$Q_{h,l}^k(s,a)$	Empirical state-action value function in phase l. (see Algorithm 13)
$V_{h,l}^k(s)$	Empirical state value function in phase l. (see Algorithm 13)
$\widehat{V}_{h,l}^k(s)$	Optimistic state value function in phase l. (see Definition 5.9.1)
$\check{V}_{h,l}^k(s)$	Pessimistic state value function in phase l. (see Algorithm 13)
π_h^k	Policy played in the episode k. (see Algorithm 13)
$\pi_{h,l}^k$	Policy induced at state s during phase l of episode k. (see Algorithm 13)
$l_h^k(s)$	The index of the phase at which state s stops in episode k. (see Algorithm 13)
ϕ_h^k	The feature vector observed in the episode k. (see Algorithm 12)
$\mathcal{V}_{h,l}^k$	Function family of all optimistic state function $\widehat{V}_{h,l}^k$. (see Definition 5.9.1)
γ_{l,l_+}	The confidence radius with covering on phase l_+. (see Definition 5.9.3)
l_+	The phase offsets for the covering statement. (see Lemma 5.9.5)
χ	The inflation on misspecification. (see Lemma 5.9.6)
L_ζ	The deepest phase that tolerance ζ misspecification. (see Lemma 5.9.8).
L_ε	The shallowest phase that guarantees ε accuracy. (see Lemma 5.9.9).
Δ_h^k	The sub-optimaility gap of played policy π_h^k at state s_h^k. (see Lemma 5.9.13)
$\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$	The event defined in Definition 5.9.3, Definition 5.9.27 and Definition 5.9.10.

Table 5.2: Notations used in algorithm and proof

5.9.1.1 Quantized State Value Function set $\mathcal{V}_{h,l}^k$

To begin our proof, we first extend the definition of $\widehat{V}_{h,l}^k$ to arbitrary l and give a formal definition of the state value function class $\mathcal{V}_{h,l}^k$ as we skip the detail of this definition in Section 5.6.

Definition 5.9.1. We extend the definition of state value function $\widehat{V}_{h,l}^k$ to any tuple $(k,h,l) \in [K] \times [H] \times \mathbb{N}^+$ by

$$\widehat{V}_{h,l}^k, \cdot, \cdot, \cdot = \text{Cert-LinUCB}\big(s; \{\widetilde{\mathbf{w}}_{h,\ell}^k\}_{\ell=1}^l, \{\widetilde{\mathbf{U}}_{h,\ell}^{k,-1}\}_{\ell=1}^l, l\big)$$

We also define the state value function family $\mathcal{V}_{h,l}^k$ be the set of all possible $\widehat{V}_{h,l}^k$.

$$\mathcal{V}_{h,l}^k = \left\{ \widehat{V}_{h,l}^k \;\middle|\; \widehat{V}_{h,l}^k, \cdot, \cdot, \cdot = \text{Cert-LinUCB}\big(s; \{\widetilde{\mathbf{w}}_{\cdot,\ell}^{\cdot}\}_{\ell=1}^l, \{\widetilde{\mathbf{U}}_{\cdot,\ell}^{\cdot,-1}\}_{\ell=1}^l, l\big) \right\}$$

where $\{\widetilde{\mathbf{w}}_{\cdot,\ell}^{\cdot}\}_{\ell=1}^l$ and $\{\widetilde{\mathbf{U}}_{\cdot,\ell}^{\cdot,-1}\}_{\ell=1}^l$ are referring to *any* possible parameters generated by Line 8 in Algorithm 12.

It is worth noting that one can check the definition of $\widehat{V}_{h,l}^k$ here is consistent with those computed in Algorithm 13 with $l < l_h^k(s)$. Therefore, we will not distinguish between the notations in the remainder of the proof.

The following lemma controls the distance between $\widehat{V}_h^k(s)$ and $\widehat{V}_{h,l}^k(s)$ for any phase l.

Lemma 5.9.2. For any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [l_h^k(s) - 1]$, it holds that

$$\check{V}_{h,l}^k(s) \leqslant \widehat{V}_h^k(s) \leqslant \widehat{V}_{h,l}^k(s), \quad |\widehat{V}_h^k(s) - \widehat{V}_{h,l}^k(s)| \leqslant 6 \cdot 2^{-l}.$$

Moreover, for any tuple $(k,h,s,l_+) \in [K] \times [H] \times \mathcal{S} \times \mathbb{N}^+$, the difference $|\widehat{V}_h^k(s) - \widehat{V}_{h,l_+}^k(s)|$ is bounded by $6 \cdot 2^{-l_+}$, following the extension of the definition scope of $\widehat{V}_{h,l_+}^k$ as outlined in Definition 5.9.1.

Lemma 5.9.2 suggests that given any phase l_+, $\widehat{V}_{h,l}^k$ is close to $\widehat{V}_h^k$. This enables us to construct covering on $\widehat{V}_h^k$ using the covering on $\widehat{V}_{h,l}^k$.

5.9.1.2 Concentration of State Value Function $\widehat{V}_h^k(s)$

In this subsection, we provide a new analysis for bounding the self-normalized concentration of $\left\| \sum_\tau \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h \widehat{V}_h^k](s_h^\tau, a_h^\tau) - \widehat{V}_h^k(s_{h+1}^\tau) \big) \right\|_{\mathbf{U}^{-1}}$ to get rid of the $\log k$ factor in Vial et al. (2022).

To facilitate our proof, we define the filtration list $\mathcal{F}_h^k = \Big\{ \big\{ s_i^j, a_i^j \big\}_{i=1,j=1}^{H,k-1}, \big\{ s_i^k, a_i^k \big\}_{i=1}^{h} \Big\}$. It is easy to verify that s_h^k, a_h^k are both $\mathcal{F}_h^k$-measurable. Also, for any function V built on $\mathcal{F}_h^k$, $[\mathbb{P}_h V](s_h^k, a_h^k) - V(s_{h+1}^k)$ is $\mathcal{F}_{h+1}^k$-measurable and it is also a zero-mean random variable conditioned on $\mathcal{F}_h^k$.

The first lemma we provide is similar with Vial et al. (2022), which shows the self-normalized concentration property for each phase l and any function $V \in \mathcal{V}_{h,l}^k$.

Definition 5.9.3. For some fixed mapping $l \mapsto l_+ = l_+(l)$ that $l_+ \geqslant l$, we define the bad event as

$$\mathcal{B}_1(k,h,l,V) = \left\{ \left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h V](s_h^\tau, a_h^\tau) - V(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} > \gamma_{l,l_+} \right\}.$$

The good event is defined by $\mathcal{G}_1 = \bigcap_{k=1}^K \bigcap_{h=1}^H \bigcap_{l \geqslant 1} \bigcap_{V \in \mathcal{V}_{h,l_+}^k} \mathcal{B}_1^\complement(k,h,l,V)$ where we define $\gamma_{l,l_+} = 5 l_+ dH \sqrt{\log(16 ldH/\delta)} = \widetilde{\mathcal{O}}(ldH \log(\delta^{-1}))$.

Lemma 5.9.4. The good event $\mathcal{G}_1$ defined in Definition 5.9.3 happens with probability at least $1 - 2\delta$.

Lemma 5.9.4 establishes the concentration bounds for any given phase l. However, the total number of phases for the state value function $V_h^k(s)$ can be bounded only trivially by$l = \mathcal{O}(\log K)$, resulting in $\log K$ dependence. To address this issue, the following lemma, as we sketched in Section 5.6.2, proposes a method to eliminate this logarithmic factor:

Lemma 5.9.5. Under event $\mathcal{G}_1$, for any $(k,h,l) \in [K] \times [H] \times \mathbb{N}^+$,

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau) - \widehat{V}_{h+1}^k(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 1.1 \gamma_l. \tag{5.9.1}$$

where we set $\gamma_l = \gamma_{l,l_+}$ with $l_+ = l + 20 + \lceil \log(ld) \rceil$.

Then Lemma 5.9.5 immediately yields the following lemma regarding the estimation error of the state-action value function $Q^k_{h,l}$:

Lemma 5.9.6. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [l^k_h(s) - f^k_h(s)], a_l \in \mathcal{A}^k_{h,l}(s)$,

$$\left|Q^k_{h,l}(s,a) - [\mathbb{B}_h \widehat{V}^k_{h+1}](s,a)\right| \leqslant 2 \cdot 2^{-l} + \chi\sqrt{l}\zeta \tag{5.9.2}$$

where we define $\chi = 12\sqrt{d}H$.

Lemma 5.9.6 build an estimation error for any $l \in [l^k_h(s) - 1]$. As we mentioned in the algorithm design, a larger l here will lead to more precise estimation (a smaller 2^{-l} term in (5.9.2)) but will suffer from a larger covering number (a larger γ_l term in (5.9.2)). Following a similar proof sketch from Vial et al. (2022), the next lemma shows that any action that is not eliminated has a low regret,

Lemma 5.9.7. Fix some arbitrary $L_0 \geqslant 1$ and let $\chi = 12\sqrt{d}H$. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [\min\{L_0, l^k_h(s) - f^k_h(s)\}], a_{l+1} \in \mathcal{A}^k_{h,l+1}(s)$,

$$\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}^k_{h+1}](s,a) - [\mathbb{B}_h \widehat{V}^k_{h+1}](s,a_{l+1}) \leqslant 8 \cdot 2^{-l} + 2l \cdot \chi\sqrt{L_0}\zeta.$$

5.9.1.3 The Impact of Misspecification Level ζ

Next, we are ready to show the criteria where Line 11 in Algorithm 13 will be triggered, which shows the impact of misspecification on this multi-phased estimation.

Lemma 5.9.8. Under event $\mathcal{G}_1$, for any $(k,h) \in [K] \times [H]$ such that $f^k_h(s^k_h) = 0$, we have $l^k_h(s^k_h) > L_\zeta$ where L_ζ is the maximal integer satisfying $2^{-L_\zeta} \geqslant \chi L_\zeta^{1.5}\zeta$ for $\chi = 12\sqrt{d}H$, i.e., $L_\zeta = \Omega(\log(1/\zeta))$.

Equipped with Lemma 5.9.8, the following lemma suggests that how much estimation precision ε can be achieved by accumulating the error $2^{-l^k_h(s^k_h)}$ that occurred in Lemma 5.9.6.

Lemma 5.9.9. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$, define L_ε to be the minimal integer satisfying $2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$, i.e., $L_\varepsilon = \lceil -\log(0.01\varepsilon/H) \rceil$. When $L_\varepsilon \leqslant L_\zeta$, then for any $\mathcal{K} \subseteq [K], h \in [H]$,

$$\sum_{k \in \mathcal{K}} 2^{-l_h^k(s_h^k)} \leqslant 0.01|\mathcal{K}| \cdot \varepsilon/H + 2^{12} L_\varepsilon dH \gamma_{L_\varepsilon}^2 \cdot \varepsilon^{-1}.$$

The relationship between $L_\varepsilon \leqslant L_\zeta$ can be translated to the relationship between ε and ζ. We characterize this condition as follows:

Definition 5.9.10. Condition $\mathcal{G}_\varepsilon$ is defined for a given ε, and is satisfied if $L_\zeta \geqslant L_\varepsilon$ where L_ε is the minimal integer satisfying $2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$ and L_ζ is the maximal integer satisfying $2^{-L_\zeta} \geqslant \chi L_\zeta^{1.5} \zeta$.

Lemma 5.9.11. If $\varepsilon \geqslant \Omega\big(\sqrt{d} H^2 \zeta \log^2(1/\zeta)\big)$, then $\mathcal{G}_\varepsilon$ is satisfied.

Proof. If $\varepsilon \geqslant \Omega\big(\sqrt{d} H^2 \zeta \log^2(1/\zeta)\big)$, we have

$$2^{-L_\varepsilon} \geqslant 0.005\varepsilon/H \geqslant 2\chi L_\zeta^{1.5} \zeta \geqslant 2^{-L_\zeta}.$$

where the first inequality is given by the definition of L_ε, the last inequality is given by the definition of L_ζ, and the second inequality holds since $H\chi L_\zeta^{1.5} \leqslant \mathcal{O}\big(\sqrt{d} H^2 \log^2(1/\zeta)\big)$, and the last inequality is given by the definition of L_ε and L_ζ, respectively. Since 2^{-l} decreases as l increases, we can conclude that $L_\varepsilon \leqslant L_\zeta$. □

The above analysis of the interplay between misspecification level ζ and precision ε yields the following important lemma in our proof, showing a local decision error across all $h \in [H]$:

Lemma 5.9.12. Under Assumption 5.3.1, let $\gamma_l = 5(l + 20 + \lceil \log(ld) \rceil) dH \sqrt{\log(16ldH/\delta)}$, for some fixed $0 < \delta < 1/3$. With probability at least $1 - 3\delta$, for any $\varepsilon > \Omega\big(\sqrt{d} H^2 \zeta \log^2(1/\zeta)\big)$ and $h \in [H]$, we have

$$\sum_{k=1}^{\infty} \mathbb{1}\left[V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \geqslant \varepsilon \right] \leqslant \mathcal{O}\big(d^3 H^4 \varepsilon^{-2} \log^4(dH\varepsilon^{-1}) \log(\delta^{-1}) \iota\big),$$

where ι refers to some polynomial of $\log\log(dH\varepsilon^{-1}\delta^{-1})$. This can also be written as

$$\Pr\left[\exists \varepsilon > \varepsilon_0, h \in [H], \sum_{k=1}^{\infty} \mathbb{1}\left[V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) > \varepsilon\right] > f(\varepsilon, \delta)\right] \leqslant \delta.$$

with $\varepsilon_0 = \widetilde{\Omega}(\sqrt{d}H^2\zeta)$ and $f(\varepsilon, \delta) = \widetilde{\mathcal{O}}(d^3H^4\varepsilon^{-2}\log(\delta^{-1}))$.

5.9.1.4 From Local Step-wise Decision Error to Constant Regret

The next lemma shows that the total incurred suboptimality gap is constant if the minimal suboptimality gap Δ satisfies $\Delta > \varepsilon_0$.

Lemma 5.9.13. Suppose an RL algorithm `Alg.` satisfies

$$\Pr\left[\exists \varepsilon > \varepsilon_0, h \in [H], \sum_{k=1}^{\infty} \mathbb{1}\left[V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) > \varepsilon\right] > f(\varepsilon, \delta)\right] \leqslant \delta,$$

such that $f(\varepsilon, \delta) = \widetilde{\mathcal{O}}(C_1/\varepsilon + C_2/\varepsilon^2)$ where $C_1, C_2 > 0$ are constant in ε, but may depend on other quantities such as $d, H, \log(\delta^{-1})$. If the minimal suboptimality gap Δ satisfies $\Delta > \varepsilon_0$, then

$$\sum_{k=1}^{K}\sum_{h=1}^{H} \Delta_h^k \leqslant \widetilde{\mathcal{O}}(C_2H/\Delta + C_1H)$$

where $\Delta_h^k = \Delta_h\big(s_h^k, \pi_h^k(s_h^k)\big) = V_h^*(s_h^k) - Q_h^*\big(s_h^k, \pi_h^k(s_h^k)\big)$ is the suboptimality gap suffered in stage h of episode k.

The following Lemma is a refined version of Lemma 6.1 in He et al. (2021a) that removes the dependence between regret and number of episodes K.

Lemma 5.9.14. For each MDP $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, \{r_h\}, \{\mathbb{P}_h\})$ and any $\delta > 0$, with probability at least $1 - \delta$, we have

$$\mathrm{Regret}(K) < \widetilde{\mathcal{O}}\bigg(\sum_{k=1}^{K}\sum_{h=1}^{H} \Delta_h^k + H^2\log(1/\delta)\bigg).$$

We are now ready to prove Theorem 5.5.1:

Proof of Theorem 5.5.1. By plugging in Lemma 5.9.12 and Lemma 5.9.13 into Lemma 5.9.14, we can reach the desired statement. □

5.9.2 Proof of Lemmas in Section 5.9.1

In this section, we prove lemmas outlined in Section 5.9.1. Any proofs not included in this section are deferred to Section 5.9.3.

5.9.2.1 Proof of Lemma 5.9.2

Proof of Lemma 5.9.2. According to the criteria for Line 11, we have $\check{V}^k_{h,l}(s) \leqslant \widehat{V}^k_{h,l}(s)$ for any $l \in [l^k_h(s) - 1]$. From the definition of $\check{V}^k_{h,l}(s)$ and $\widehat{V}^k_{h,l}(s)$, they are monotonic in l that $\widehat{V}^k_{h,l-1}(s) \leqslant \widehat{V}^k_{h,l}(s)$ and $\widehat{V}^k_{h,l}(s) \leqslant \widehat{V}^k_{h,l-1}(s)$ hold. Combining with $\widehat{V}^k_{h+1}(s) = \widehat{V}^k_{h,l^k_h(s)-1}$, we have

$$\forall l \in [l^k_h(s) - 1], \check{V}^k_{h,l}(s) \leqslant \widehat{V}^k_h(s) \leqslant \widehat{V}^k_{h,l}(s) \tag{5.9.3}$$

From the definition of $\widehat{V}^k_{h,l}(s)$ and $\check{V}^k_{h,l}(s)$, we have

$$0 \leqslant \widehat{V}^k_{h,l}(s) - \check{V}^k_{h,l}(s) \leqslant \big(\widehat{V}^k_{h,l}(s) - V^k_{h,l}(s)\big) + \big(V^k_{h,l}(s) - \check{V}^k_{h,l}(s)\big) \leqslant 6 \cdot 2^{-l}. \tag{5.9.4}$$

Plugging (5.9.3) into (5.9.4), we conclude that for any phase $l \in [l^k_h(s) - 1]$, it holds that $|\widehat{V}^k_h(s) - \widehat{V}^k_{h,l}(s)| \leqslant 6 \cdot 2^{-l}$.

Now consider the extended state value function $\widehat{V}^k_{h,l_+}$ with an arbitrary $l_+ \in \mathbb{N}^+$. For every s where $l_+ \leqslant l^k_h(s) - 1$, we have $|\widehat{V}^k_h(s) - V^k_{h,l_+}(s)| \leqslant 6 \cdot 2^{-l_+}$ as reasoned above. For the other $s \in \mathcal{S}$ where $l_+ \geqslant l^k_h(s)$, we have $\widehat{V}^k_{h,l}(s) = \widehat{V}^k_h(s)$ following the procedure of Algorithm 13. This suggest that $|\widehat{V}^k_h(s) - \widehat{V}^k_{h,l_+}(s)| \leqslant 6 \cdot 2^{-l_+}$ always holds. □

5.9.2.2 Proof of Lemma 5.9.4

The following Lemma shows the rounding only cast bounded effects on the recovered parameters.

Lemma 5.9.15. For any $(k, h, s) \in [K] \times [H] \times \mathcal{S}, l \in [l^k_h(s) - f^k_h(s)], a \in \mathcal{A}^k_{h,l}(s)$, it holds that

$$\big|\big\langle \boldsymbol{\phi}(s,a), \mathbf{w}^k_{h,l}\big\rangle - \big\langle \boldsymbol{\phi}(s,a), \widetilde{\mathbf{w}}^k_{h,l}\big\rangle\big| \leqslant 0.01 \cdot 2^{-4l}, \big|\|\boldsymbol{\phi}(s,a)\|_{(\mathbf{U}^k_{h,l})^{-1}} - \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}^{k,-1}_{h,l}}\big| \leqslant 0.1 \cdot 2^{-2l}.$$

The following lemma shows the number of episodes that are taken into regression $|\mathcal{C}_{h,l}^k|$ is bounded independently from the number of episodes k.

Lemma 5.9.16. For any tuple $(k,h,l) \in [K] \times [H] \times \mathbb{N}^+$, we have $|\mathcal{C}_{h,l}^k| \leqslant 16l \cdot 4^l \gamma_l^2 d$.

The following lemma shows the number of possible state value functions $|\mathcal{V}_{h,l}^k|$ is bounded independently from the number of episodes k.

Lemma 5.9.17. For any tuple $(k,h,l) \in [K] \times [H] \times \mathbb{N}^+$, we have $|\mathcal{V}_{h,l}^k| \leqslant (2^{22} d^6 H^4)^{l^2 d^2}$.

Now we are ready to prove Lemma 5.9.4.

Proof of Lemma 5.9.4. Recall in Definition 5.9.3, the good event defined by the union of each single bad event:

$$\mathcal{G}_1 = \bigcap_{k=1}^{K} \bigcap_{h=1}^{H} \bigcap_{l \geqslant 1} \bigcap_{V \in \mathcal{V}_{h,l_+}^k} \mathcal{B}_1^{\complement}(k,h,l,V),$$

where each single bad event is given by

$$\mathcal{B}_1(k,h,l,V) = \left\{ \left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h V](s_h^\tau, a_h^\tau) - V(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} > \gamma_l \right\},$$

in which $[\mathbb{P}_h V](s,a) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} V(s)$.

Consider some fixed $(h,l) \in [H] \times \mathbb{N}^+$, $V \in \mathcal{V}_{h,l_+}^K$. Arrange elements of $\mathcal{C}_{h,l}^k$ in ascending order as $\{\tau_i\}_i$. Since the environment sample $s_{h+1}^{\tau_i}$ according to $\mathbb{P}_h(\cdot|s_h^{\tau_i}, a_h^{\tau_i})$, we have $[\mathbb{P}_h V](s_h^{\tau_i}, a_h^{\tau_i}) - V(s_{h+1}^{\tau_i})$ is $\mathcal{F}_h^{\tau_i}$-measurable with $\mathbb{E}\big[[\mathbb{P}_h V](s_h^{\tau_i}, a_h^{\tau_i}) - V(s_{h+1}^{\tau_i}) \big| \mathcal{F}_h^{\tau_i} \big] = 0$. Since

$0 \leqslant V(s_{h+1}^{\tau_i}) \leqslant H$, we have $|[\mathbb{P}_h V](s_h^{\tau_i}, a_h^{\tau_i}) - V(s_{h+1}^{\tau_i})| \leqslant H$. This further leads to

$$\begin{aligned}
&\Big\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^{\tau} \big([\mathbb{P}_h V](s_h^{\tau}, a_h^{\tau}) - V(s_{h+1}^{\tau}) \big) \Big\|_{(\mathbf{U}_{h,l}^k)^{-1}} \\
&= \Big\| \sum_{i=1}^{|\mathcal{C}_{h,l}^{k-1}|} \boldsymbol{\phi}_h^{\tau_i} \big([\mathbb{P}_h V](s_h^{\tau_i}, a_h^{\tau_i}) - V(s_{h+1}^{\tau_i}) \big) \Big\|_{(\mathbf{U}_{h,l}^k)^{-1}} \\
&\leqslant H \sqrt{2d \ln \big(1 + |\mathcal{C}_{h,l}^k|/(d\lambda)\big) + 2 \ln(l^2 H |\mathcal{V}_{h,l_+}^K|/\delta)} \\
&\leqslant H \sqrt{2d \ln(1 + l \cdot 4^l \gamma_l^2) + 2 \ln(l^2 H (2^{22} d^6 H^4)^{l_+^2 d^2}/\delta)} \\
&\leqslant \gamma_{l,l_+},
\end{aligned}$$

where the first inequality holds following from the good event of probability $1 - \delta/(l^2 H |\mathcal{V}_{h,l_+}^K|)$ defined in Lemma 4.9.9 over filtration $\{\mathcal{F}_h^{\tau_i}\}_i$, the second inequality is derived from combining Lemma 5.9.16 and Lemma 5.9.17, and the last inequality is given by Lemma 5.9.39. According to Lemma 4.9.9, we have the bad event $\bigcup_{k=1}^K \mathcal{B}_1(k, h, l, V)$ happens with probability at most $\delta/(l^2 H |\mathcal{V}_{h,l_+}^K|)$. Taking union bound over all $(h, l) \in [H] \times \mathbb{N}^+$, $V \in \mathcal{V}_{h,l_+}^K$, we have the bad event happens with probability at most

$$\Pr[\mathcal{G}_1^{\complement}] \leqslant \sum_{h=1}^{H} \sum_{l=1}^{\infty} \sum_{V \in \mathcal{V}_{h,l_+}^K} \Pr\Big[\bigcup_{k=1}^{K} \mathcal{B}_1(k, h, l, V) \Big] \leqslant \sum_{h=1}^{H} \sum_{l=1}^{\infty} \sum_{V \in \mathcal{V}_{h,l_+}^K} \frac{\delta}{l^2 H |\mathcal{V}_{h,l_+}^K|} \leqslant 2\delta,$$

where the last inequality holds due to $\sum_{n \geqslant 1} n^{-2} = \pi^2/6$. This completes our proof. □

5.9.2.3 Proof of Lemma 5.9.5

Proof of Lemma 5.9.5. Denote the martingale difference between $\widehat{V}_{h,l_+}^k - \widehat{V}_h^k$ as:

$$\mu_{h,l}^k = [\mathbb{P}_h(\widehat{V}_{h,l_+}^k - \widehat{V}_{h+1}^k)](s_h^k, \pi_h^k(s_h^k)) - \big(\widehat{V}_{h,l_+}^k(s_{h+1}^k) - \widehat{V}_{h+1}^k(s_{h+1}^k) \big).$$

By triangle inequality:

$$\begin{aligned}
&\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau) - \widehat{V}_{h+1}^k(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \\
\leqslant &\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h V_{h,l_+}^k](s_h^\tau, a_h^\tau) - \widehat{V}_{h,l_+}^k(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} + \left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \mu_{h,l_+}^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}}. \qquad (5.9.5)
\end{aligned}$$

According to the definition of event $\mathcal{G}_1$, we can upper bound the first term by

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h V_{h,l_+}^k](s_h^\tau, a_h^\tau) - \widehat{V}_{h,l_+}^k(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant \gamma_{l,l_+} = \gamma_l. \qquad (5.9.6)$$

According to Lemma 5.9.2, we have $|\widehat{V}_{h,l_+}^k(s) - \widehat{V}_{h+1}^k(s)| \leqslant 6 \cdot 2^{-l_+}$ for any $s \in \mathcal{S}$. Thus, the difference can be bounded by $|\mu_{h,l_+}^\tau| \leqslant 6 \cdot 2^{-l_+}$. Consequently, we can bound the second term by

$$\begin{aligned}
\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \mu_{h,l_+}^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} &\leqslant 6 \cdot 2^{-l_+} \sqrt{|\mathcal{C}_{h,l}^k|} \\
&\leqslant 6 \cdot 2^{-l_+} \sqrt{16 l \cdot 4^l \gamma_l^2 d} \\
&= 24 \cdot 2^{l - l_+} \gamma_l \sqrt{ld}, \qquad (5.9.7)
\end{aligned}$$

where the first inequality is provided by Lemma 4.9.10, utilizing the condition $|\mu_{h,l_+}^\tau| \leqslant 6 \cdot 2^{-l_+}$, the second inequality is from Lemma 5.9.16. By plugging in the definition of l_+, we can further bound the final term of (5.9.7) by

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \mu_{h,l_+}^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 24 \cdot 2^{l - l_+} \gamma_l \sqrt{ld} \leqslant 24 \cdot 2^{-20} \gamma_l \leqslant 0.1 \gamma_l. \qquad (5.9.8)$$

Plugging (5.9.6) and (5.9.8) into (5.9.5) yields the desired statement such that

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \big([\mathbb{P}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau) - \widehat{V}_{h+1}^k(s_{h+1}^\tau) \big) \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 1.1 \gamma_l,$$

which concludes our proof. □

5.9.2.4 Proof of Lemma 5.9.6

The following lemma shows the state-action value function $Q^k_{h,l}(s,a)$ is always well estimated.

Lemma 5.9.18. Under event $\mathcal{G}_1$, for any $(k,h,l,s,a) \in [K] \times [H] \times \mathbb{N}^+ \times \mathcal{S} \times \mathcal{A}$,

$$\left|Q^k_{h,l}(s,a) - [\mathbb{B}_h \widehat{V}^k_{h+1}](s,a)\right| \leqslant \left(1.2 + 8\sqrt{ld}H \cdot 2^l\zeta\right)\gamma_l \|\boldsymbol{\phi}(s,a)\|_{(\mathbf{U}^k_{h,l})^{-1}} + 0.01 \cdot 2^{-4l} + 2H\zeta.$$

Equipped with Lemma 5.9.15 and Lemma 5.9.18, we are ready to prove Lemma 5.9.6.

Proof of Lemma 5.9.6. In case that $l \leqslant l^k_h(s) - f^k_h(s)$, for any $a \in \mathcal{A}^k_{h,l}(s)$, we have that

$$\begin{aligned}\|\boldsymbol{\phi}(s,a)\|_{(\mathbf{U}^k_{h,l})^{-1}} &\leqslant \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}^{k,-1}_{h,l}} + \left|\|\boldsymbol{\phi}(s,a)\|_{(\mathbf{U}^k_{h,l})^{-1}} - \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}^{k,-1}_{h,l}}\right| \\ &\leqslant 2^{-l}\gamma_l^{-1} + 0.1 \cdot 2^{-2l} \leqslant 1.1 \cdot 2^{-l}\gamma_l^{-1}, \end{aligned} \tag{5.9.9}$$

where the first inequality holds due to triangle inequality, and in the second inequality, the first term is satisfied since state s passes the criterion in Line 9 in phase l and the second term follows from Lemma 5.9.15, and the last inequality is given by Lemma 5.9.38 which implies $2^l > \gamma_l$. Plugging (5.9.9) into Lemma 5.9.18 gives

$$\begin{aligned}\left|Q^k_{h,l}(s,a) - [\mathbb{B}_h \widehat{V}^k_{h+1}](s,a)\right| &\leqslant 0.01 \cdot 2^{-4l} + 1.32 \cdot 2^{-l} + 8.8\sqrt{ld}H\zeta + 2H\zeta \\ &\leqslant 2 \cdot 2^{-l} + 12\sqrt{ld}H\zeta,\end{aligned}$$

which proves the desired statement. □

5.9.2.5 Proof of Lemma 5.9.7

Equipped with Lemma 5.9.6, we are able to show several properties of the state value function $V^k_{h,l}$ through the arm-elimination process. The first lemma suggests that for any action $a_l \in \mathcal{A}^k_{h,l}(s)$, there is at least one action $a_{l+1} \in \mathcal{A}^k_{h,l+1}(s)$ close to a_l in terms of the Bellman operator $[\mathbb{B}_h \widehat{V}^k_{h+1}](s,a)$ after the elimination.

Lemma 5.9.19. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [\min\{L_0, l_h^k(s) - f_h^k(s)\}], a_l \in \mathcal{A}_{h,l}^k(s)$, there exists $a_{l+1} \in \mathcal{A}_{h,l+1}^k(s)$ that

$$[\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_l) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_{l+1}) \leqslant 2\chi\sqrt{L_0}\zeta$$

where $\chi = 12\sqrt{d}H$ for arbitrary $L_0 \geqslant 1$.

Then the following lemma shows that by induction on stage $h \in [H]$, we can show the elimination process keep at least one near-optimal action $a_{l+1} \in \mathcal{A}_{h,l+1}^k(s)$.

Lemma 5.9.20. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [\min\{L_0, l_h^k(s) - f_h^k(s)\}]$, there exists $a_{l+1} \in \mathcal{A}_{h,l+1}^k(s)$ that,

$$\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_{l+1}) \leqslant 2l \cdot \chi\sqrt{L_0}\zeta.$$

where $\chi = 12\sqrt{d}H$ for arbitrary $L_0 \geqslant 1$.

The following two lemmas indicate that the state value function $V_{h,l}^k(s)$ on stage h is a good estimation for the value function given by Bellman operator $V(s) = \max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a)$.

Lemma 5.9.21. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [\min\{L_0, l_h^k(s) - f_h^k(s)\}]$,

$$\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a) - V_{h,l}^k(s) \leqslant 2 \cdot 2^{-l} + (2l-1)\chi\sqrt{L_0}\zeta.$$

where $\chi = 12\sqrt{d}H$ for arbitrary $L_0 \geqslant 1$.

Lemma 5.9.22. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K] \times [H] \times \mathcal{S}, l \in [\min\{L_0, l_h^k(s) - f_h^k(s)\}]$,

$$V_{h,l}^k(s) - \max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a) \leqslant 2 \cdot 2^{-l} + \chi\sqrt{L_0}\zeta,$$

where $\chi = 12\sqrt{d}H$ for arbitrary $L_0 \geqslant 1$.

Now we are ready to show any actions remaining in the elimination process are near-optimal.

Proof of Lemma 5.9.7. First, according to Lemma 5.9.21, we can write

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - V_{h,l}^k(s) \leqslant 2\cdot 2^{-l} + (2l-1)\chi\sqrt{L_0}\zeta. \tag{5.9.10}$$

Any action $a_{l+1} \in \mathcal{A}_{h,l+1}^k(s)$ passes the elimination process will satisfy:

$$Q_{h,l}^k(s,a_{l+1}) \geqslant V_{h,l}^k(s) - 4\cdot 2^{-l}. \tag{5.9.11}$$

According to Lemma 5.9.6 with the condition that $l \leqslant L_0$, we have that the empirical state-action value function $Q_{h,l}^k(s,\cdot)$ is a good estimation for $[\mathbb{B}_h\widehat{V}_{h+1}^k](s,\cdot)$ among every $a_{l+1} \in \mathcal{A}_l^k(s)$ under event $\mathcal{G}_1$:

$$\left|[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a_{l+1}) - Q_{h,l}^k(s,a_{l+1})\right| \leqslant 2\cdot 2^{-l} + \chi\sqrt{L_0}\zeta. \tag{5.9.12}$$

Combining (5.9.10), (5.9.11), and (5.9.12) gives

$$\begin{aligned}
&\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,a_{l+1}) \\
&= \big(\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - V_{h,l}^k(s)\big) + \big(V_{h,l}^k(s) - Q_{h,l}^k(s,a_{l+1})\big) \\
&\quad + \big(Q_{h,l}^k(s,a_{l+1}) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,a_{l+1})\big) \\
&\leqslant \big(2\cdot 2^{-l} + (2l-1)\chi\sqrt{L_0}\zeta\big) + 4\cdot 2^{-l} + \big(2\cdot 2^{-l} + \chi\sqrt{L_0}\zeta\big) \\
&= 8\cdot 2^{-l} + 2l\cdot\chi\sqrt{L_0}\zeta,
\end{aligned}$$

which proves the desired statement. □

5.9.2.6 Proof of Lemma 5.9.8

The following two lemmas demonstrate that, at stage h, both the optimistic state value function, $\widehat{V}_{h,l}^k(s)$, and the pessimistic state value function, $\check{V}_{h,l}^k(s)$, exhibit a gap relative to the state value function determined by the Bellman operator as $V(s) = \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)$.

Lemma 5.9.23. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K]\times[H]\times\mathcal{S}, l \in [\min\{L_0, l_h^k(s) - f_h^k(s)\}]$,

$$\min\left\{V_{h,l}^k(s) + 3\cdot 2^{-l}, \widehat{V}_{h,l-1}^k(s)\right\} - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) \geqslant 2^{-l} - (2l-1)\chi\sqrt{L_0}\zeta,$$

where $\chi = 12\sqrt{d}H$ for arbitrary $L_0 \geqslant 1$. In case that $l \leqslant l_h^k(s)-1$, the inequality is equivalent to

$$\widehat{V}_{h,l}^k(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) \geqslant 2^{-l} - (2l-1)\chi\sqrt{L_0}\zeta.$$

Lemma 5.9.24. Under event $\mathcal{G}_1$, for any $(k,h,s) \in [K]\times[H]\times\mathcal{S}, l \in [\min\{L_0, l_h^k(s)-f_h^k(s)\}]$,

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \max\left\{V_{h,l}^k(s) - 3\cdot 2^{-l}, \check{V}_{h,l-1}^k(s)\right\} \geqslant 2^{-l} - \chi\sqrt{L_0}\zeta,$$

where $\chi = 12\sqrt{d}H$ for arbitrary $L_0 \geqslant 1$. In case that $l \leqslant l_h^k(s)-1$, the inequality is equivalent to

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \check{V}_{h,l}^k(s) \geqslant 2^{-l} - \chi\sqrt{L_0}\zeta.$$

Proof of Lemma 5.9.8. Set $L_0 = L_\zeta$ be the maximal integer satisfying $2^{-L_\zeta} - \chi L_\zeta^{1.5}\zeta \geqslant 0$. Combining Lemma 5.9.24 and Lemma 5.9.23, for any $l \in [\min\{L_0, l_h^k(s) - f_h^k(s)\}]$, we have that

$$\begin{aligned}
&\min\left\{V_{h,l}^k(s) + 3\cdot 2^{-l}, \widehat{V}_{h,l-1}^k(s)\right\} - \max\left\{V_{h,l}^k(s) - 3\cdot 2^{-l}, \check{V}_{h,l-1}^k(s)\right\} \\
&= \left(\widehat{V}_{h,l}^k(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)\right) + \left(\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \check{V}_{h,l}^k(s)\right) \\
&\geqslant \left(2^{-l} - (2l-1)\chi\sqrt{L_0}\zeta\right) + \left(2^{-l} - \chi\sqrt{L_0}\zeta\right) \\
&= 2\cdot 2^{-l} - 2l\cdot\chi\sqrt{L_0}\zeta \\
&\geqslant 2\cdot 2^{-L_0} - 2\chi L_0^{1.5}\zeta \geqslant 0.
\end{aligned}$$

where the second inequality holds since 2^{-l} decreases as l increases and the last inequality holds according to the selection of L_0.

When $f_h^k(s) = 0$, consider $l = l_h^k(s)$. The above reasoning indicates the criterion in Line 11 can never satisfied. Thus $f_h^k(s) = 0$ can only happen if $l_h^k(s) > L_0 = L_\zeta$. □

5.9.2.7 Proof of Lemma 5.9.9

By partitioning $[K]$ based on whether Algorithm 13 stops before phase L_ε, we can prove Lemma 5.9.9. Specifically, Lemma 5.9.16 bounds the number of episodes in which Algo-

rithm 13 stops before phase L_ε. This allows us to establish an upper bound for the desired summation over these episodes. Furthermore, for episodes that stop after phase L_ε, the contribution of $2^{-l_h^k(s_h^k)}\gamma_{l_h^k(s_h^k)}$ is small according to the definition of L_ε.

Proof of Lemma 5.9.9. Denote $\mathcal{C}_{h,+}^K = [K] - \bigcup_{l=1}^{L_\varepsilon - 1} \mathcal{C}_{h,l}^K$. In this sense, we have

$$\sum_{k\in\mathcal{K}} 2^{-l_h^k(s_h^k)} = \sum_{k\in\mathcal{K}\cap\mathcal{C}_{h,+}^K} 2^{-l_h^k(s_h^k)} + \sum_{l=1}^{L_\varepsilon-1} \sum_{k\in\mathcal{K}\cap\mathcal{C}_{h,l}^K} 2^{-l_h^k(s_h^k)}. \tag{5.9.13}$$

From the construction of $\mathcal{C}_{h,l}^K$, we have $l_h^k(s_h^k) = l$ for any $k \in \mathcal{C}_{h,l}^K$. Fix some $k \in \mathcal{C}_{h,+}^K$. If $f_h^k(s_h^k) = 0$, we have $l_h^k(s_h^k) \geqslant L_\zeta \geqslant L_\varepsilon$ where the first inequality is given by Lemma 5.9.8 and the second inequality is given by the assignment of L_ε. Otherwise, we have $l_h^k(s_h^k) \geqslant L_\varepsilon$ according to the definition of $\mathcal{C}_{h,l}^K$. This indicates $l_h^k(s_h^k) \geqslant L_\varepsilon$ holds for any $k \in \mathcal{C}_{h,+}^K$. This allow is to bound the first term by

$$\sum_{k\in\mathcal{K}\cap\mathcal{C}_{h,+}^K} 2^{-l_h^k(s_h^k)} \leqslant \sum_{k\in\mathcal{K}\cap\mathcal{C}_{h,+}^K} 2^{-L_\varepsilon} \leqslant 0.01|\mathcal{K}| \cdot \varepsilon/H, \tag{5.9.14}$$

where the first inequality holds since $l_h^k(s_h^k) > L_\varepsilon$ and the second inequality holds from both $2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$ and $|\mathcal{K} \cap \mathcal{C}_{h,+}^K| \leqslant |\mathcal{K}|$.

Furthermore, we can bound the second term by

$$\begin{aligned}
\sum_{l=1}^{L_\varepsilon-1} \sum_{k\in\mathcal{K}\cap\mathcal{C}_{h,l}^K} 2^{-l_h^k(s_h^k)} &\leqslant \sum_{l=1}^{L_\varepsilon-1} |\mathcal{K} \cap \mathcal{C}_{h,l}^K| \cdot 2^{-l} \\
&\leqslant \sum_{l=1}^{L_\varepsilon-1} 16l \cdot 4^l \gamma_l^2 d \cdot 2^{-l} \\
&\leqslant 16 L_\varepsilon d \cdot 2^{L_\varepsilon} \gamma_{L_\varepsilon}^2 \leqslant 2^{12} L_\varepsilon dH \gamma_{L_\varepsilon}^2 \varepsilon^{-1}.
\end{aligned} \tag{5.9.15}$$

where the second inequality is given by Lemma 5.9.16, and the last inequality holds due to $0.005\varepsilon/H \leqslant 2^{-L_\varepsilon}$ which is because L_ε is a minimal integer such that $2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$.

Finally, plugging (5.9.14) and (5.9.15) into (5.9.13) gives

$$\sum_{k\in\mathcal{K}} 2^{-l_h^k(s_h^k)} \leqslant 0.01|\mathcal{K}| \cdot \varepsilon/H + 2^{12} L_\varepsilon dH \gamma_{L_\varepsilon}^2 \varepsilon^{-1}.$$

□

5.9.2.8 Proof of Lemma 5.9.12

The following lemma provides an upper bound for the underestimation error of the empirical state value function $\widehat{V}_h^k$ with respect to the optimal state value function V_h^*.

Lemma 5.9.25. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $(k, h, s) \in [K] \times [H] \times \mathcal{S}$,

$$V_h^*(s) - \widehat{V}_h^k(s) \leqslant 0.07\varepsilon.$$

As $\widehat{V}_h^k$ represents an empirical state value function with potentially optimal policy $\pi_h^k(s)$, the following lemma provides an upper bound for the overestimation error of $\widehat{V}_h^k$ with respect to deploying the policy $\pi_h^k(s)$ on the ground-truth transition kernel.

Lemma 5.9.26. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $(k, h, s) \in [K] \times [H] \times \mathcal{S}$,

$$\widehat{V}_h^k(s) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, \pi_h^k(s)) \leqslant 20 \cdot 2^{-l_h^k(s)} + 0.16\varepsilon/H.$$

To start with, we define a good event according to:

Definition 5.9.27. For some $\varepsilon > 0$, let $\mathcal{K}_h^\varepsilon = \{k \in [K] : V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \geqslant \varepsilon\}$. We define the bad event as

$$\mathcal{B}_2(h, \varepsilon) = \left\{ \sum_{k \in \mathcal{K}_h^\varepsilon} \sum_{h'=h}^{H} \eta_{h'}^k > 4\sqrt{H^3 |\mathcal{K}_h^\varepsilon| \log(4H|\mathcal{K}_h^\varepsilon| \log(\varepsilon^{-1})/\delta)} \right\}.$$

where $\eta_h^k = [\mathbb{P}_h(\widehat{V}_{h+1}^k - V_{h+1}^{\pi^k})](s_h^k, \pi_h^k(s_h^k)) - \big(\widehat{V}_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi^k}(s_{h+1}^k)\big)$. The good event is defined as $\mathcal{G}_2 = \bigcap_{h=1}^H \bigcap_{l \geqslant 1} \mathcal{B}_2^\complement(h, 2^{-l})$.

The following lemma provides the concentration property such that the cumulative sample error is small with high probability.

Lemma 5.9.28. Event $\mathcal{G}_2$ happens with probability at least $1 - \delta$.

Using the above results, we can bound the instantaneous regret of any subsets once the misspecification level is appropriately controlled,

Lemma 5.9.29. Under event $\mathcal{G}_1, \mathcal{G}_2$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $\mathcal{K} \subseteq [K]$ and $h \in [H]$, it satisfies that

$$\sum_{k \in \mathcal{K}} \left(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\right) \leqslant 0.49|\mathcal{K}|\varepsilon + 2^{17} L_\varepsilon d H^2 \gamma_{L_\varepsilon}^2 \varepsilon^{-1} + 4\sqrt{H^3|\mathcal{K}| \log(4H|\mathcal{K}| \log(\varepsilon^{-1})/\delta)}.$$

With all lemmas stated above, we can show Cert-LSVI-UCB achieves constant step-wise decision error. The following lemma gives a sufficient condition that $\mathcal{G}_\varepsilon$ defined in Definition 5.9.10 is satisfied.

Now, we are ready to prove Lemma 5.9.12.

Proof of Lemma 5.9.12. We focus on the case where the good event $\mathcal{G}_1 \cap \mathcal{G}_2 \cap \mathcal{G}_\varepsilon$ occurs. By the union bound statement over Lemma 5.9.4 and Lemma 5.9.28, and Lemma 5.9.11, this good event happens with a probability of at least $1 - 3\delta$ and with the condition that $\varepsilon \geqslant \Omega(\zeta \sqrt{d} H^2 \log^2(dH\zeta^{-1}))$. W.l.o.g, consider $\mathcal{K}_h^\varepsilon$ for some $h \in [H]$ and $\varepsilon = 2^{-l}$ where $l > 0$ is an integer. On the one hand, we have

$$\sum_{k \in \mathcal{K}_h^\varepsilon} \left(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\right) \geqslant |\mathcal{K}_h^\varepsilon|\varepsilon. \tag{5.9.16}$$

On the other hand, Lemma 5.9.29 gives

$$\begin{aligned}\sum_{k \in \mathcal{K}_h^\varepsilon} \left(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\right) \leqslant{}& 0.49|\mathcal{K}_h^\varepsilon|\varepsilon + 2^{17} L_\varepsilon d H^2 \gamma_{L_\varepsilon}^2 \varepsilon^{-1} \\ &+ 4\sqrt{H^3|\mathcal{K}_h^\varepsilon| \log(4H|\mathcal{K}_h^\varepsilon| \log(\varepsilon^{-1})/\delta)}.\end{aligned} \tag{5.9.17}$$

Combining (5.9.16) and (5.9.17) gives

$$0.51|\mathcal{K}_h^\varepsilon|\varepsilon \leqslant 2^{17} L_\varepsilon d H^2 \gamma_{L_\varepsilon}^2 \varepsilon^{-1} + 4\sqrt{H^3|\mathcal{K}_h^\varepsilon| \log(4H|\mathcal{K}_h^\varepsilon| \log(\varepsilon^{-1})/\delta)}.$$

Plugging the value of γ_{L_ε}, we have

$$\begin{aligned}0.51|\mathcal{K}_h^\varepsilon|\varepsilon \leqslant{}& 2^{22} L_\varepsilon (L_\varepsilon + \log(2^{20} dH))^2 d^3 H^4 \varepsilon^{-1} \log(16 L_\varepsilon d/\delta) \\ &+ 4\sqrt{H^3|\mathcal{K}_h^\varepsilon| \log(4H|\mathcal{K}_h^\varepsilon| \log(\varepsilon^{-1})/\delta)}.\end{aligned} \tag{5.9.18}$$

According to Lemma 5.9.40, (5.9.18) implies

$$|\mathcal{K}_h^\varepsilon| \leqslant \mathcal{O}(L_\varepsilon(L_\varepsilon + \log(dH))^2 d^3 H^4 \varepsilon^{-2} \log(L_\varepsilon d) \log(\delta^{-1})\iota),$$

where ι refers to a polynomial of $\log\log(dH\varepsilon^{-1}\delta^{-1})$. With the definition of L_ε, we conclude that

$$|\mathcal{K}_h^\varepsilon| \leqslant \mathcal{O}(d^3 H^4 \varepsilon^{-2} \log^4(dH\varepsilon^{-1}) \log(\delta^{-1})\iota).$$

□

5.9.2.9 Proof of Lemma 5.9.13

Proof of Lemma 5.9.13. From the definition of suboptimality gap, we have

$$\Delta_h^k = V_h^*(s_h^k) - [\mathbb{B}_h V_{h+1}^*](s_h^k, \pi_h^k(s_h^k)) \leqslant V_h^*(s_h^k) - [\mathbb{B}_h V_{h+1}^{\pi^k}](s_h^k, \pi_h^k(s_h^k)) = V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k). \tag{5.9.19}$$

According to the assumption,

$$\sum_{k=1}^K \mathbb{1}\left[V_h^*(s_1^k) - V_h^{\pi^k}(s_h^k) \geqslant \varepsilon\right] \leqslant \left(\frac{C_1}{\varepsilon} + \frac{C_2}{\varepsilon^2}\right) \log^a\left(\frac{C_1}{\varepsilon} + \frac{C_2}{\varepsilon^2}\right)$$

holds for every $\varepsilon > \varepsilon_0$ with probability at least $1-\delta$, replacing the $V_h^*(s_1^k) - V_h^{\pi^k}(s_h^k)$ with its lower bound Δ_h^k yields for every $\varepsilon > \varepsilon_0$,

$$\sum_{k=1}^K \mathbb{1}\left[\Delta_h^k \geqslant \varepsilon\right] \leqslant \left(\frac{C_1}{\varepsilon} + \frac{C_2}{\varepsilon^2}\right) \log^a\left(\frac{C_1}{\varepsilon} + \frac{C_2}{\varepsilon^2}\right).$$

In addition, according to the definition of minimal suboptimality gap Δ in Definition 5.3.3, we have Δ_h^k is either 0 or no less than Δ. Since for any $x \in \{0\} \cup [\Delta, H]$, it holds that $x \leqslant \Delta \cdot \mathbb{1}[x \geqslant \Delta] + \int_\Delta^H \mathbb{1}[x \geqslant \varepsilon]\,\mathrm{d}\varepsilon$, we decompose the total suboptimality incurred in stage h by

$$\begin{aligned}\sum_{k=1}^K \Delta_h^k &\leqslant \sum_{k=1}^K \left(\Delta \cdot \mathbb{1}\left[\Delta_h^k \geqslant \Delta\right] + \int_\Delta^H \mathbb{1}\left[\Delta_h^k \geqslant \varepsilon\right] \mathrm{d}\varepsilon\right) \\ &= \Delta \sum_{k=1}^K \mathbb{1}\left[\Delta_h^k \geqslant \Delta\right] + \int_\Delta^H \sum_{k=1}^K \mathbb{1}\left[\Delta_h^k \geqslant \varepsilon\right] \mathrm{d}\varepsilon. \end{aligned} \tag{5.9.20}$$

In case that $\varepsilon_0 \leqslant \Delta$, the first term in (5.9.20) can be bounded by

$$\Delta \sum_{k=1}^{K} \mathbb{1}\left[\Delta_h^k \geqslant \Delta\right] \leqslant \Delta\Big(\frac{C_1}{\Delta} + \frac{C_2}{\Delta^2}\Big) \log^a\Big(\frac{C_1}{\Delta} + \frac{C_2}{\Delta^2}\Big). \tag{5.9.21}$$

We can further bound the second term by

$$\begin{aligned}\int_{\Delta}^{H} \sum_{k=1}^{K} \mathbb{1}\left[V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k) \geqslant \varepsilon\right] \mathrm{d}\varepsilon &\leqslant \int_{\Delta}^{H} \Big(\frac{C_1}{\varepsilon} + \frac{C_2}{\varepsilon^2}\Big) \log^a\Big(\frac{C_1}{\varepsilon} + \frac{C_2}{\varepsilon^2}\Big) \mathrm{d}\varepsilon \\ &\leqslant \log^a\Big(\frac{C_1}{\Delta} + \frac{C_2}{\Delta^2}\Big) \cdot \Big(C_1 \ln\frac{H}{\Delta} + \frac{C_2}{\Delta}\Big) \\ &\leqslant (C_1 \log H + C_2/\Delta) \cdot \mathrm{polylog}(C_1, C_2, \Delta^{-1}). \end{aligned} \tag{5.9.22}$$

Plugging (5.9.21) and (5.9.22) into (5.9.20) with summation over $h \in [H]$, we conclude that the total suboptimality incurred in stage h is bounded by

$$\begin{aligned}\sum_{k=1}^{K}\sum_{h=1}^{H} \Delta_h^k &\leqslant H \cdot (C_1 + C_2/\Delta + C_1 \log H + C_2/\Delta) \cdot \mathrm{polylog}(C_1, C_2, \Delta^{-1}) \\ &\leqslant \widetilde{\mathcal{O}}(C_2 H/\Delta + C_1 H).\end{aligned}$$

□

5.9.2.10 Proof of Lemma 5.9.14

We first introduce the Freedman inequality:

Lemma 5.9.30 (Freedman inequality, Cesa-Bianchi and Lugosi (2006))**.** Let $\{\eta^k\}_{k=1}^K$ be a martingale difference sequence with respect to a filtration $\{\mathcal{F}^k\}_{k=1}^K$ satisfying $|\eta^k| \leqslant M$ for some constant $M > 0$ and η^k is $\mathcal{F}^{k+1}$-measurable with $|\mathbb{E}[\eta^k|\mathcal{F}^k]| = 0$. Then for some fixed $k \in [K]$, $a > 0$ and $v > 0$, we have

$$\Pr\Big(\sum_{\tau=1}^{k} \eta^\tau \geqslant a, \sum_{\tau=1}^{k} \mathrm{Var}[\eta^\tau|\mathcal{F}^\tau] \leqslant v\Big) \leqslant \exp\Big(\frac{-a^2}{2v + 2aM/3}\Big).$$

We are now ready to present the proof of Lemma 5.9.14.

Proof of Lemma 5.9.14. For a given policy π and any state $s_h \in \mathcal{S}$, we have

$$\begin{aligned}
&V_h^*(s_h) - V_h^\pi(s_h) \\
&= \big(V_h^*(s_h) - [\mathbb{B}_h V_{h+1}^*](s_h, \pi_h(s_h))\big) + \big([\mathbb{B}_h V_{h+1}^*](s_h, \pi_h(s_h)) - [\mathbb{B}_h V_{h+1}^\pi](s_h, \pi_h(s_h))\big) \\
&= \Delta_h(s_h, \pi_h(s_h)) + [\mathbb{P}_h(V_{h+1}^* - V_{h+1}^\pi)](s_h, \pi_h(s_h)),
\end{aligned}$$

where the second equality is given by the definition of suboptimality gap $\Delta_h(\cdot, \cdot)$ in Definition 5.3.3. Taking expectation on both sides with respect to the randomness of state-transition and taking telescoping sum over all $h \in [H]$ gives

$$V_1^*(s_1) - V_h^\pi(s_1) = \mathbb{E}\Big[\sum_{h=1}^H \Delta_h(s_h, \pi_h(s_h))\Big],$$

where $s_{h+1} \sim \mathbb{P}_h(\cdot|s_h, \pi_h(s_h))$. Let the filtration list be $\mathcal{F}^k = \Big\{\big\{s_i^j, a_i^j\big\}_{i=1,j=1}^{H,k-1}\Big\}$, we have

$$\mathbb{E}\Big[\sum_{h=1}^H \Delta_h^k \Big| \mathcal{F}^k\Big] = V_1^*(s_1^k) - V_h^{\pi^k}(s_1^k).$$

Denote random variable $\eta^k = \big(V_1^*(s_1^k) - V_h^{\pi^k}(s_1^k)\big) - \sum_{h=1}^H \Delta_h^k$. We can see η^k is $\mathcal{F}_{k+1}$-measurable with $|\mathbb{E}[\eta^k|\mathcal{F}^k]| = 0$. Furthermore, for the variance of η^k, we have

$$\begin{aligned}
\mathrm{Var}[\eta^k|\mathcal{F}^k] &\leqslant \mathbb{E}\Big[\Big(\sum_{h=1}^H \Delta_h^k\Big)^2 \Big| \mathcal{F}^k\Big] \\
&\leqslant H^2 \mathbb{E}\Big[\sum_{h=1}^H \Delta_h^k \Big| \mathcal{F}^k\Big] \\
&= H^2\big(V_1^*(s_1^k) - V_h^{\pi^k}(s_1^k)\big),
\end{aligned}$$

where the first inequality holds due to $\mathrm{Var}[X] \leqslant \mathbb{E}[(X-t)^2]$ for any fixed t, the second inequality follows $0 \leqslant \Delta_h^k \leqslant H$. As a result, the total variance of the random variables $\{\eta^k\}$ can be bounded by

$$\sum_{k=1}^K \mathrm{Var}[\eta^k|\mathcal{F}^k] \leqslant \sum_{k=1}^K H^2\big(V_1^*(s_1^k) - V_h^{\pi^k}(s_1^k)\big) = H^2 \mathrm{Regret}(K).$$

Let $F(x) = \sqrt{2xH^2\log(x/\delta)} + H^2\log(x/\delta)$, using peeling technique, we can write

$$
\begin{aligned}
&\Pr\Big[\Big(\sum_{k=1}^{K}\eta^k\Big) \geqslant F(\mathrm{Regret}(K)), 1 < \mathrm{Regret}(K)\Big] \\
&= \Pr\Big[\Big(\sum_{k=1}^{K}\eta^k\Big) \geqslant F(\mathrm{Regret}(K)), 1 < \mathrm{Regret}(K), \sum_{k=1}^{K}\mathrm{Var}[\eta^k|\mathcal{F}^k] \leqslant H^2\mathrm{Regret}(K)\Big] \\
&\leqslant \sum_{i=1}^{\infty}\Pr\Big[\Big(\sum_{k=1}^{K}\eta^k\Big) \geqslant F(\mathrm{Regret}(K)), 2^{i-1} < \mathrm{Regret}(K) \leqslant 2^i, \\
&\qquad\qquad \sum_{k=1}^{K}\mathrm{Var}[\eta^k|\mathcal{F}^k] \leqslant H^2\mathrm{Regret}(K)\Big] \\
&\leqslant \sum_{i=1}^{\infty}\Pr\Big[\Big(\sum_{k=1}^{K}\eta^k\Big) \geqslant F(2^i), \sum_{k=1}^{K}\mathrm{Var}[\eta^k|\mathcal{F}^k] \leqslant 2^iH^2\Big] \\
&\leqslant \sum_{i=1}^{\infty}\exp\Big(\frac{-F(2^i)^2}{2^{i+1}H^2 + 2F(2^i)H^2/3}\Big), \qquad (5.9.23)
\end{aligned}
$$

where the last inequality follows Lemma 5.9.30. Plugging $F(x) = \sqrt{2xH^2\log(x/\delta)} + H^2\log(x/\delta)$ back into (5.9.23) yields

$$
\begin{aligned}
&\Pr\Big[\Big(\sum_{k=1}^{K}\eta^k\Big) \geqslant \sqrt{2\mathrm{Regret}(K)H^2\log(\mathrm{Regret}(K)/\delta)} + H^2\log(\mathrm{Regret}(K)/\delta), 1 < \mathrm{Regret}(K)\Big] \\
&\leqslant \sum_{i=1}^{\infty}\exp(-\log(2^i/\delta)) = \sum_{i=1}^{\infty}\delta/2^i = \delta.
\end{aligned}
$$

Therefore, whenever $\mathrm{Regret}(K) > 1$, with probability at least $1-\delta$, we have

$$
\sum_{k=1}^{K}\eta^k < \sqrt{2\mathrm{Regret}(K)H^2\log(\mathrm{Regret}(K)/\delta)} + H^2\log(\mathrm{Regret}(K)/\delta).
$$

Combining with the fact that $\mathrm{Regret}(K) = \sum_{k=1}^{K}\eta^k + \sum_{k=1}^{K}\sum_{h=1}^{H}\Delta_h^k$, we have

$$
\mathrm{Regret}(K) < \sum_{k=1}^{K}\sum_{h=1}^{H}\Delta_h^k + \sqrt{2\mathrm{Regret}(K)H^2\log(\mathrm{Regret}(K)/\delta)} + H^2\log(\mathrm{Regret}(K)/\delta),
$$

whenever $\mathrm{Regret}(K) > 1$. Since $x \leqslant a + \sqrt{bx}$ implies $x \leqslant 2a + 2b$, absorbing the case $\mathrm{Regret}(K) \leqslant 1$ into the $\widetilde{\mathcal{O}}(\cdot)$ factor yields

$$
\mathrm{Regret}(K) < \widetilde{\mathcal{O}}\Big(\sum_{k=1}^{K}\sum_{h=1}^{H}\Delta_h^k + H^2\log(1/\delta)\Big).
$$

□

5.9.3 Proof of Lemmas in Section 5.9.2

In this section, we prove lemmas outlined in Section 5.9.2. Any proofs not included in this section are deferred to Section 5.9.4.

5.9.3.1 Proof of Lemma 5.9.15

We first introduce the claim from Vial et al. (2022) controlling the rounding error:

Lemma 5.9.31 (Claim 1, Vial et al. 2022, restate)**.** For any $(k, h, l, s, a) \in [K] \times [H] \times \mathbb{N}^+ \times \mathcal{S} \times \mathcal{A}$, we have

$$\boldsymbol{\phi}(s,a)^\top(\mathbf{w}_{h,l}^k - \widetilde{\mathbf{w}}_{h,l}^k) \leqslant \sqrt{d}\kappa_l, \left|\|\boldsymbol{\phi}(s,a)_{(\mathbf{U}_{h,l}^k)^{-1}} - \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}_{h,l}^{k,-1}}\right| \leqslant \sqrt{d\kappa_l},$$

where κ_l is used to quantify the vector $\mathbf{w}_{h,l}^k$ and inverse matrix $(\mathbf{U}_{h,l}^l)^{-1}$.

Proof of Lemma 5.9.15. From Lemma 5.9.31 we have

$$\left|\left\langle \boldsymbol{\phi}(s,a), \mathbf{w}_{h,l}^k \right\rangle - \left\langle \boldsymbol{\phi}(s,a), \widetilde{\mathbf{w}}_{h,l}^k \right\rangle\right| \leqslant \sqrt{d}\kappa_l \leqslant 0.01 \cdot 2^{-4l}$$

where the first inequality is due to Lemma 5.9.31, and the second inequality is valid due to $\kappa_l = 0.01 \cdot 2^{-4l}$. Similarly, we have

$$\left|\|\boldsymbol{\phi}(s,a)\|_{(\mathbf{U}_{h,l}^k)^{-1}} - \|\boldsymbol{\phi}(s,a)\|_{\widetilde{\mathbf{U}}_{h,l}^{k,-1}}\right| \leqslant \sqrt{d\kappa_l} \leqslant 0.1 \cdot 2^{-2l}.$$

□

5.9.3.2 Proof of Lemma 5.9.16

Proof of Lemma 5.9.16. First, both $l_h^\tau(s_h^\tau) = l$ and $f_h^\tau(s_h^\tau) = 1$ held for any $\tau \in \mathcal{C}_{h,l}^k$. This implies that the criteria for either Line 7 or Line 9 holds as $l = l_h^\tau(s_h^\tau)$. For τ that satisfies the first criterion, we have $l_h^\tau(s_h^\tau) > L_\tau$. Note that $L_\tau = \max\{\lceil \log_4(\tau/d) \rceil, 0\}$, so this only happens for $\tau < 4^l d$. For other τ that satisfies the second criterion, we have that

$$\|\boldsymbol{\phi}_h^\tau\|_{(\mathbf{U}_{h,l}^\tau)^{-1}} \geqslant \|\boldsymbol{\phi}_h^\tau\|_{\widetilde{\mathbf{U}}_{h,l}^{\tau,-1}} - \left|\|\boldsymbol{\phi}_h^\tau\|_{\widetilde{\mathbf{U}}_{h,l}^{\tau,-1}} - \|\boldsymbol{\phi}_h^\tau\|_{(\mathbf{U}_{h,l}^\tau)^{-1}}\right| \geqslant 2^{-l}\gamma_l^{-1} - 0.1 \cdot 2^{-l}\gamma_l^{-1} = 0.9 \cdot 2^{-l}\gamma_l^{-1},$$

where the first inequality holds due to the triangle inequality. In the second inequality, the first term $\|\boldsymbol{\phi}_h^\tau\|_{\widetilde{\mathbf{U}}_{h,l}^{\tau,-1}}$ is bounded by criterion in Line 9 while the second term $\big|\|\boldsymbol{\phi}_h^\tau\|_{\widetilde{\mathbf{U}}_{h,l}^{\tau,-1}} - \|\boldsymbol{\phi}_h^\tau\|_{(\mathbf{U}_{h,l}^\tau)^{-1}}\big|$ follows from Lemma 5.9.15.

Arrange elements of $\mathcal{C}_{h,l}^k$ in ascending order as $\{\tau_i\}_i$. According to the above reasoning, the number of elements $\tau \in \mathcal{C}_{h,l}^k$ that $\|\boldsymbol{\phi}_h^\tau\|_{(\mathbf{U}_{h,l}^\tau)^{-1}} \geqslant 0.9 \cdot 2^{-l}\gamma_l^{-1}$ is at least $|\mathcal{C}_{h,l}^k| - 4^l d$. This gives

$$\sum_{i=1}^{|\mathcal{C}_{h,l}^k|} \min\{1, \|\boldsymbol{\phi}_h^\tau\|^2_{(\mathbf{U}_{h,l}^\tau)^{-1}}\} \geqslant (0.9 \cdot 2^{-l}\gamma_l^{-1})^2 \cdot (|\mathcal{C}_{h,l}^k| - 4^l d). \tag{5.9.24}$$

On the other hand, Lemma 2.8.15 upper bounds the LHS of (5.9.24) by

$$\sum_{i=1}^{|\mathcal{C}_{h,l}^k|} \min\{1, \|\boldsymbol{\phi}_h^\tau\|^2_{(\mathbf{U}_{h,l}^\tau)^{-1}}\} \leqslant 2d \ln\big(1 + |\mathcal{C}_{h,l}^k|/(d\lambda)\big). \tag{5.9.25}$$

Combining (5.9.24) and (5.9.25) gives

$$0.81 \cdot 4^{-l}\gamma_l^{-2}(|\mathcal{C}_{h,l}^k| - 4^l d) \leqslant 2d \ln\big(1 + |\mathcal{C}_{h,l}^k|/(16d)\big). \tag{5.9.26}$$

From algebra analysis in Lemma 5.9.37, a necessary condition for (5.9.26) is $|\mathcal{C}_{h,l}^k| \leqslant 16l \cdot 4^l\gamma_l^2 d$. □

5.9.3.3 Proof of Lemma 5.9.17

We first present a claim from Vial et al. (2022) controlling the infinite norm of coefficient $\mathbf{w}$.

Lemma 5.9.32 (Claim 10, Vial et al. 2022)**.** For any $(k, h, l) \in [K] \times [H] \times \mathbb{N}^+$, we have $\|\mathbf{w}_{h,l}^k\|_\infty \leqslant \|\mathbf{w}_{h,l}^k\|_2 \leqslant (2^l dH)^4$.

Proof of Lemma 5.9.17. Denote $\mathcal{X}_\ell$ as the set of all $\widetilde{\mathbf{w}}_{h,\ell}^k$ and $\mathcal{Y}_\ell$ as the set of all $\widetilde{\mathbf{U}}_{h,\ell}^{k,-1}$. From the definition of $\mathcal{V}_{h,l}^k$, we have that $|\mathcal{V}_{h,l}^k| \leqslant \prod_{\ell=1}^l \big(|\mathcal{X}_\ell| \cdot |\mathcal{Y}_\ell|\big)$. From Lemma 5.9.32, we have $\|\mathbf{w}_{h,\ell}^k\|_\infty \leqslant (2^\ell dH)^4$. Note that $\mathbf{w}_{h,\ell}^k \in \mathbb{R}^d$, we have the number of different $\widetilde{\mathbf{w}}_{h,\ell}^k$ controlled by

$$|\mathcal{X}_\ell| \leqslant (1 + 2 \cdot (2^\ell dH)^4/\kappa_\ell)^d \leqslant (2 \cdot (2^\ell dH)^4 \cdot 2^{6+4\ell} d)^d \leqslant 2^{(7+8\ell)d} d^{5d} H^{4d}.$$

In addition, we have $\|(\mathbf{U}_{h,l}^k)^{-1}\|_\infty \leqslant 1/\lambda = 1/16$. So we can bound the number of $\widetilde{\mathbf{U}}_{h,\ell}^{k,-1}$ by

$$|\mathcal{Y}_\ell| \leqslant (1 + 2 \cdot 1/(16\kappa_\ell))^{d^2} \leqslant (2 \cdot 2^{2+4\ell} d)^{d^2} \leqslant 2^{(3+4\ell)d^2} d^{d^2}.$$

As a result, we can conclude that

$$|\mathcal{V}_{h,l}^k| \leqslant \prod_{\ell=1}^{l} \big(|\mathcal{X}_\ell| \cdot |\mathcal{Y}_\ell|\big) \leqslant \prod_{\ell=1}^{l} \big(2^{(7+8\ell)d} d^{5d} H^{4d} \cdot 2^{(3+4\ell)d^2} d^{d^2}\big) \leqslant (2^{22} d^5 H^4)^{l^2 d^2}.$$

□

5.9.3.4 Proof of Lemma 5.9.18

Proof of Lemma 5.9.18. According to Proposition 5.3.2, there exists a parameter $\mathbf{w}_h$ such that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, it holds that $\big|\langle \boldsymbol{\phi}(s,a), \mathbf{w}_h \rangle - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a)\big| \leqslant 2H\zeta$. Denoting $\eta_h^\tau = \langle \boldsymbol{\phi}_h^\tau, \mathbf{w}_h \rangle - [\mathbb{B}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau)$ and $\varepsilon_h^\tau = \big(\widehat{V}_{h+1}^k(s_{h+1}^\tau) - [\mathbb{P}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau)\big)$, we have

$$\begin{aligned}
\mathbf{U}_{h,l}^k(\mathbf{w}_{h,l}^k - \mathbf{w}_h) &= \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \Big(r_h^\tau + \widehat{V}_{h+1}^k(s_{h+1}^\tau)\Big) - \Big(\lambda \mathbf{I} + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau (\boldsymbol{\phi}_h^\tau)^\top\Big)\mathbf{w}_h \\
&= -\lambda \mathbf{w}_h + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \Big(r_h^\tau + \widehat{V}_{h+1}^k(s_{h+1}^\tau) - \langle \boldsymbol{\phi}_h^\tau, \mathbf{w}_h \rangle\Big) \\
&= -\lambda \mathbf{w}_h + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \Big(r_h^\tau + \widehat{V}_{h+1}^k(s_{h+1}^\tau) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau)\Big) + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau \\
&= -\lambda \mathbf{w}_h + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \varepsilon_h^\tau + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau, \qquad (5.9.27)
\end{aligned}$$

where the first equality holds due to the definition of $\mathbf{U}_{h,l}^k, \mathbf{w}_{h,l}^k$, the second equality holds by rearranging the terms, the third equality holds according the definition of η_h^τ, and the last equality holds from the relationship that $[\mathbb{B}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau) = r_h^\tau + [\mathbb{P}_h \widehat{V}_{h+1}^k](s_h^\tau, a_h^\tau)$. Therefore,

for any vector $\boldsymbol{\phi} \in \mathbb{R}^d$, it holds that

$$\begin{aligned}\left|\left\langle \boldsymbol{\phi}, \mathbf{w}_{h,l}^k - \mathbf{w}_h \right\rangle\right| &= \left|\boldsymbol{\phi}^\top \left(\mathbf{U}_{h,l}^k\right)^{-1} \mathbf{U}_{h,l}^k (\mathbf{w}_{h,l}^k - \mathbf{w}_h)\right| \\ &= \left|\boldsymbol{\phi}^\top (\mathbf{U}_{h,l}^k)^{-1} \cdot \left(-\lambda \mathbf{w}_h + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \varepsilon_h^\tau + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau \right)\right| \\ &\leqslant \|\boldsymbol{\phi}\|_{(\mathbf{U}_{h,l}^k)^{-1}} \left\| -\lambda \mathbf{w}_h + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \varepsilon_h^\tau + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}}, \end{aligned} \tag{5.9.28}$$

where the second equality follows (5.9.27) and the inequality holds from Cauchy–Schwarz inequality (i.e., $|\mathbf{x}^\top \mathbf{U} \mathbf{y}| \leqslant \|\mathbf{x}\|_{\mathbf{U}} \|\mathbf{y}\|_{\mathbf{U}}$). From the triangle inequality, we have

$$\begin{aligned}&\left\| -\lambda \mathbf{w}_h + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \varepsilon_h^\tau + \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \\ &\leqslant \lambda \|\mathbf{w}_h\|_{(\mathbf{U}_{h,l}^k)^{-1}} + \left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \varepsilon_h^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} + \left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}}. \end{aligned} \tag{5.9.29}$$

There are three terms which we will bound respectively. For the first term, we have

$$\lambda \|\mathbf{w}_h\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 2\sqrt{d\lambda} H \leqslant 0.1 \gamma_l, \tag{5.9.30}$$

where the first inequality holds due to the fact that $\|\mathbf{w}_h\|_2 \leqslant 2H\sqrt{d}$ as of Proposition 5.3.2 and the fact that $\mathbf{U}_{h,l}^k \succeq \lambda \mathbf{I}$. Under the good event $\mathcal{G}_1$ and Lemma 5.9.5, the second term can be bounded by the following:

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \varepsilon_h^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 1.1 \gamma_l. \tag{5.9.31}$$

And the last term can be bounded by:

$$\left\| \sum_{\tau \in \mathcal{C}_{h,l}^{k-1}} \boldsymbol{\phi}_h^\tau \eta_h^\tau \right\|_{(\mathbf{U}_{h,l}^k)^{-1}} \leqslant 2H\zeta \sqrt{|\mathcal{C}_{h,l}^k|} \leqslant 2H\zeta \sqrt{16l \cdot 4^l \gamma_l^2 d} = 8\sqrt{ld} H \cdot 2^l \gamma_l \zeta, \tag{5.9.32}$$

where the first inequality is due to Lemma 4.9.10, and the second inequality follows from Lemma 5.9.16. Plugging (5.9.29), (5.9.30), (5.9.31), and (5.9.32) into (5.9.28) gives

$$\left|\left\langle \boldsymbol{\phi}, \mathbf{w}_{h,l}^k - \mathbf{w}_h \right\rangle\right| \leqslant \left(1.2\gamma_l + 8\sqrt{ld} H \cdot 2^l \gamma_l \zeta\right) \|\boldsymbol{\phi}\|_{(\mathbf{U}_{h,l}^k)^{-1}}. \tag{5.9.33}$$

So for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\begin{aligned}
&\big|Q_{h,l}^k(s,a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a)\big| = \big|\big\langle \boldsymbol{\phi}(s,a), \widetilde{\mathbf{w}}_{h,l}^k \big\rangle - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a)\big| \\
&\leqslant \big|\big\langle \boldsymbol{\phi}(s,a), \widetilde{\mathbf{w}}_{h,l}^k - \mathbf{w}_{h,l}^k \big\rangle\big| + \big|\big\langle \boldsymbol{\phi}(s,a), \mathbf{w}_{h,l}^k - \mathbf{w}_h \big\rangle\big| + \big|\big\langle \boldsymbol{\phi}(s,a), \mathbf{w}_h \big\rangle - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a)\big| \\
&\leqslant 0.01 \cdot 2^{-4l} + \big(1.2 + 8\sqrt{ld}H \cdot 2^l \zeta\big)\gamma_l \|\boldsymbol{\phi}(s,a)\|_{(\mathbf{U}_{h,l}^k)^{-1}} + 2H\zeta. \qquad (5.9.34)
\end{aligned}$$

where the first inequality holds from the triangle inequality, and there are three terms in the second inequality which we will bound them respectively: the first term is given by Lemma 5.9.15, the second term follows (5.9.33), and the third term holds from the definition of $\mathbf{w}_h$. □

5.9.3.5 Proof of Lemma 5.9.19

Proof of Lemma 5.9.19. We prove by doing case analysis. In case that action $a_l \in \mathcal{A}_{h,l+1}^k(s)$, we can assign $a_{l+1} = a_l \in \mathcal{A}_{h,l+1}^k(s)$ so that

$$[\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_l) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_{l+1}) = 0. \qquad (5.9.35)$$

On the other hand, in the case that $a_l \notin \mathcal{A}_{h,l+1}^k(s)$, the action a_l is eliminated with $Q_{h,l}^k(s,a_l) < V_{h,l}^k(s) - 4 \cdot 2^{-l}$. Note in this case, there exists $a_{l+1} = \pi_{h,l}^k(s) \in \mathcal{A}_{h,l+1}^k(s)$ such that

$$Q_{h,l}^k(s,a_l) + 4 \cdot 2^{-l} < V_{h,l}^k(s) = Q_{h,l}^k(s,a_{l+1}). \qquad (5.9.36)$$

According to Lemma 5.9.6 and the condition that $l \leqslant L_0$, we have that empirical state-value function $Q_{h,l}^k(s,\cdot)$ is a good estimation for $[\mathbb{B}_h \widehat{V}_{h+1}^k](s,\cdot)$ on actions $a_l, a_{l+1} \in \mathcal{A}_l^k(s)$ under event $\mathcal{G}_1$:

$$\big|[\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_l) - Q_{h,l}^k(s,a_l)\big| \leqslant 2 \cdot 2^{-l} + \chi\sqrt{L_0}\zeta \qquad (5.9.37)$$

$$\big|[\mathbb{B}_h \widehat{V}_{h+1}^k](s,a_{l+1}) - Q_{h,l}^k(s,a_{l+1})\big| \leqslant 2 \cdot 2^{-l} + \chi\sqrt{L_0}\zeta. \qquad (5.9.38)$$

Moreover,

$$
\begin{aligned}
&[\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_l) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_{l+1}) \\
&= \big([\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_l) - Q_{h,l}^k(s, a_l)\big) \\
&\quad + \big(Q_{h,l}^k(s, a_l) - Q_{h,l}^k(s, a_{l+1})\big) + \big(Q_{h,l}^k(s, a_{l+1}) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_{l+1})\big) \\
&\leqslant 2 \cdot \big(2 \cdot 2^{-l} + \chi\sqrt{L_0}\zeta\big) - 4 \cdot 2^{-l} \\
&= 2\chi\sqrt{L_0}\zeta. \qquad (5.9.39)
\end{aligned}
$$

where the first inequality is derived from combining (5.9.36), (5.9.37), and (5.9.38). So from (5.9.35) and (5.9.39), we have that $[\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_l) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_{l+1}) \leqslant 2\chi\sqrt{L_0}\zeta$ holds in both cases. □

5.9.3.6 Proof of Lemma 5.9.20

Proof of Lemma 5.9.20. We prove by induction on l. The induction basis holds at $l = 0$ by selecting $a_1 = \operatorname{argmax}_{a \in \mathcal{A}}[\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) \in \mathcal{A}$ which ensures $\max_{a \in \mathcal{A}}[\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_1) = 0$. Additionally, if the induction hypothesis holds for $l - 1$, we have that

$$
\begin{aligned}
&\max_{a \in \mathcal{A}}[\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_{l+1}) \\
&= \big(\max_{a \in \mathcal{A}}[\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_l)\big) + \big([\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_l) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a_{l+1})\big) \\
&\leqslant 2(l-1)\chi\sqrt{L_0}\zeta + 2\chi\sqrt{L_0}\zeta \\
&= 2l \cdot \chi\sqrt{L_0}\zeta,
\end{aligned}
$$

where the first inequality term is due to combining induction hypothesis with Lemma 5.9.19. We can then reach desired statement holds for all l in the range by induction. □

5.9.3.7 Proof of Lemma 5.9.21

Proof of Lemma 5.9.21. According to Lemma 5.9.20, there exists some action $a_l \in \mathcal{A}_{h,l}^k(s)$ that

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,a_l) \leqslant 2(l-1)\chi\sqrt{L_0}\zeta. \tag{5.9.40}$$

Moreover, we have

$$[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a_l) - V_{h,l}^k(s) \leqslant [\mathbb{B}_h\widehat{V}_{h+1}^k](s,a_l) - Q_{h,l}^k(s,a_l) \leqslant 2\cdot 2^{-l} + \chi\sqrt{L_0}\zeta, \tag{5.9.41}$$

where the first inequality comes from the definition $V_{h,l}^k(s) = \max_{a\in\mathcal{A}_{h,l}^k} Q_{h,l}^k(s,a)$ and the second inequality holds according to Lemma 5.9.6 with $l \leqslant L_0$. Adding up (5.9.40) and (5.9.41) leads to

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - V_{h,l}^k \leqslant 2\cdot 2^{-l} + (2l-1)\chi\sqrt{L_0}\zeta.$$

This completes the proof. □

5.9.3.8 Proof of Lemma 5.9.22

Proof of Lemma 5.9.22. The statement holds by simply checking:

$$\begin{aligned} V_{h,l}^k(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) &\leqslant V_{h,l}^k(s) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,\pi_{h,l}^k(s)) \\ &= Q_{h,l}^k(s,\pi_{h,l}^k(s)) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,\pi_{h,l}^k(s)) \\ &\leqslant 2\cdot 2^{-l} + \chi\sqrt{L_0}\zeta, \end{aligned}$$

where the first inequality holds from $\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) \geqslant [\mathbb{B}_h\widehat{V}_{h+1}^k](s,\pi_{h,l}^k(s))$, the equality is from the definition $V_{h,l}^k(s) = Q_{h,l}^k(s,\pi_{h,l}^k(s))$, and the last inequality holds according to Lemma 5.9.6 with the condition $l \leqslant L_0$. □

5.9.3.9 Proof of Lemma 5.9.23

Proof of Lemma 5.9.23. The statement holds by checking

$$\begin{aligned}
&\min\left\{V_{h,l}^k(s)+3\cdot 2^{-l},\widehat{V}_{h,l-1}^k(s)\right\}-\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)\\
=&\min_{\ell=1}^{l}\{V_{h,\ell}^k(s)+3\cdot 2^{-\ell}\}-\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)\\
\geqslant&\min_{\ell=1}^{l}\{3\cdot 2^{-\ell}-(2\cdot 2^{-l}+(2\ell-1)\chi\sqrt{L_0}\zeta)\}\\
=&\,2^{-l}-(2l-1)\chi\sqrt{L_0}\zeta,
\end{aligned}$$

where the first equality holds due to $\widehat{V}_{h,l}^k(s)=\min_{\ell=1}^{l}\{V_{h,\ell}^k(s)+3\cdot 2^{-\ell}\}$, the inequality holds according to Lemma 5.9.21, and the last equality holds since 2^{-l} decreases as l increases. □

5.9.3.10 Proof of Lemma 5.9.24

Proof of Lemma 5.9.24. The statement holds by checking

$$\begin{aligned}
&\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)-\max\left\{V_{h,l}^k(s)-3\cdot 2^{-l},\check{V}_{h,l-1}^k(s)\right\}\\
=&\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)-\max_{\ell=1}^{l}\{V_{h,\ell}^k(s)-3\cdot 2^{-\ell}\}\\
=&\min_{\ell=1}^{l}\left\{\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a)-V_{h,\ell}^k(s)+3\cdot 2^{-\ell}\right\}\\
\geqslant&\min_{\ell=1}^{l}\{-(2\cdot 2^{-l}+\chi\sqrt{L_0}\zeta)+3\cdot 2^{-\ell}\}\\
=&\,2^{-l}-\chi\sqrt{L_0}\zeta,
\end{aligned}$$

where the first equality holds due to the design of Algorithm 13 which would guarantee that $\check{V}_{h,l}^k(s)=\max_{\ell=1}^{l}\{V_{h,\ell}^k(s)-3\cdot 2^{-\ell}\}$, the inequality holds according to Lemma 5.9.22, and the last equality holds since 2^{-l} decreases as l increases. □

5.9.3.11 Proof of Lemma 5.9.25

We prove Lemma 5.9.25 in this subsection. The first lemma which we introduce establishes an upper bound on the underestimation of the state value function $\widehat{V}_h^k$ for every action and

every state through a categorised discussion based on whether Algorithm 13 reaches phase L_ε for state s. Specifically, if the process does not reach phase L_ε, we can substantiate the statement by applying Lemma 5.9.23 to phase $l_h^k(s) - 1$. Conversely, if the process reaches phase L_ε, the statement can be proven by applying Lemma 5.9.21 to phase L_ε.

Lemma 5.9.33. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $(k, h, s) \in [K] \times [H] \times \mathcal{S}$,

$$\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - \widehat{V}_h^k(s) \leqslant 0.07\varepsilon/H.$$

Now we are ready to prove Lemma 5.9.25 by induction.

Proof of Lemma 5.9.25. We prove by induction on stage $h \in [H]$. It is sufficient to show for any $h \in [H], s \in \mathcal{S}$,

$$V_h^*(s) - \widehat{V}_h^k(s) \leqslant 0.07\varepsilon \cdot (H + 1 - h)/H. \tag{5.9.42}$$

We use induction on h from $H + 1$ to 1 to prove the statement. The induction basis holds from the definition that $V_{H+1}^*(s) = \widehat{V}_{H+1}^k(s) = 0$. Assume the induction hypothesis (5.9.42) holds for $h + 1$, we have

$$\begin{aligned}
\max_{a \in \mathcal{A}} [\mathbb{B}_h V_{h+1}^*](s, a) - \max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) &\leqslant \max_{a \in \mathcal{A}} [\mathbb{B}_h (V_{h+1}^* - \widehat{V}_{h+1}^k)](s, a) \\
&\leqslant \max_{s' \in \mathcal{S}} \left(V_{h+1}^*(s') - \widehat{V}_{h+1}^k(s') \right) \\
&\leqslant 0.07\varepsilon \cdot (H - h)/H. \qquad (5.9.43)
\end{aligned}$$

So for level h, it holds that

$$\begin{aligned}
& V_h^*(s) - \widehat{V}_h^k(s) \\
= & \left(\max_{a \in \mathcal{A}} [\mathbb{B}_h V_{h+1}^*](s, a) - \max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) \right) + \left(\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - \widehat{V}_h^k(s) \right) \\
\leqslant & 0.07\varepsilon \cdot (H - h)/H + 0.07\varepsilon/H \leqslant 0.07\varepsilon \cdot (H + 1 - h)/H,
\end{aligned}$$

where the first inequality holds by combining (5.9.43) with Lemma 5.9.33. This proves the induction statement (5.9.42) for h, which leads to the desired statement. □

5.9.3.12 Proof of Lemma 5.9.26

We prove Lemma 5.9.26 in this subsection, the first lemma we use establishes an upper bound on the overestimation of the state value function $\widehat{V}_h^k$ for the executed policy $\pi_h^k(s)$ across all states.

Lemma 5.9.34. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $(k, h, s) \in [K] \times [H] \times \mathcal{S}$,

$$\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, \pi_h^k(s)) \leqslant 16 \cdot 2^{-l_h^k(s)} + 0.10\varepsilon/H.$$

Then the following lemma establishes an upper bound on the decision error induced by the arm-elimination process with respect to the state-action value function given by the ground-truth transform.

Lemma 5.9.35. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $(k, h, s) \in [K] \times [H] \times \mathcal{S}$,

$$\widehat{V}_h^k(s) - \max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) \leqslant 10 \cdot 2^{-l_h^k(s)} + 0.06\varepsilon/H.$$

Proof of Lemma 5.9.26. We can directly reach the desired result by taking summation on Lemma 5.9.34 and Lemma 5.9.35:

$$\begin{aligned}
&\widehat{V}_h^k(s) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, \pi_h^k(s)) \\
&\leqslant \big(\max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s, \pi_h^k(s)) \big) + \big(\widehat{V}_h^k(s) - \max_{a \in \mathcal{A}} [\mathbb{B}_h \widehat{V}_{h+1}^k](s, a) \big) \\
&\leqslant \big(16 \cdot 2^{-l_h^k(s)} + 0.10\varepsilon/H \big) + \big(10 \cdot 2^{-l_h^k(s)} + 0.06\varepsilon/H \big) \\
&= 26 \cdot 2^{-l_h^k(s)} + 0.16\varepsilon/H.
\end{aligned}$$

□

5.9.3.13 Proof of Lemma 5.9.28

We can prove the statement by applying a union bound to the concentration event, as given by the Azuma-Hoeffding inequality.

Proof of Lemma 5.9.28. Consider some fixed $h \in [H]$ and $\varepsilon = 2^{-l} > 0$. List the episodes index k such that $V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) > \varepsilon$ holds in ascending order as $\{\tau_i\}_i$. Recall that

$$\eta_h^{\tau_i} = [\mathbb{P}_h(\widehat{V}_{h+1}^{\tau_i} - V_{h+1}^{\pi^{\tau_i}})](s_h^{\tau_i}, \pi_h^{\tau_i}(s_h^{\tau_i})) - \big(\widehat{V}_{h+1}^{\tau_i}(s_{h+1}^{\tau_i}) - V_{h+1}^{\pi^{\tau_i}}(s_{h+1}^{\tau_i})\big).$$

Since the environment sample $s_{h'+1}^{\tau_i}$ according to $\mathbb{P}_{h'}(\cdot|s_{h'}^{\tau_i}, a_{h'}^{\tau_i})$, $\eta_{h'}^{\tau_i}$ is $\mathcal{F}_{h'+1}^{\tau_i}$-measurable with $\mathbb{E}\big[\eta_{h'}^{\tau_i}\big|\mathcal{F}_{h'}^{\tau_i}\big] = 0$. Since both $0 \leqslant \widehat{V}_{h'+1}^{\tau_i}(s_{h'+1}^{\tau_i}) \leqslant H$ and $0 \leqslant V_{h'+1}^{\pi^{\tau_i}}(s_{h'+1}^{\tau_i}) \leqslant H$ hold, we have $|\eta_{h'}^{\tau_i}| \leqslant 2H$. According to Lemma 2.8.16 over filtration

$$\mathcal{F}_h^{\tau_1} \subseteq \mathcal{F}_{h+1}^{\tau_1} \subseteq \cdots \subseteq \mathcal{F}_H^{\tau_1} \subseteq \mathcal{F}_h^{\tau_2} \subseteq \mathcal{F}_{h+1}^{\tau_2} \subseteq \cdots \subseteq \mathcal{F}_H^{\tau_2} \subseteq \cdots \subseteq \mathcal{F}_{h'}^{\tau_i} \subseteq \cdots$$

for some fixed $S = |\mathcal{K}_h^\varepsilon|$, the good event that

$$\sum_{i=1}^{|\mathcal{K}_h^\varepsilon|} \sum_{h'=h}^{H} \eta_{h'}^{\tau_i} \leqslant 2H\sqrt{2HS\log(4HS^2l^2/\delta)} = 4\sqrt{H^3|\mathcal{K}_h^\varepsilon|\log(4H|\mathcal{K}_h^\varepsilon|\log(\varepsilon^{-1})/\delta)}$$

happens with probability at least $1 - \delta/(4HS^2l^2)$. By the union bound statement over all $(h, S, l) \in [H] \times [K] \times \mathbb{N}^+$, we have the bad event happens with probability at most

$$\Pr[\mathcal{G}_2^\complement] \leqslant \sum_{h=1}^{H}\sum_{S=1}^{K}\sum_{l=1}^{\infty} \Pr[\mathcal{B}_2(h, 2^{-l})] \leqslant \sum_{h=1}^{H}\sum_{S=1}^{K}\sum_{l=1}^{\infty} \frac{\delta}{4HS^2l^2} \leqslant \delta,$$

where the last inequality holds from $\sum_{n\geqslant 1} n^{-2} = \pi^2/6$, which reach the desired statement. □

5.9.3.14 Proof of Lemma 5.9.29

We first provide the following instantaneous regret upper bound by combining Lemma 5.9.25 and Lemma 5.9.26.

Lemma 5.9.36. Under event $\mathcal{G}_1$ and for all $\varepsilon > 0$ that $\mathcal{G}_\varepsilon$ is satisfied, for any $(k, h) \in [K] \times [H]$,

$$V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) \leqslant 0.23\varepsilon + 26\sum_{h'=h}^{H} 2^{-l_{h'}^k(s_{h'}^k)} + \sum_{h'=h}^{H} \eta_{h'}^k,$$

where $\eta_h^k = [\mathbb{P}_h(\widehat{V}_{h+1}^k - V_{h+1}^{\pi^k})](s_h^k, \pi_h^k(s_h^k)) - \big(\widehat{V}_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi^k}(s_{h+1}^k)\big)$ is a $\mathcal{F}_{h+1}^k$-measurable random variable that $\mathbb{E}[\eta_h^k|\mathcal{F}_h^k] = 0$ and $|\eta_h^k| \leqslant H$.

Together with Lemma 5.9.9 and the definition of $\mathcal{G}_2$, we can provide an upper bound for arbitrary subsets.

Proof of Lemma 5.9.29. Taking summation on result given by Lemma 5.9.36 to all $k \in \mathcal{K}$ gives

$$\sum_{k\in\mathcal{K}} \left(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\right) \leqslant 0.23|\mathcal{K}|\varepsilon + 26\sum_{k\in\mathcal{K}}\sum_{h'=h}^{H} 2^{-l_{h'}^k(s_{h'}^k)} + \sum_{k\in\mathcal{K}}\sum_{h'=h}^{H}\eta_{h'}^k. \tag{5.9.44}$$

We can bound the second term according to Lemma 5.9.9,

$$26\sum_{k\in\mathcal{K}}\sum_{h'=h}^{H} 2^{-l_{h'}^k(s_{h'}^k)} \leqslant 0.26|\mathcal{K}|\varepsilon + 2^{17}L_\varepsilon dH^2\gamma_{L_\varepsilon}^2\varepsilon^{-1}. \tag{5.9.45}$$

Under event $\mathcal{G}_2$, the third term satisfies that

$$\sum_{k\in\mathcal{K}}\sum_{h'=h}^{H}\eta_{h'}^k \leqslant 4\sqrt{H^3|\mathcal{K}|\log(4H|\mathcal{K}|\log(\varepsilon^{-1})/\delta)}. \tag{5.9.46}$$

Plugging (5.9.45) and (5.9.46) into (5.9.44) gives

$$\sum_{k\in\mathcal{K}} \left(V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k)\right) \leqslant 0.49|\mathcal{K}|\varepsilon + 2^{17}L_\varepsilon dH^2\gamma_{L_\varepsilon}^2\varepsilon^{-1} + 4\sqrt{H^3|\mathcal{K}|\log(4H|\mathcal{K}|\log(\varepsilon^{-1})/\delta)}. \tag{5.9.47}$$

□

5.9.4 Proof of Lemmas in Section 5.9.3

5.9.4.1 Proof of Lemma 5.9.33

Proof of Lemma 5.9.33. We start the proof by discussing different cases. First, if $l_h^k(s) \leqslant L_\varepsilon$, we have $l_h^k(s) - 1 \leqslant \min\{L_\varepsilon, l_h^k(s) - 1\}$, according to the definition of $\widehat{V}_{h,l}^k(s)$,

$$\begin{aligned}
\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \widehat{V}_h^k(s) &= \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \widehat{V}_{h,l_h^k(s)-1}^k(s) \\
&\leqslant -2^{-(l_h^k(s)-1)} + 2(l_h^k(s)-1)\chi\sqrt{L_\varepsilon}\zeta \\
&\leqslant 0 + 2\chi L_\varepsilon^{1.5}\zeta \\
&\leqslant 0.02\varepsilon/H,
\end{aligned} \tag{5.9.48}$$

where the first inequality holds from Lemma 5.9.23, and the last inequality holds due to $\chi L_\varepsilon^{1.5}\zeta \leqslant 2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$ given by $\mathcal{G}_\varepsilon$.

On the other hand, when $l_h^k(s) > L_\varepsilon$, we have $L_\varepsilon \leqslant \min\{L_\varepsilon, l_h^k(s)-1\}$ and thus

$$\widehat{V}_h^k(s) \geqslant \check{V}_{h,L_\varepsilon}^k(s) \geqslant V_{h,L_\varepsilon}^k(s) - 3\cdot 2^{-L_\varepsilon} \tag{5.9.49}$$

where the first inequality is due to Lemma 5.9.2 and the second inequality holds due to the definition of $\check{V}_{h,L_\varepsilon}^k(s)$. Therefore, $L_\varepsilon \leqslant \min\{L_\varepsilon, l_h^k(s)-1\}$ yields

$$\begin{aligned}
\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \widehat{V}_h^k(s) &\leqslant \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - V_{h,L_\varepsilon}^k(s) + 3\cdot 2^{-L_\varepsilon} \\
&\leqslant 5\cdot 2^{-L_\varepsilon} + (2L_\varepsilon - 1)\chi\sqrt{L_\varepsilon}\zeta \\
&\leqslant 0.05\varepsilon/H + 0.02\varepsilon/H = 0.07\varepsilon/H,
\end{aligned} \tag{5.9.50}$$

where the first inequality is given by (5.9.49), the second inequality is given by Lemma 5.9.21, and the last inequality holds from $\chi L_\varepsilon^{1.5}\zeta \leqslant 2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$ given by $\mathcal{G}_\varepsilon$. So considering both (5.9.48) and (5.9.50), we have the first statement

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - \widehat{V}_h^k(s) \leqslant 0.07\varepsilon/H$$

always holds under event $\mathcal{G}_1$. □

5.9.4.2 Proof of Lemma 5.9.34

We prove Lemma 5.9.34 by applying Lemma 5.9.7 on phase $\min\{L_\varepsilon, l_h^k(s)-1\}$, in this subsection.

Proof of Lemma 5.9.34. Note we have $\pi_{h,l_h^k(s)-1}^k(s) \in \mathcal{A}_{h,l_h^k(s)}^k(s)$ according to the definition of $\mathcal{A}_{h,l+1}^k(s)$. This implies $\pi_h^k(s) \in \mathcal{A}_{h,l_h^k(s)}^k(s)$ during the elimination process.

If $l_h^k(s) \leqslant L_\varepsilon$, we have $l_h^k(s) - 1 \leqslant \min\{L_\varepsilon, l_h^k(s) - 1\}$. Thus,

$$\begin{aligned}\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,\pi_h^k(s)) &\leqslant 8\cdot 2^{-(l_h^k(s)-1)} + 2l_h^k(s)\cdot\chi\sqrt{L_\varepsilon}\zeta \\ &\leqslant 16\cdot 2^{-l_h^k(s)} + 2\chi L_\varepsilon^{1.5}\zeta \\ &\leqslant 16\cdot 2^{-l_h^k(s)} + 0.02\varepsilon/H, \end{aligned} \tag{5.9.51}$$

where the first inequality follows from Lemma 5.9.7 with $\pi_h^k(s) \in \mathcal{A}_{h,l_h^k(s)}^k(s)$ and the last inequality holds due to $\chi L_\varepsilon^{1.5}\zeta \leqslant 0.01\varepsilon/H$ given by $\mathcal{G}_\varepsilon$.

Otherwise, we have $L_\varepsilon \leqslant \min\{L_\varepsilon, l_h^k(s) - 1\}$. In this case, we have

$$\begin{aligned}\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,\pi_h^k(s)) &\leqslant 8\cdot 2^{-L_\varepsilon} + 2\chi L_\varepsilon^{1.5}\zeta \\ &\leqslant 0.08\varepsilon/H + 0.02\varepsilon/H = 0.10\varepsilon/H, \end{aligned} \tag{5.9.52}$$

where the first inequality follows from Lemma 5.9.7 with $\pi_h^k(s) \in \mathcal{A}_{h,l_h^k(s)}^k(s) \subseteq \mathcal{A}_{h,L_\varepsilon}^k(s)$ according to the elimination routine and the final inequality holds due to $\chi L_\varepsilon^{1.5}\zeta \leqslant 2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$ given by $\mathcal{G}_\varepsilon$. So by combining (5.9.51) and (5.9.52), we have the desired statement that

$$\max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}_{h+1}^k](s,a) - [\mathbb{B}_h\widehat{V}_{h+1}^k](s,\pi_h^k(s)) \leqslant 16\cdot 2^{-l_h^k(s)} + 0.10\varepsilon/H.$$

□

5.9.4.3 Proof of Lemma 5.9.35

We prove Lemma 5.9.35 in this section by applying Lemma 5.9.22 on phase $\min\{L_\varepsilon, l_h^k(s)-1\}$.

Proof of Lemma 5.9.35. If $l_h^k(s) \leqslant L_\varepsilon$, we have $l_h^k(s) - 1 \leqslant \min\{L_\varepsilon, l_h^k(s) - 1\}$. Firstly, we have

$$\widehat{V}_h^k(s) \leqslant \widehat{V}_{h,l_h^k(s)-1}^k(s) \leqslant V_{h,l_h^k(s)-1}^k(s) + 3\cdot 2^{-(l_h^k(s)-1)}. \tag{5.9.53}$$

where the first inequality is given by Lemma 5.9.2 and the second inequality follows from the definition of $\widehat{V}^k_{h,l^k_h(s)-1}(s)$. This leads to

$$\begin{aligned}\widehat{V}^k_h(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}^k_{h+1}](s,a) &\leqslant \big(\widehat{V}^k_h(s) - V^k_{h,l^k_h(s)-1}(s)\big) + \big(V^k_{h,l^k_h(s)-1}(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}^k_{h+1}](s,a)\big)\\ &\leqslant 3\cdot 2^{-(l^k_h(s)-1)} + 2\cdot 2^{-(l^k_h(s)-1)} + \chi\sqrt{L_\varepsilon}\zeta\\ &\leqslant 10\cdot 2^{-l^k_h(s)} + 0.01\varepsilon/H, \end{aligned} \tag{5.9.54}$$

where in the second inequality, the first term is given by (5.9.53) and the second term holds according to Lemma 5.9.22, and the third inequality holds from $\chi\sqrt{L_\varepsilon}\zeta \leqslant 0.01\varepsilon/H$ given by $\mathcal{G}_\varepsilon$.

Otherwise, we have $L_\varepsilon \leqslant \min\{L_\varepsilon, l^k_h(s)-1\}$, this leads to

$$\begin{aligned}\widehat{V}^k_h(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}^k_{h+1}](s,a) &\leqslant \big(\widehat{V}^k_h(s) - V^k_{h,L_\varepsilon}(s)\big) + \big(V^k_{h,L_\varepsilon}(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}^k_{h+1}](s,a)\big)\\ &\leqslant 3\cdot 2^{-L_\varepsilon} + 2\cdot 2^{-L_\varepsilon} + \chi\sqrt{L_\varepsilon}\zeta\\ &\leqslant 0.03\varepsilon/H + 0.02\varepsilon/H + 0.01\varepsilon/H = 0.06\varepsilon/H, \end{aligned} \tag{5.9.55}$$

where in the second inequality, the first term is given by the definition of $\widehat{V}^k_h(s)$ and the second term holds according to Lemma 5.9.22, and the third inequality holds from $\chi L_\varepsilon^{1.5}\zeta \leqslant 2^{-L_\varepsilon} \leqslant 0.01\varepsilon/H$ given by $\mathcal{G}_\varepsilon$. Combining (5.9.54) and (5.9.55) gives the desired statement

$$\widehat{V}^k_h(s) - \max_{a\in\mathcal{A}}[\mathbb{B}_h\widehat{V}^k_{h+1}](s,a) \leqslant 10\cdot 2^{-l^k_h(s)} + 0.06\varepsilon/H.$$

□

5.9.4.4 Proof of Lemma 5.9.36

Proof of Lemma 5.9.36. According to the definition in which $V^{\pi^k}_h(s^k_h) = [\mathbb{B}_h V^{\pi^k}_{h+1}](s^k_h, \pi^k_h(s^k_h))$ and $\eta^k_h + [\mathbb{P}_h(\widehat{V}^k_{h+1} - V^{\pi^k}_{h+1})](s^k_h, \pi^k_h(s^k_h)) - \big(\widehat{V}^k_{h+1}(s^k_{h+1}) - V^{\pi^k}_{h+1}(s^k_{h+1})\big)$. We can write

$$\widehat{V}^k_h(s^k_h) - V^{\pi^k}_h(s^k_h) = \big(\widehat{V}^k_h(s^k_h) - [\mathbb{B}_h\widehat{V}^k_{h+1}](s^k_h, \pi^k_h(s^k_h))\big) + \eta^k_h + \big(\widehat{V}^k_{h+1}(s^k_{h+1}) - V^{\pi^k}_{h+1}(s^k_{h+1})\big).$$

By a telescoping statement from h to H with the final terminal value $\widehat{V}_{H+1}^k(\cdot) = V_{H+1}^{\pi^k}(\cdot) = 0$, we reach

$$\widehat{V}_h^k(s_h^k) - V_h^{\pi^k}(s_h^k) = \sum_{h'=h}^{H} \left(\widehat{V}_h^k(s_h^k) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s_h^k, \pi_h^k(s_h^k))\right) + \sum_{h'=h}^{H} \eta_{h'}^k. \tag{5.9.56}$$

As a result, we can bound the desired term by

$$\begin{aligned}
V_h^*(s_h^k) - V_h^{\pi^k}(s_h^k) &\leqslant \widehat{V}_h^k(s_h^k) - V_h^{\pi^k}(s_h^k) + 0.07\varepsilon \\
&= \sum_{h'=h}^{H} \left(\widehat{V}_h^k(s_h^k) - [\mathbb{B}_h \widehat{V}_{h+1}^k](s_h^k, \pi_h^k(s_h^k))\right) + \sum_{h'=h}^{H} \eta_{h'}^k + 0.07\varepsilon \\
&\leqslant \sum_{h'=h}^{H} \left(26 \cdot 2^{-l_{h'}^k(s_{h'}^k)} + 0.16\varepsilon/H\right) + \sum_{h'=h}^{H} \eta_{h'}^k + 0.07\varepsilon \\
&= 0.23\varepsilon + 26 \sum_{h'=h}^{H} 2^{-l_{h'}^k(s_{h'}^k)} + \sum_{h'=h}^{H} \eta_{h'}^k.
\end{aligned}$$

where the first inequality is given by Lemma 5.9.25, the first equality is given by (5.9.56), and the final inequality is given by Lemma 5.9.26. □

5.9.5 Technical Numerical Lemmas

Lemma 5.9.37. If $|\mathcal{C}_{h,l}^k| \leqslant 4^l d + 2.5 \cdot 4^l \gamma_l^2 d \ln\left(1 + |\mathcal{C}_{h,l}^k|/(16d)\right)$, then $|\mathcal{C}_{h,l}^k| \leqslant 16l \cdot 4^l \gamma_l^2 d$.

Proof. Denote $c = |\mathcal{C}_{h,l}^k|/(l \cdot 4^l \gamma_l^2 d)$. We have that

$$cl \cdot 4^l \gamma_l^2 d \leqslant 4^l d + 2.5 \cdot 4^l \gamma_l^2 d \ln(1 + cl \cdot 4^l \gamma_l^2/16).$$

Dividing both sides by $4^l \gamma_l^2 d$, we have that

$$\begin{aligned}
cl &\leqslant 1/\gamma_l^2 + 2.5 \ln(1 + cl \cdot 4^l \gamma_l^2/16) \\
&\leqslant 1/\gamma_l^2 + 2.5 \ln(4c \cdot 5^l \gamma_l^2/16) \leqslant 1/\gamma_l^2 + 4.1l + 2.5 \ln(c).
\end{aligned}$$

Since $l \geqslant 1$ and $\gamma_l \geqslant 1$, we can further conclude that

$$c \leqslant 5.1 + 2.5 \ln(c) \leqslant 5.1 + 2.5(1 + c/6).$$

The necessary condition for the above inequality is $c \leqslant 16$, which proves the desired statement. □

Lemma 5.9.38. For any $l \geqslant 1$, $\gamma_{l+1}/\gamma_l \leqslant 1.4$.

Proof. Firstly, we have that

$$\frac{l + 22 + \log(l+1)}{l + 20 + \log(l)} \leqslant \frac{l + 22 + 0.2l + 2}{l + 20} = 1.2, \tag{5.9.57}$$

where the first inequality holds due to $\log(x+1) \leqslant 0.2x + 2$. In addition, we have

$$\frac{4 + \log(l+1)}{4 + \log(l)} \leqslant \frac{4 + \log(l) + 1}{4 + \log(l)} \leqslant 1.25, \tag{5.9.58}$$

where the first inequality holds due to $\log(x+1) \leqslant \log(x) + 1$. As a result, we can reach the desired statement according to

$$\begin{aligned}
\frac{\gamma_{l+1}}{\gamma_l} &= \frac{5(l + 1 + \lceil 20 + \log((l+1)d)\rceil)dH\sqrt{\log(16(l+1)dH/\delta)}}{5(l + \lceil 20 + \log(ld)\rceil)dH\sqrt{\log(16ldH/\delta)}} \\
&\leqslant \frac{l + 22 + \log(l+1) + \log(d)}{l + 20 + \log(l) + \log(d)} \cdot \sqrt{\frac{\log(l+1) + \log(16dH/\delta)}{\log(l) + \log(16dH/\delta)}} \\
&\leqslant \frac{l + 22 + \log(l+1)}{l + 20 + \log(l)} \cdot \sqrt{\frac{\log(l+1)}{\log(l)}} \\
&\leqslant 1.2\sqrt{1.25} \\
&\leqslant 1.4,
\end{aligned}$$

where the third inequality holds from plugging both (5.9.57) and (5.9.58). □

Lemma 5.9.39.

$$\sqrt{2d\ln(1 + l \cdot 4^l\gamma_l^2) + 2\ln(l^2H(2^{22}d^6H^4)^{l_+^2 d^2}/\delta)} \leqslant \gamma_{l,l_+}$$

Proof. By calculation, we have that

$$\begin{aligned}
&H\sqrt{2d\ln(1+l\cdot 4^l\gamma_l^2)+2\ln(l^2H(2^{22}d^6H^4)^{l_+^2d^2}/\delta)}\\
\leqslant\ &H\sqrt{2d\ln(1+l\cdot 4^l\cdot 1.4^{2l}\gamma_1^2)}+H\sqrt{12l_+^2d^2\ln(2^4ldH/\delta)}\\
\leqslant\ &l_+dH\sqrt{2\ln(2^4ldH/\delta)}+l_+dH\sqrt{12\ln(2^4ldH/\delta)}\\
\leqslant\ &5l_+dH\sqrt{\log(2^4\gamma_{l_+}ldH/\delta)}\\
=\ &\gamma_{l,l_+}.
\end{aligned}$$

□

Lemma 5.9.40. If some constant $c_1, c_2 > 0$ that

$$|\mathcal{K}_h^\varepsilon| < c_1L_\varepsilon(L_\varepsilon+\log(dH))^2d^3H^4\varepsilon^{-2}\log(L_\varepsilon d/\delta)+\varepsilon^{-1}\sqrt{c_2H^3|\mathcal{K}_h^\varepsilon|\log(H|\mathcal{K}_h^\varepsilon|\log(\varepsilon^{-1})/\delta)}.$$

Then, there exists $c_3 > 0$ such that

$$|\mathcal{K}_h^\varepsilon| < c_3L_\varepsilon(L_\varepsilon+\log(dH))^2d^3H^4\varepsilon^{-2}\log(L_\varepsilon d)\log(\delta^{-1})\iota,$$

where ι is a polynomial of $\log\log(L_\varepsilon dH\delta^{-1})$.

Proof. Let $x = |\mathcal{K}_h^\varepsilon|/\log(|\mathcal{K}_h^\varepsilon|)$. We have that

$$x < c_1L_\varepsilon(L_\varepsilon+\log(dH))^2d^3H^4\varepsilon^{-2}\log(L_\varepsilon d/\delta)+\varepsilon^{-1}\sqrt{c_2H^3x\log(H\log(\varepsilon^{-1})/\delta)}.$$

Since $x < a+\sqrt{bx}$ implies $x < 2a+2b$, so the above inequality implies

$$x < 2c_1L_\varepsilon(L_\varepsilon+\log(dH))^2d^3H^4\varepsilon^{-2}\log(L_\varepsilon d/\delta)+2c_2H^3\varepsilon^{-2}\log(H\log(\varepsilon^{-1})/\delta).$$

Moreover, since $y/\log(y) < a$ implies $y < 2a\log a$, we can conclude that there exists $c_3 > 0$ that

$$|\mathcal{K}_h^\varepsilon| < c_3L_\varepsilon(L_\varepsilon+\log(dH))^2d^3H^4\varepsilon^{-2}\log(L_\varepsilon d)\log(\delta^{-1})\iota,$$

where ι is a polynomial of $\log\log(L_\varepsilon dH\varepsilon^{-1}\delta^{-1})$. □

CHAPTER 6

Conclusions and Future Directions

This dissertation addressed several key concerns in reinforcement learning, including *unsupervised exploration* in the face of the uncertainty of the environment and *model robustness* in the face of the uncertainty of the function approximation, from the perspective of theoretical analysis. Several practical algorithms were also proposed to achieve competitive performance with theoretical guarantees. In particular, in the first part of this dissertation, we analyzed reward-free exploration with linear function approximations then extended the analysis to general function approximations. We affirmatively answered the question: *How to explore the environment without human supervision* by building the connection between reward-free exploration with unsupervised reinforcement learning from both theoretical and empirical perspectives. In the second part of this dissertation, we discussed model misspecification for decision making systems. We answered the question by providing a theoretical threshold showing *How large a model misspecification can be tolerated in order to make good decisions*. We also proposed algorithms in the context of misspecified linear bandits and reinforcement learning. All of the proposed algorithms will provably only suffer from finite suboptimality over infinite runs, without additional prior assumptions. To the best of our knowledge, these are the first constant regret results in the literature.

This dissertation also suggests several open questions for future research. In particular, the first part of this dissertation assumes that the reward function is provided as an oracle during the planning phase. However, learning the reward function can be challenging in practice. One might ask, *How would current reward-free exploration methods integrate with*

reward learning processes? For example, it would be valuable to investigate how the reward learning process, such as fine-tuning (Laskin et al., 2020), RL with human feedback (Peng et al., 2023; Christiano et al., 2017; Rafailov et al., 2024), could affect the planning phase in reward-free exploration. In the second part of this thesis, we have demonstrated a minimax-optimal separation between model misspecification and the suboptimality gap. However, several open questions remain unresolved in this context. First, there is a $\log |\mathcal{D}|$ gap between the positive and negative results (Lattimore et al., 2020). In RL, this translates into the difference between misspecifications in the $\|\cdot\|_{\mathrm{TV}}$ norm and the $\|\cdot\|_{\infty}$ norm, as used in Du et al. (2019), which can become significant as the number of states increases. Second, the current multi-phase estimation approach is challenging to implement in practice and suffers from a large constant, although it is of the same order in $\widetilde{\mathcal{O}}(\cdot)$ notation. It remains an open question whether this gap between positive and negative results can be closed and whether a more practical algorithm for this multi-phase estimation regime can be developed. Such investigations would not only enhance the theoretical understanding of RL but also enhance confidence in applying RL to critical tasks such as dynamic clinical treatments or autonomous scientific discoveries.

In addition to the technical open questions highlighted earlier, this dissertation opens several promising directions for future research. For instance, *foundation models*, such as diffusion models (Ho et al., 2020; Song et al., 2020) and large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023), have shown potential in enhancing our understanding of linguistic and visual inputs. On one front, reinforcement learning methods are widely used to finetune these models with online human preferences (Ouyang et al., 2022; Rafailov et al., 2024). These applications necessitate a thorough investigation into the robustness and efficiency of these algorithms. For example, a key question to explore is whether RL methods would exacerbate or mitigate hallucination in LLMs. Furthermore, there is considerable potential in harnessing the capabilities of existing foundation models to better understand environmental interactions and enhance decision-making processes (Zhao et al., 2024). It is

crucial to develop a framework that specifically analyzes the behavior of these models within RL agents, moving beyond the use of general function approximators.

Lastly, while the application of RL in gaming has been well explored, extending these methodologies to more practical fields, such as drug design (Popova et al., 2018) or policy-making for pandemic control (Kwak et al., 2021), could be highly beneficial. Integrating RL with autonomous systems (Sheng et al., 2024) would significantly enhance the efficiency of these applications. Moreover, establishing performance guarantees for the robustness, explainability and accountability of RL agents becomes imperative, particularly in critical domains such as healthcare, science discovery or autonomous laboratory.

Bibliography

Abbasi-Yadkori, Y., Pál, D. and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, vol. 24.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* .

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L. and Schapire, R. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*. PMLR.

Agarwal, A., Jin, Y. and Zhang, T. (2022). Voql: Towards optimal regret in model-free rl with nonlinear function approximation. *arXiv preprint arXiv:2212.06069* .

Agrawal, S. and Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*. PMLR.

Auer, P. (2002). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* **3** 397–422.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M. and Yang, L. (2020). Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*. PMLR.

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series* **19** 357–367.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C. et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* .

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T. and Efros, A. A. (2018a). Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355* .

Burda, Y., Edwards, H., Storkey, A. and Klimov, O. (2018b). Exploration by random network distillation. *arXiv preprint arXiv:1810.12894* .

Cai, Q., Yang, Z., Jin, C. and Wang, Z. (2020). Provably efficient exploration in policy optimization. In *International Conference on Machine Learning*. PMLR.

Camilleri, R., Jamieson, K. and Katz-Samuels, J. (2021). High-dimensional experimental design and kernel bandits. In *International Conference on Machine Learning*. PMLR.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chen, F., Mei, S. and Bai, Y. (2022a). Unified algorithms for rl with decision-estimation coefficients: No-regret, pac, and reward-free learning. *arXiv preprint arXiv:2209.11745* .

Chen, J., Modi, A., Krishnamurthy, A., Jiang, N. and Agarwal, A. (2022b). On the statistical efficiency of reward-free exploration in non-linear rl. *Advances in Neural Information Processing Systems* **35** 20960–20973.

Chen, X., Hu, J., Yang, L. and Wang, L. (2021). Near-optimal reward-free exploration for linear mixture mdps with plug-in solver. In *International Conference on Learning Representations*.

Chen, Z., Li, C. J., Yuan, A., Gu, Q. and Jordan, M. I. (2022c). A general framework for sample-efficient function approximation in reinforcement learning. *arXiv preprint arXiv:2209.15634* .

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S. and Amodei, D. (2017). Deep reinforcement learning from human preferences. *Advances in neural information processing systems* **30**.

Chu, W., Li, L., Reyzin, L. and Schapire, R. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings.

Dann, C., Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J. and Schapire, R. E. (2018). On oracle-efficient pac rl with rich observations. *Advances in neural information processing systems* **31**.

Dann, C., Marinov, T. V., Mohri, M. and Zimmert, J. (2021). Beyond value-function gaps: Improved instance-dependent regret bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems* **34**.

Dantzig, G. B. (1965). *Linear programming and extensions*, vol. 48. Princeton university press.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*.

Dilokthanakul, N., Kaplanis, C., Pawlowski, N. and Shanahan, M. (2019). Feature control as intrinsic motivation for hierarchical reinforcement learning. *IEEE transactions on neural networks and learning systems* **30** 3409–3418.

Du, S. S., Kakade, S. M., Wang, R. and Yang, L. F. (2019). Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016* .

Elson, J., Douceur, J. J., Howell, J. and Saul, J. (2007). Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Con-*

ference on Computer and Communications Security (CCS). Association for Computing Machinery, Inc.

Eysenbach, B., Gupta, A., Ibarz, J. and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070* .

Foster, D. J., Gentile, C., Mohri, M. and Zimmert, J. (2020). Adapting to misspecification in contextual bandits. *Advances in Neural Information Processing Systems* **33**.

Foster, D. J., Kakade, S. M., Qian, J. and Rakhlin, A. (2021). The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487* .

Ghosh, A., Chowdhury, S. R. and Gopalan, A. (2017). Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31.

Hafner, D., Lillicrap, T., Ba, J. and Norouzi, M. (2019a). Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603* .

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H. and Davidson, J. (2019b). Learning latent dynamics for planning from pixels. In *International conference on machine learning*. PMLR.

Hansen, S., Dabney, W., Barreto, A., Van de Wiele, T., Warde-Farley, D. and Mnih, V. (2019). Fast task inference with variational intrinsic successor features. *arXiv preprint arXiv:1906.05030* .

Hao, B., Lattimore, T. and Szepesvari, C. (2020). Adaptive exploration in linear contextual bandit. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

He, J., Zhao, H., Zhou, D. and Gu, Q. (2022a). Nearly minimax optimal reinforcement learning for linear markov decision processes. *arXiv preprint arXiv:2212.06132* .

He, J., Zhou, D. and Gu, Q. (2021a). Logarithmic regret for reinforcement learning with linear function approximation. In *International Conference on Machine Learning*. PMLR.

He, J., Zhou, D. and Gu, Q. (2021b). Uniform-PAC bounds for reinforcement learning with linear function approximation. In *Advances in Neural Information Processing Systems*.

He, J., Zhou, D., Zhang, T. and Gu, Q. (2022b). Nearly optimal algorithms for linear contextual bandits with adversarial corruptions. In *Advances in Neural Information Processing Systems*.

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Ho, J., Jain, A. and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33** 6840–6851.

Hu, P., Chen, Y. and Huang, L. (2022). Towards minimax optimal reward-free reinforcement learning in linear mdps. In *The Eleventh International Conference on Learning Representations*.

Jaksch, T., Ortner, R. and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research* **11**.

Jia, Z., Yang, L., Szepesvari, C. and Wang, M. (2020). Model-based reinforcement learning with value-targeted regression. In *Learning for Dynamics and Control*. PMLR.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J. and Schapire, R. E. (2017). Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*. PMLR.

Jin, C., Allen-Zhu, Z., Bubeck, S. and Jordan, M. I. (2018). Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*.

Jin, C., Krishnamurthy, A., Simchowitz, M. and Yu, T. (2020a). Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Jin, C., Liu, Q. and Miryoosefi, S. (2021). Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in neural information processing systems* **34** 13406–13418.

Jin, C., Yang, Z., Wang, Z. and Jordan, M. I. (2020b). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*. PMLR.

Kalashnikov, D., Varley, J., Chebotar, Y., Swanson, B., Jonschkowski, R., Finn, C., Levine, S. and Hausman, K. (2022). Scaling up multi-task robotic reinforcement learning. In *Conference on Robot Learning*. PMLR.

Kang, Y., Zhao, E., Zang, Y., Li, K. and Xing, J. (2022). Towards a unified benchmark for reinforcement learning in sparse reward environments. In *International Conference on Neural Information Processing*. Springer.

Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*.

Kaufmann, E., Ménard, P., Domingues, O. D., Jonsson, A., Leurent, E. and Valko, M. (2021a). Adaptive reward-free exploration. In *Algorithmic Learning Theory*. PMLR.

Kaufmann, E., Ménard, P., Domingues, O. D., Jonsson, A., Leurent, E. and Valko, M. (2021b). Adaptive reward-free exploration. In *Algorithmic Learning Theory*. PMLR.

Kearns, M. and Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine learning* **49** 209–232.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems* **30**.

Kober, J., Bagnell, J. A. and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* **32** 1238–1274.

Kong, D., Salakhutdinov, R., Wang, R. and Yang, L. F. (2021). Online sub-sampling for reinforcement learning with general function approximation. *arXiv preprint arXiv:2106.07203* .

Kwak, G. H., Ling, L. and Hui, P. (2021). Deep reinforcement learning approaches for global public health strategies for covid-19 pandemic. *PloS one* **16** e0251550.

Laskin, M., Srinivas, A. and Abbeel, P. (2020). Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Laskin, M., Yarats, D., Liu, H., Lee, K., Zhan, A., Lu, K., Cang, C., Pinto, L. and Abbeel, P. (2021). Urlb: Unsupervised reinforcement learning benchmark. *arXiv preprint arXiv:2110.15191* .

Lattimore, T. and Szepesvári, C. (2020). *Bandit algorithms*. Cambridge University Press.

Lattimore, T., Szepesvari, C. and Weisz, G. (2020). Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*. PMLR.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S. and Salakhutdinov, R. (2019). Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274* .

Levine, S., Finn, C., Darrell, T. and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* **17** 1334–1373.

Li, L., Chu, W., Langford, J. and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*.

Li, L., Chu, W., Langford, J. and Wang, X. (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*.

Li, Y., Wang, R. and Yang, L. F. (2022). Settling the horizon-dependence of sample complexity in reinforcement learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE.

Liu, H. and Abbeel, P. (2021a). Aps: Active pretraining with successor features. In *International Conference on Machine Learning*. PMLR.

Liu, H. and Abbeel, P. (2021b). Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems* **34** 18459–18473.

Lykouris, T., Simchowitz, M., Slivkins, A. and Sun, W. (2021). Corruption-robust exploration in episodic reinforcement learning. In *Conference on Learning Theory*. PMLR.

Mai, V., Mani, K. and Paull, L. (2022). Sample efficient deep reinforcement learning via uncertainty estimation. *arXiv preprint arXiv:2201.01666* .

Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E. and Valko, M. (2020). Fast active learning for pure exploration in reinforcement learning. *arXiv preprint arXiv:2007.13442* .

Ménard, P., Domingues, O. D., Jonsson, A., Kaufmann, E., Leurent, E. and Valko, M. (2021). Fast active learning for pure exploration in reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Michael, S. P. S. T. J. and Jordan, I. (1995). Reinforcement learning with soft state aggregation. *Advances in neural information processing systems 7* **7** 361.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* .

Modi, A., Jiang, N., Tewari, A. and Singh, S. (2020). Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems* **35** 27730–27744.

Papini, M., Tirinzoni, A., Pacchiano, A., Restelli, M., Lazaric, A. and Pirotta, M. (2021a). Reinforcement learning in linear mdps: Constant regret and representation selection. *Advances in Neural Information Processing Systems* **34** 16371–16383.

Papini, M., Tirinzoni, A., Restelli, M., Lazaric, A. and Pirotta, M. (2021b). Leveraging good representations in linear contextual bandits. In *International Conference on Machine Learning*. PMLR.

Pathak, D., Agrawal, P., Efros, A. A. and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. PMLR.

Pathak, D., Gandhi, D. and Gupta, A. (2019). Self-supervised exploration via disagreement. In *International conference on machine learning*. PMLR.

Peng, B., Li, C., He, P., Galley, M. and Gao, J. (2023). Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277* .

Popova, M., Isayev, O. and Tropsha, A. (2018). Deep reinforcement learning for de novo drug design. *Science advances* **4** eaap7885.

Qiu, S., Wang, L., Bai, C., Yang, Z. and Wang, Z. (2022). Contrastive ucb: Provably efficient contrastive self-supervised learning in online reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S. and Finn, C. (2024). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* **36**.

Ren, T., Li, J., Dai, B., Du, S. S. and Sanghavi, S. (2021). Nearly horizon-free offline reinforcement learning. *Advances in neural information processing systems* **34** 15621–15634.

Russo, D. and Van Roy, B. (2013). Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*. Citeseer.

Sallab, A. E., Abdou, M., Perot, E. and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532* .

Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*. PMLR.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P. and Lee, K. (2021). State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*. PMLR.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Sharma, A., Gu, S., Levine, S., Kumar, V. and Hausman, K. (2019). Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657* .

Sheng, H., Sun, J., Rodríguez, O., Hoar, B. B., Zhang, W., Xiang, D., Tang, T., Hazra, A., Min, D. S., Doyle, A. G. et al. (2024). Autonomous closed-loop mechanistic investigation of molecular electrochemistry via automation. *Nature Communications* **15** 2781.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature* **529** 484–489.

Simchowitz, M. and Jamieson, K. G. (2019). Non-asymptotic gap-dependent regret bounds for tabular mdps. *Advances in Neural Information Processing Systems* **32** 1153–1162.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S. and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Stooke, A., Lee, K., Abbeel, P. and Laskin, M. (2021). Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*. PMLR.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A. and Langford, J. (2019). Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*. PMLR.

Sutton, R. S., Barto, A. G. et al. (1998). Introduction to reinforcement learning. vol. 135.

Takemura, K., Ito, S., Hatano, D., Sumita, H., Fukunaga, T., Kakimura, N. and Kawarabayashi, K.-i. (2021). A parameter-free algorithm for misspecified linear contextual bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Tarbouriech, J., Zhou, R., Du, S. S., Pirotta, M., Valko, M. and Lazaric, A. (2021). Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret. *Advances in Neural Information Processing Systems* **34** 6843–6855.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* .

Tunyasuvunakool, S., Muldal, A., Doron, Y., Liu, S., Bohez, S., Merel, J., Erez, T., Lillicrap, T., Heess, N. and Tassa, Y. (2020). dm_control: Software and tasks for continuous control. *Software Impacts* **6** 100022.

Uehara, M., Zhang, X. and Sun, W. (2021). Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652* .

Van Roy, B. and Dong, S. (2019). Comments on the du-kakade-wang-yang lower bounds. *arXiv preprint arXiv:1911.07910* .

Vial, D., Parulekar, A., Shakkottai, S. and Srikant, R. (2022). Improved algorithms for misspecified linear markov decision processes. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P. et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* **575** 350–354.

Wagenmaker, A. J., Chen, Y., Simchowitz, M., Du, S. and Jamieson, K. (2022). Reward-free rl is no harder than reward-aware rl in linear markov decision processes. In *International Conference on Machine Learning*. PMLR.

Wang, R., Du, S. S., Yang, L. F. and Kakade, S. M. (2020a). Is long horizon reinforcement learning more difficult than short horizon reinforcement learning? *arXiv preprint arXiv:2005.00527* .

Wang, R., Du, S. S., Yang, L. F. and Salakhutdinov, R. (2020b). On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems* .

Wang, R., Salakhutdinov, R. R. and Yang, L. (2020c). Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems* **33** 6123–6135.

Wang, Y., Wang, R., Du, S. S. and Krishnamurthy, A. (2019). Optimism in reinforcement learning with generalized linear function approximation. In *International Conference on Learning Representations*.

Wei, C.-Y., Dann, C. and Zimmert, J. (2022). A model selection approach for corruption robust reinforcement learning. In *International Conference on Algorithmic Learning Theory*. PMLR.

Weisz, G., Amortila, P. and Szepesvári, C. (2021). Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*. PMLR.

Wu, Y., Zhou, D. and Gu, Q. (2021). Nearly minimax optimal regret for learning infinite-horizon average-reward mdps with linear function approximation. *arXiv preprint arXiv:2102.07301* .

Yang, K., Yang, L. and Du, S. (2021). Q-learning with logarithmic regret. In *International Conference on Artificial Intelligence and Statistics*. PMLR.

Yang, L. and Wang, M. (2019). Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*. PMLR.

Yang, L. and Wang, M. (2020a). Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*. PMLR.

Yang, L. and Wang, M. (2020b). Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*. PMLR.

Yarats, D., Fergus, R., Lazaric, A. and Pinto, L. (2021a). Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*. PMLR.

Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J. and Fergus, R. (2021b). Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35.

Ye, C., Yang, R., Gu, Q. and Zhang, T. (2023). Corruption-robust offline reinforcement learning with general function approximation. *arXiv preprint arXiv:2310.14550* .

Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M. and Lazaric, A. (2020a). Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*.

Zanette, A., Lazaric, A., Kochenderfer, M. and Brunskill, E. (2020b). Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*. PMLR.

Zanette, A., Lazaric, A., Kochenderfer, M. J. and Brunskill, E. (2020c). Learning near optimal policies with low inherent bellman error. In *ICML*.

Zanette, A., Lazaric, A., Kochenderfer, M. J. and Brunskill, E. (2020d). Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems* .

Zhang, J., Zhang, W. and Gu, Q. (2023a). Optimal horizon-free reward-free exploration for linear mixture mdps. *arXiv preprint arXiv:2303.10165* .

Zhang, W., He, J., Fan, Z. and Gu, Q. (2023b). On the interplay between misspecification and sub-optimality gap: From linear contextual bandits to linear MDPs.

Zhang, W., He, J., Fan, Z. and Gu, Q. (2023c). On the interplay between misspecification and sub-optimality gap in linear contextual bandits. *arXiv preprint arXiv:2303.09390* .

Zhang, W., He, J., Zhou, D., Zhang, A. and Gu, Q. (2021a). Provably efficient representation learning in low-rank markov decision processes. *arXiv preprint arXiv:2106.11935* .

Zhang, W., Zhou, D. and Gu, Q. (2021b). Reward-free model-based reinforcement learning with linear function approximation. *Advances in Neural Information Processing Systems* **34**.

Zhang, Z., Du, S. S. and Ji, X. (2020). Nearly minimax optimal reward-free reinforcement learning. *arXiv preprint arXiv:2010.05901* .

Zhang, Z., Ji, X. and Du, S. (2021c). Is reinforcement learning more difficult than bandits? a near-optimal algorithm escaping the curse of horizon. In *Conference on Learning Theory*. PMLR.

Zhang, Z., Ji, X. and Du, S. (2022). Horizon-free reinforcement learning in polynomial time: the power of stationary policies. In *Conference on Learning Theory*. PMLR.

Zhang, Z., Yang, J., Ji, X. and Du, S. S. (2021d). Improved variance-aware confidence sets for linear bandits and linear mixture mdp. *Advances in Neural Information Processing Systems* **34**.

Zhang, Z., Zhou, Y. and Ji, X. (2021e). Model-free reinforcement learning: from clipped pseudo-regret to sample complexity. In *International Conference on Machine Learning*. PMLR.

Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J. and Huang, G. (2024). Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38.

Zhao, H., He, J. and Gu, Q. (2023). A nearly optimal and low-switching algorithm for reinforcement learning with general function approximation. *arXiv preprint arXiv:2311.15238* .

Zhou, D. and Gu, Q. (2022a). Computationally efficient horizon-free reinforcement learning for linear mixture mdps. *arXiv preprint arXiv:2205.11507* .

Zhou, D. and Gu, Q. (2022b). Computationally efficient horizon-free reinforcement learning for linear mixture mdps. *arXiv preprint arXiv:2205.11507* .

Zhou, D., Gu, Q. and Szepesvari, C. (2021a). Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*. PMLR.

Zhou, D., Gu, Q. and Szepesvari, C. (2021b). Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*. PMLR.

Zhou, D., He, J. and Gu, Q. (2021c). Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*. PMLR.

Zhou, D., He, J. and Gu, Q. (2021d). Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*. PMLR.