Tackling Challenges in Modern Reinforcement Learning: Long Planning Horizons and Large State Spaces

Ruosong Wang

CMU-CS-22-100

February 2022

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Ruslan Salakhutdinov, Chair
Zico Kolter
Aarti Singh
Sham M. Kakade (Harvard University)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by Apple, Inc., the Defense Advanced Research Projects Agency under award numbers HR00111890033, HR00111990016, and FA875018C0150, the Office of Naval Research under award numbers N000141512791 and N000141812562, and the Army Research Office under award number W911NF1920104 . The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: Reinforcement Learning, Sample Complexity, Function Approximation

To my parents

Abstract

Modern reinforcement learning (RL) methods have achieved phenomenal success on various applications. However, reinforcement learning problems with large state spaces and long planning horizons remain challenging due to the excessive sample complexity burden, and our current understanding is rather limited for such problems. Moreover, there are important problems in RL that cannot be addressed by the classical frameworks. In this thesis, we study the above issues to build a better understanding of modern RL methods.

This thesis is divided into the following three parts:

Part I: RL with Long Planning Horizons. Learning to plan for long horizons is a central challenge in RL, and a fundamental question is to understand how the difficulty of RL scales as the horizon increases. In the first part of this thesis, we show that tabular reinforcement learning is possible with a sample complexity that is completely independent of the planning horizon, and therefore, long horizon RL is no more difficult than short horizon RL, at least in a minimax sense.

Part II: RL with Large State Spaces. In modern RL methods, function approximation schemes are deployed to deal with large state spaces. Empirically, combining RL algorithms with neural networks for feature extraction has led to tremendous success on various tasks. However, these methods often require a large amount of samples to learn a good policy, and it is unclear if there are fundamental statistical limits on such methods. In the second part of this thesis, we study necessary and sufficient conditions on the representation power of the features that permit sample-efficient reinforcement learning, through both theoretical analysis and experiments.

Part III: RL in Other Settings. Classical reinforcement learning paradigms aim to maximize the cumulative reward when the agent has access to the reward values. Despite being able to formulate a large family of sequential decision-making problems, there are important applications that cannot be casted into the classical frameworks. In the third part of this thesis, we study two new settings, the reward-free exploration setting and planning with general objective functions, that generalize the classical frameworks.

Acknowledgments

First and foremost, I would like to thank my advisor Ruslan Salakhutdinov for his guidance and support. Russ is not only a brilliant researcher but also the best advisor one can hope for. Russ always gives me the freedom to do research in fields that I am passionate about and is extremely supportive when times are hard. I want to thank the rest of my thesis committee, Zico Kolter, Aarti Singh, and Sham M. Kakade, for their help and insights which improved this thesis tremendously. I also want to thank my undergraduate mentors, Jian Li, Seth Pettie, Ran Duan, and Pingzhong Tang, for developing my interest in scientific research.

During my graduate studies, I was very fortunate to have had the opportunity to collaborate and discuss research with many talented people. I would like to thank my past and current collaborators: Sanjeev Arora, Lijie Chen, Simon S. Du, Artur W. Dubrawski, Fei Feng, Dean P. Foster, Kangcheng Hou, Wei Hu, Sham M. Kakade, Akshay Krishnamurthy, Jason D. Lee, Yi Li, Yuanzhi Li, Zhiyuan Li, Shachar Lovett, Yuping Luo, Gaurav Mahajan, Barnabás Póczos, Ruslan Salakhutdinov, Aarti Singh, Zhao Song, Wen Sun, Yining Wang, Yifan Wu, Keyulu Xu, Yichong Xu, Lin F. Yang, Wotao Yin, Dingli Yu, Hanrui Zhang, Hongyang Zhang, Peilin Zhong.

I am very grateful to everyone in the computer science department at CMU for contributing to a great environment for graduate studies. In particular, I would like to thank Deborah Cavlovich, for her tireless hard work in helping Ph.D. students in the CSD.

Graduate school could not have been as enjoyable without a supportive group of friends. I would like to thank my friends, Simon S. Du, Zhili Feng, Zhiyuan Li, Yifan Song, Lin F. Yang, Hanrui Zhang, and Hongyu Zheng, for making my experience at CMU memorable.

Finally, I would like to thank my parents, Shaoqing Xia and Songxian Wang, for their unconditional support and love. I dedicate this thesis to them.

Contents

1 Introduction **1**
- 1.1 Overview . . . 2
  - 1.1.1 Part I: RL with Long Planning Horizons . . . 2
  - 1.1.2 Part II: RL with Large State Spaces . . . 2
  - 1.1.3 Part III: RL in Other Settings . . . 3
- 1.2 Organization . . . 3

2 Background and Notations **5**

I RL with Long Planning Horizons **7**

3 RL with Long Planning Horizons **9**
- 3.1 Introduction . . . 9
- 3.2 Preliminaries . . . 10
- 3.3 Technical Overview . . . 12
  - 3.3.1 Technical Overview of Theorem 3.1.1 . . . 12
  - 3.3.2 Technical Overview of Theorem 3.1.2 . . . 13
- 3.4 Proof of Theorem 3.1.1 . . . 17
  - 3.4.1 An ε-net For Non-Stationary Policies . . . 17
  - 3.4.2 Evaluating Policies . . . 19
  - 3.4.3 The Algorithm . . . 24
- 3.5 Proof of Theorem 3.1.2 . . . 26
  - 3.5.1 Properties of Stationary Policies . . . 26
  - 3.5.2 The Algorithm . . . 34

II RL with Large State Spaces **59**

4 RL with Large State Spaces: Upper Bound in the Online Setting **61**
- 4.1 Introduction . . . 61
- 4.2 Notations and Assumptions . . . 62
- 4.3 Algorithm . . . 64
  - 4.3.1 Stable UCB via Importance Sampling . . . 65

4.4 Theoretical Guarantee . . . 66
4.4.1 Analysis of the Stable Bonus Function . . . 69
4.4.2 Analysis of the Algorithm . . . 75

5 RL with Large State Spaces: Lower Bounds in the Online Setting 83
5.1 Introduction . . . 83
5.2 Main Results . . . 84
5.2.1 Lower Bound for Value-Based Learning . . . 85
5.2.2 Lower Bound for Policy-Based Learning . . . 86
5.2.3 Proof Ideas . . . 86
5.3 Separations . . . 88
5.4 Proofs of Lower Bounds . . . 88
5.4.1 Proof of Lower Bound for Value-Based Learning . . . 89
5.4.2 Proof of Lower Bound for Policy-Based Learning . . . 91

6 RL with Large State Spaces: the Offline Setting 95
6.1 Introduction . . . 95
6.2 Preliminaries . . . 97
6.3 The Lower Bound: Realizability and Coverage are Insufficient . . . 98
6.4 Upper Bounds: Low Distribution Shift or Policy Completeness are Sufficient . . . 101
6.5 Experiments . . . 106
6.5.1 Experimental Methodology . . . 106
6.5.2 Results and Analysis . . . 108

III RL in Other Settings 115

7 Planning with General Objective Functions 117
7.1 Introduction . . . 117
7.1.1 Our Results . . . 118
7.2 Algorithm for General Objective Functions . . . 120
7.3 Hardness Results . . . 121
7.4 Proof of Theorem 7.1.1 . . . 123
7.5 Proof of Lower Bounds . . . 125

8 Reward-Free Exploration with Linear Function Approximation 131
8.1 Introduction . . . 131
8.2 Notations and Background . . . 133
8.2.1 Notations . . . 133
8.2.2 Linear Function Approximation . . . 133
8.2.3 Reward-Free Exploration . . . 134
8.3 Reward-Free Exploration for Linear MDPs . . . 135
8.3.1 Analysis . . . 136
8.3.2 Missing Proofs in Section 8.3.1 . . . 137

8.4 Lower Bound for Reward-Free Exploration under Linear Q^* Assumption 144
8.4.1 Missing Proofs 147

IV Conclusion **149**

9 Conclusion and Future Directions **151**

Bibliography **153**

List of Figures

5.1 An illustration of the hard instance for value-based learning 91
5.2 An illustration of the hard instance for policy-based learning 93
5.3 An illustration of the hard instance for policy-based learning 93

6.1 An illustration of the hard Instance for offline RL 99
6.2 Performance of FQI with features from pre-trained neural networks and datasets induced by random policies 110
6.3 Performance of FQI with features from pre-trained neural networks and datasets induced by lower performance policies 111
6.4 Performance of FQI with random Fourier features and datasets induced by random policies 111
6.5 Performance of FQI on Ant-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ . . 112
6.6 Performance of FQI on CartPole-v0, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ 112
6.7 Performance of FQI on HalfCheetah-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ 112
6.8 Performance of FQI on Hopper-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ 113
6.9 Performance of FQI on MountainCar-v0, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ 113
6.10 Performance of FQI on Walker2d-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ 113

8.1 An illustration of the hard instance for reward-free exploration 144

List of Tables

6.1 Values of randomly chosen states 108
6.2 Policy comparison results 109
6.3 Performance of LSTD 110

Chapter 1

Introduction

In the past decade, by exploiting the power of deep neural networks, significant progress has been made on static prediction problems such as image classification and natural language understanding. Recently, sequential decision-making problems have gained lots of interest from the machine learning community due to their wide applications in robotics [46], game playing [58], healthcare [106] and education [47]. In these problems, instead of making a single static prediction, the agent decides a sequence of actions to maximize the cumulative utility. These problems are challenging as we need to design a mechanism to balance exploration and exploitation, and reinforcement learning (RL) is a framework to formalize these problems. In recent years, by combining reinforcement learning algorithms with deep learning techniques (a.k.a. deep RL), strong empirical success has been achieved on a wide variety of real-world problems.

Despite their great empirical performance, a major problem is that deep RL methods require a large number of samples to learn a good policy. For example, deep Q-networks [58] require millions of samples to play a simple Atari game. It is unclear if there are fundamental statistical limits on these methods, or such sample complexity burden can be alleviated by a better algorithm. When dealing with real-world problems, practitioners heavily rely on heuristics for better sample efficiency, which limits the scope that RL algorithms can be applied to and also makes RL systems less robust and less transparent.

There is still a significant gap between modern RL algorithms and existing theory. For example, existing theory usually assumes a small state space. However, real-world decision-making problems usually have huge or even continuous state spaces (e.g., images, languages, or rich sensory inputs). Moreover, previous theoretical analysis usually assumes a short planning horizon, and an algorithm is said to be efficient if its sample complexity is polynomial in the planning horizon. However, in modern RL applications, the planning horizon could be as large as a few thousand, and even polynomial dependence is unacceptable. Furthermore, there are emerging settings in various applications (including unsupervised RL and general objective functions) that cannot be addressed by standard RL frameworks where the unknown environment is often modeled as a Markov decision process (MDP).

In this thesis, we propose to study the above issues to build a better understanding of modern RL methods.

1.1 Overview

In this section we give an overview of this thesis. This thesis is divided into the following three parts.

1.1.1 Part I: RL with Long Planning Horizons

Long horizons is the differentiator between RL problems and simpler bandit problems. In RL, actions taken at early stages could substantially impact the future. In contrast, for bandit problems, the action taken at each time step is independent of the future. Problems with long horizons are also ubiquitous in real-world applications. Therefore, understanding the optimal sample complexity dependence on the planning horizon is an important problem in RL.

In a COLT open problem [37], it was conjectured that for tabular episodic RL problems, there exists a sample complexity lower bound which exhibits a polynomial dependence on the planning horizon. In Chapter 3, we refute this conjecture by proving that tabular episodic RL is possible with a sample complexity that scales only logarithmically with the planning horizon. An informal statement of our main result is provided below.

Theorem 1.1.1 (Informal version of Theorem 3.1.1)**.** *There exists a RL algorithm that returns an ε-optimal policy with probability at least $1-\delta$ by sampling at most*

$$\mathrm{poly}\left(|\mathcal{S}|, |\mathcal{A}|, \log H, 1/\varepsilon, \log(1/\delta)\right)$$

episodes.

Although the conjecture in [37] has been refuted by Theorem 1.1.1, it is still unclear if tabular RL is possible with a sample complexity that is completely independent of the planning horizon. In Chapter 3, we further develop a new tabular RL algorithm whose sample complexity is completely independent of the planning horizon when the number of states and actions are constants, and thus completely answer the open problem in [37].

Theorem 1.1.2 (Informal version of Theorem 3.1.2)**.** *There exists a RL algorithm that returns an ε-optimal policy with probability at least $1-\delta$ by sampling at most*

$$(|\mathcal{S}||\mathcal{A}|)^{O(|\mathcal{S}|)} \cdot \log(1/\delta)/\varepsilon^5$$

episodes.

1.1.2 Part II: RL with Large State Spaces

The first part of this thesis is mainly concerned with the tabular setting where the number of states is bounded. However, in practice, the state space could be huge or even continuous, and function approximation schemes are deployed to deal with the curse of dimensionality. Despite great empirical success [58, 74], a major drawback is that these methods often require a large amount of samples to learn a good policy, and it is unclear if there are fundamental statistical limits on such methods.

In the second part of this thesis, we study necessary and sufficient conditions on the representation power of the features that permit sample-efficient reinforcement learning, in both the

online setting and the offline setting. Perhaps surprisingly, our theoretical results show that in both settings, conditions that permit sample-efficient supervised learning are generally insufficient for sample-efficient RL. Therefore, successful RL requires conditions that are substantially stronger that those sufficient for supervised learning.

Given these hardness results, it is natural to ask to what extent these worst-case characterizations are reflective of the scenarios that arise in practical applications. In Chapter 6, we provide a careful empirical investigation to understand these issues. Our experiments confirm the phenomenon predicted by our theoretical analysis and demonstrate that even in practice, the definition of a good representation in offline RL is more subtle than in supervised learning.

1.1.3 Part III: RL in Other Settings

Classical reinforcement learning paradigms aim to maximize the cumulative reward when the agent has access to the reward distributions. Despite being able to formulate a large family of sequential decision-making problems, there are important applications that cannot be casted into this framework. In the third part of this thesis, we study two new settings, the reward-free exploration setting and planning with general objective functions, to generalize the classical RL frameworks.

Planning with General Objective Functions. Standard RL paradigms aim to maximize the cumulative reward. However, this paradigm fails to model important practical applications. In Chapter 7, we consider a class of general objective functions that map scalar reward values to a real objective value, and give necessary and sufficient conditions on the objective function so that the problem is tractable.

Reward-Free Exploration. Exploration is widely regarded as one of the most challenging aspects of RL. To isolate the challenges of exploration, Jin et al. [39] propose a new reward-free exploration framework. During the exploration phase, an agent collects samples without using a pre-specified reward function. After the exploration phase, a reward function is given, and the agent uses samples collected during the exploration phase to compute a near-optimal policy. Jin et al. [39] show that in the tabular setting, the agent only needs to collect polynomial number of samples (in terms of the number of states, the number of actions, and the planning horizon) for reward-free exploration. However, in practice, the number of states and actions can be large, and thus function approximation schemes are required for generalization. In Chapter 8, we give both positive and negative results for reward-free exploration with linear function approximation. Our results imply several interesting exponential separations on the sample complexity of reward-free exploration.

1.2 Organization

The remaining part of this thesis is organized as follows.

0. Background

- In Chapter 2, we introduce notations and necessary background.

1. RL with Long Planning Horizons
   - In Chapter 3, we present our results for RL with long horizons. This chapter is based on a paper that appeared in NeurIPS 2020 [89] and another paper that appeared in FOCS 2021 [52].
2. RL with Large State Spaces
   - In Chapter 4, we present our upper bound for online RL with large state spaces. This chapter is based on a paper that appeared in NeurIPS 2020 [92].
   - In Chapter 5, we present our hardness results for online RL with large state spaces. This chapter is based on a paper that appeared in ICLR 2020 [24].
   - In Chapter 6, we present our results for offline RL with large state spaces. This chapter is based on a paper that appeared in ICLR 2021 [91] and another paper that appeared in ICML 2021 [94].
3. RL in Other Settings
   - In Chapter 7, we present our results for planning with general objective functions. This chapter is based on a paper that appeared in NeurIPS 2020 [95].
   - In Chapter 8, we present our results for reward-free exploration. This chapter is based on a paper that appeared in NeurIPS 2020 [90].
4. Conclusion and Future Directions
   - In Chapter 9, we conclude the thesis and list future directions.

Excluded Research. In order to keep this thesis succinct and coherent, a significant portion of the author's Ph.D. work has been excluded. The excluded research includes:

- work on numerical linear algebra and sketching algorithms [18, 51, 53, 79, 87, 93];
- work on the neural tangent kernel theory [8, 9, 10, 22];
- work on other theoretical aspects of reinforcement learning [23, 25, 26, 30, 96, 97, 101].

Chapter 2

Background and Notations

Notations. Throughout this thesis, for a given non-negative integer H, we use $[H]$ to denote the set $\{1, 2, \ldots, H\}$. For a condition $\mathcal{E}$, we use $\mathbb{I}[\mathcal{E}]$ to denote the indicator function, i.e., $\mathbb{I}[\mathcal{E}] = 1$ if $\mathcal{E}$ holds and $\mathbb{I}[\mathcal{E}] = 0$ otherwise. For a vector $x \in \mathbb{R}^d$, we use $\|x\|_p$ to denote its ℓ_p norm. For a positive semidefinite matrix A, we use $\|A\|_2$ to denote its operator norm, and $\sigma_{\min}(A)$ to denote its smallest eigenvalue. For two positive semidefinite matrices A and B, we write $A \succeq B$ to denote the Löwner partial ordering of matrices, i.e, $A \succeq B$ if and only if $A - B$ is positive semidefinite. For a vector $x \in \mathbb{R}^d$ and a positive semidefinite matrix $A \in \mathbb{R}^{d \times d}$, we use $\|x\|_A$ to denote $\sqrt{x^\top A x}$. Throughout this thesis, we use $\widetilde{O}(\cdot)$ to omit logarithmic factors.

Episodic Reinforcement Learning. Let $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ be a *Markov Decision Process* (MDP) where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition operator which takes a state-action pair as input and returns a distribution over states, $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathbb{R})$ is the reward distribution, $H \in \mathbb{Z}_+$ is the planning horizon (episode length), and $\mu \in \Delta(\mathcal{S})$ is the initial state distribution. We use *contextual bandit problem* to denote an MDP with $H = 1$. We use *deterministic system* to denote an MDP when the rewards, the transition operators and the initial state distribution are all deterministic. Throughout this thesis, for a state $s \in \mathcal{S}$, we occasionally abuse notation and use s to denote the deterministic distribution that always takes s.

A policy π chooses an action a based on the current state $s \in \mathcal{S}$ and the time step $h \in [H]$. Formally, $\pi = \{\pi_h\}_{h=1}^H$ where for each $h \in [H]$, $\pi_h : \mathcal{S} \to \mathcal{A}$ maps a given state to an action. Given an MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$, a policy π induces a (random) trajectory

$$(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_H, a_H, r_H), s_{H+1},$$

where $s_1 \sim \mu$, $a_1 = \pi_1(s_1)$, $r_1 \sim R(s_1, a_1)$, $s_2 \sim P(s_1, a_1)$, $a_2 = \pi_2(s_2)$, etc. For a policy π and $h \in [H]$, we use μ_h^π to denote the marginal distribution of s_h under π, i.e.,

$$\mu_h^\pi(s) = \Pr[s_h = s \mid \pi].$$

An important concept in RL is the Q-function. Given a policy π, a level $h \in [H]$ and a

state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, the Q-function is defined as

$$Q_h^\pi(s, a) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, a_h = a, \pi\right].$$

Similarly, the value function of a given state $s \in \mathcal{S}$ is defined as

$$V_h^\pi(s) = \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'} \mid s_h = s, \pi\right].$$

For a policy π, we write $V_M^\pi = \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]$ to denote its value in an MDP M, i.e., the expected total reward of π. We omit M from the subscript when it is clear from the context. For a given MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ and an integer H', for a given policy π, we define $V_{M,H}^\pi$ to be $V_{M'}^\pi$ where $M' = (\mathcal{S}, \mathcal{A}, P, R, H', \mu)$.

Throughout the thesis, we use π^* to denote a policy that maximizes V^π. It is well-known (see e.g. [67]) that the optimal value of M can be achieved by a deterministic policy, and hence, we only consider deterministic policies. For notational convenience, we also write $Q_h^*(s, a) = Q_h^{\pi^*}(s, a)$ and $V_h^*(s) = V_h^{\pi^*}(s)$.

Here we discuss four possible query models when interacting with an MDP.

- Online RL: In the Online RL model, the agent can only interact with the MDP by choosing actions and observe the next state and the reward.
- Offline RL: In the Offline RL model, the agent does not have direct access to the MDP and instead is given access to data distributions $\{\mu_h\}_{h=1}^H$ where for each $h \in [H]$, $\mu_h \in \Delta(\mathcal{S} \times \mathcal{A})$. The inputs of the agent are H datasets $\{D_h\}_{h=1}^H$, and for each $h \in [H]$, D_h consists i.i.d. samples of the form $(s, a, r, s') \in \mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S}$ tuples, where $(s, a) \sim \mu_h$, $r \sim R(s, a)$, $s' \sim P(s, a)$.
- Generative Model: Compared to the Online RL model, a stronger query model assumes the agent can transit to any state [42, 44, 76]. This query model is available in certain robotic applications where one can control the robot to reach the target state.
- Known Transition: Another query model considered here is that the agent can not only transit to any state, but also knows the whole transition operator. In this model, only the reward is unknown.

Now we discuss two possible goals in RL.

Policy Optimization. In the policy optimization problem, the goal is to find a policy π that maximizes the expected total reward $\mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]$ while minimizing the number of samples queried. We say a policy π is ε-optimal if

$$\mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right] \geq \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi^*\right] - \varepsilon.$$

Policy Evaluation. In the policy evaluation problem, the agent is given a policy π, and the goal is to estimate the value of π, i.e., V^π, while minimizing the number of samples queried.

Part I

RL with Long Planning Horizons

Chapter 3

RL with Long Planning Horizons

3.1 Introduction

Long horizons, along with the state dependent transitions, is the differentiator between RL problems and simpler contextual bandit problems. In RL, actions taken at early stages could substantially impact the future. In contextual bandit problems, the action taken at each time step is independent of the future. Jiang and Agarwal [37] proposed to study this distinction by examining how the sample complexity depends on the horizon length in a finite horizon episodic MDP. Clearly, as the horizon H grows, we will observe more samples in each episode. To appropriately measure the sample complexity, we consider a normalized notion: we are interested in the *number of episodes* it takes to provably discover an ε-optimal policy, where the value is measured by the *normalized* cumulative reward in each episode (i.e., values are normalized to be bounded between 0 and 1). Here, all existing sample complexity upper bounds depend polynomially on the horizon H, while lower bounds do not provide any dependence on H. Motivated by these observations, in a COLT 2018 open problem, Jiang and Agarwal [37] conjectured a sample complexity lower bound with linear dependence on the horizon, which is consistent with all existing upper bounds. In other words, the conjecture is that, even when the values are appropriately normalized, long horizon RL is more difficult than short horizon RL.

In this chapter we resolve this question, with, perhaps surprisingly, *negative* answers. Here we give an informal version of our first result.

Theorem 3.1.1. *Suppose the reward values satisfy $r_h \geq 0$ and $\sum_{h=1}^{H} r_h \leq 1$ almost surely. Given a target accuracy $0 < \varepsilon < 1$, there is an algorithm that returns an ε-optimal policy with probability at least $1 - \delta$ by sampling at most*

$$O\left(|\mathcal{S}|^3|\mathcal{A}|^3 \log^2 H/\varepsilon^3 \log(|\mathcal{S}||\mathcal{A}|/\varepsilon) \cdot \left(|\mathcal{S}|^2|\mathcal{A}| \log(H|\mathcal{S}|/\varepsilon) + \log(1/\delta)\right)\right)$$

episodes.

Importantly, this sample complexity scales only *logarithmically* with H. Although the conjecture in [37] has been refuted by Theorem 3.1.1, it is still unclear if tabular RL is possible with a sample complexity that is completely independent of the planning horizon. We further develop an algorithm whose sample complexity is completely independent of the planning horizon, at the cost of worse dependence on the number of states and actions.

Theorem 3.1.2. *Suppose the reward values satisfy $r_h \geq 0$ and $\sum_{h=1}^{H} r_h \leq 1$ almost surely. Given a target accuracy $0 < \varepsilon < 1$, there is an algorithm that returns an ε-optimal policy with probability at least $1 - \delta$ by sampling at most*

$$(|\mathcal{S}||\mathcal{A}|)^{O(|\mathcal{S}|)} \cdot \log(1/\delta)/\varepsilon^5$$

episodes.

In the context of the discussion in [37], these results suggest that perceived differences between long horizon RL and short horizon RL are not attributable to the horizon dependence, at least in a minimax sense.

3.2 Preliminaries

Notations. Throughout this chapter, for a random variable X and a real number $\varepsilon \in (0, 1]$, its ε-quantile $\mathcal{Q}_\varepsilon(X)$ is defined so that

$$\mathcal{Q}_\varepsilon(X) = \sup\{x \mid \Pr[X \geq x] \geq \varepsilon\}.$$

For a policy π, we define

$$\mathcal{Q}_\delta^\pi(s, a) = \mathcal{Q}_\delta \left[\sum_{t=1}^{H} \mathbb{I}[(s, a) = (s_t, a_t)] \right]$$

to be the δ-quantile of the visitation frequency of a state-action pair (s, a), where

$$(s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}$$

is a random trajectory induced by executing π.

Markov Chains. Let $C = (\mathcal{S}, P, \mu)$ be a Markov chain where $\mathcal{S}$ is the state space, $P : \mathcal{S} \to \Delta(\mathcal{S})$ is the transition operator and $\mu \in \Delta(\mathcal{S})$ is the initial state distribution. A Markov chain C induces a sequence of random states

$$s_1, s_2, \ldots$$

where for each $s_1 \sim \mu$ and $s_{h+1} \sim P(s_h)$ for each $h \geq 1$.

Stationary Policies. For the sake of the analysis, we shall also consider stationary policies. A stationary deterministic policy π chooses an action a solely based on the current state $s \in \mathcal{S}$, i.e, $\pi_1 = \pi_2 = \ldots = \pi_H$. We use Π_{st} to denote the set of all stationary policies. Note that $|\Pi_{\mathsf{st}}| = |\mathcal{A}|^{|\mathcal{S}|}$.

For an MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ and a stationary policy $\pi : \mathcal{S} \to \mathcal{A}$, we use $M^\pi = (\mathcal{S}, P^\pi, \mu)$ to denote the Markov chain induced by M and π, where the transition operator P^π is defined so that

$$P^\pi(s' \mid s) = P(s' \mid s, \pi(s)).$$

Assumption on Rewards. Below, we introduce the *bounded total reward assumption*.

Assumption 3.2.1 (Bounded Total Reward)**.** *For any policy π, and a random trajectory*

$$(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_H, a_H, r_H), s_{H+1}$$

induced by π, we have $r_h \in [0, 1]$ for all $h \in [H]$, and

$$\sum_{h=1}^{H} r_h \leq 1$$

almost surely.

As discussed in [37], this assumption is more general than the standard assumption where $r_h \in [0, 1/H]$ for all $h \in [H]$.

The above assumption in fact implies a very interesting consequence.

Lemma 3.2.1. *Under Assumption 3.2.1, for any $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ with $H \geq |\mathcal{S}|$, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, if there exists a (possibly non-stationary) policy π such that for the random trajectory*

$$(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_H, a_H, r_H), s_{H+1}$$

induced by executing π in M, we have

$$\Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s_h, a_h) = (s, a)] > 1\right] \geq \varepsilon$$

for some $\varepsilon > 0$, then $R(s, a) \leq 2|\mathcal{S}|/H$ almost surely and therefore $\mathbb{E}[(R(s, a))^2] \leq 4|\mathcal{S}|^2/H^2$.

Proof. By the assumption, there exists a trajectory

$$(s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}$$

such that there exists $1 \leq h_1 < h_2 \leq H$ with $s_{h_1} = s_{h_2}$. Moreover,

$$\mu(s_1) \prod_{h=1}^{h_2-1} P(s_{h+1} \mid s_h, a_h) \geq \varepsilon > 0.$$

We may assume $h_1 \leq |\mathcal{S}|$ and $h_2 - h_1 \leq |\mathcal{S}|$, since otherwise we can replace sub-trajectories that start and end with the same state by that state, and the resulting trajectory still appears with strictly positive probability. Now consider the policy $\widehat{\pi}$ which is defined so that for each $h < h_1$, $\widehat{\pi}_h(s_h) = a_h$ and for each $0 \leq t < h_2 - h_1$,

$$\widehat{\pi}_{h_1+t}(s_{h_1+t}) = \widehat{\pi}_{h_1+(h_2-h_1)+t}(s_{h_1+t}) = \widehat{\pi}_{h_1+2(h_2-h_1)+t}(s_{h_1+t}) = \cdots = a_{h_1+t},$$

i.e., repeating the trajectory's actions in $[h_1, h_2]$ indefinitely. $\widehat{\pi}$ is defined arbitrarily for other states and time steps.

By executing $\widehat{\pi}$, with strictly positive probability, (s, a) is visited for $\lfloor H/|\mathcal{S}| \rfloor \geq H/(2|\mathcal{S}|)$ times. Therefore, by Assumption 3.2.1, $R(s, a) \leq 2|\mathcal{S}|/H$ with probability 1 and thus

$$\mathbb{E}[(R(s, a))^2] \leq 4|\mathcal{S}|^2/H^2.$$

□

Discounted Markov Decision Processes. We also introduce another variant of MDP, discounted MDP, which is specified by $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$, where $\gamma \in (0, 1)$ is a discount factor and all other components have the same meaning as in an episodic MDP. The difference between a discounted MDP and an episodic MDP is that discounted MDPs have an infinite horizon length, i.e., the length of a trajectory can be infinite. To measure the value of a policy π in a discounted MDP, suppose π induces a random trajectory

$$(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots,$$

we define

$$V^{\pi}_{M,\gamma} = \mathbb{E}\left[\sum_{h=1}^{\infty} \gamma^{h-1} r_h \mid \pi\right]$$

as the discounted value of π. Throughout this chapter, for a (discounted or episodic) MDP $M = (\mathcal{S}, \mathcal{A}, P, R, \cdot, \mu)$, we define $V^{\pi}_{M,H}$ to be the value of π in $(\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ and $V^{\pi}_{M,\gamma}$ to be the value of π in $(\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$.

3.3 Technical Overview

3.3.1 Technical Overview of Theorem 3.1.1

An ε-net For Non-Stationary Policies. We first construct a set of polices Π which contains an ε-optimal policy for any MDP. Importantly, the size of Π satisfies $|\Pi| = (H/\varepsilon)^{\mathrm{poly}(|\mathcal{S}||\mathcal{A}|)}$, which is acceptable since the overall sample complexity of our algorithm depends only logarithmically on $|\Pi|$. To define such a set of policies, we consider all discretized MDPs whose transition probabilities and reward values are integer multiples of $\mathrm{poly}(\varepsilon/(|\mathcal{S}||\mathcal{A}|H))$. Clearly, there are most $(H/\varepsilon)^{\mathrm{poly}(|\mathcal{S}||\mathcal{A}|)}$ such discretized MDPs, and for each discretized MDP M, we add an optimal policy of M into Π. It remains to show that for any M, there exists a policy $\pi \in \Pi$ which is an ε-optimal policy of M. This can be seen since there exists a discretized MDP $\widehat{M}$ whose transition probabilities and reward values are close enough to those of M, and by standard perturbation analysis, it can be easily shown that an optimal policy of $\widehat{M}$ is an ε-optimal policy of M.

The Trajectory Synthesis Method. Now we show how to evaluate values of all policies in the policy set Π constructed above by sampling at most $\mathrm{poly}(|\mathcal{S}|, |\mathcal{A}|, 1/\varepsilon, \log|\Pi|, \log H)$ episodes. To achieve this goal, we design a trajectory simulator, which, for every policy in the set, either interacts with the environment to collect trajectories, or simulates trajectories using collected samples. In either case, the simulator obtains trajectories of the policy with distribution close enough to those sampled by interacting with the environment. The most natural idea is to collect trajectories for each policy π separately by interacting with the environment. This method, although is guaranteed to output "true" trajectories for every policy, has sample complexity at least linear in the size of the policy set $|\Pi|$ and is thus insufficient for our goal. Another possible way to evaluate policies is to build an empirical model (an estimation of transition probability and reward function) and evaluate policies on the empirical model (or to build a trajectory tree as in [43]). However, we do not know how to deal with the dependency issue in building the

empirical model and to prove a sample complexity bound that scales logarithmically with the planning horizon. The analysis based on performance difference lemma can lead to polynomial dependency on the planning horizon [42].

Reuse Samples. A key observation is that once we obtain a trajectory for a policy by interacting with the environment, samples collected during this process can be used to simulate trajectories for other policies. To better illustrate this idea, we use $\Pi_{\mathcal{D}}$ to denote the set of policies for which we have obtained trajectories by interacting with the environment, and denote

$$\mathcal{D}_{s,a} = \left[\left(s^{(1)}_{(s,a)}, r^{(1)}_{(s,a)}\right), \left(s^{(2)}_{(s,a)}, r^{(2)}_{(s,a)}\right), \ldots\right]$$

to be the sequence of samples obtained from $P(s,a)$ and $R(s,a)$. These samples are sorted in chronological order. Suppose that now we are given a new policy π and for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, $|\mathcal{D}^{(t)}_{s,a}| \geq H$. Then it is easy to simulate a trajectory for π using the set of samples $\{\mathcal{D}_{s,a}\}_{s\in\mathcal{S},a\in\mathcal{A}}$. Indeed, we start from state s_1 and set (s_2, r_2) to be the first pair in $\mathcal{D}_{s_1,\pi_1(s_1)}$, and then set (s_3, r_3) to be the first pair in $\mathcal{D}_{s_2,\pi_2(s_2)}$ that has not been used, etc. In general, suppose we are at state s_h for some $h < H$, we set (s_{h+1}, r_{h+1}) to be the first pair in $\mathcal{D}_{s_h,\pi_h(s_h)}$ that has not been used. Note that such a procedure generates a trajectory for π with exactly the same distribution as that generated by interacting with the environment.

Avoid Unnecessary Sampling. We have described the approach to reuse samples in the above paragraph. Nevertheless, there is a problem intrinsic to the above approach: if the process of simulating a policy π fails (i.e., some $(s_h, \pi_h(s))$ has been visited $j \leq H$ times but $|\mathcal{D}_{s_h,\pi_h(s)}| < j$), should we interact with the environment to generate a trajectory or simply claim failure? Note that claiming failure is acceptable as long as the overall failure probability is small.

In order to decide when to interact with the environment, we design a procedure to estimate the probability of simulation failure. If the failure probability is already small enough, there is no need to interact with the environment. Otherwise, we interact with the environment to obtain a trajectory. To bound the overall sample complexity, one key observation is that if the failure probability is large, then the policy will visit some state-action pair more frequently than all existing policies. In the formal analysis, we make this intuition rigorous by designing a potential function to measure the overall progress made by our algorithm.

3.3.2 Technical Overview of Theorem 3.1.2

To introduce the high-level ideas, we first start with the simpler setting, the generative model, where exploration is not a concern. We then switch to the more challenging RL setting, where we need to carefully design policies to explore the state-action space so that a good policy can be learned. For simplicity, throughout the discussion in this section, we assume $|\mathcal{S}|$, $|\mathcal{A}|$ and $1/\varepsilon$ are all constants.

Algorithm and Analysis in the Generative Model. Our algorithm in the generative model is conceptually simple: for each state-action pair (s,a), we draw $O(H)$ samples from $P(s,a)$

and $R(s,a)$ and then return the optimal policy with respect to the empirical model $\widehat{M}$ which is obtained by using the empirical estimators for P and R (denoted as $\widehat{P}$ and $\widehat{R}$). Here for simplicity, we assume $R = \widehat{R}$ which allows us to focus on the estimation error induced by the transition probabilities. Moreover, we assume that P differs from $\widehat{P}$ only for a single state-action pair (s,a). To further simplify the discussion, we assume that there are only two different states on the support of $P(s' \mid s,a)$ (say s_1 and s_2).

In order to prove the correctness of the algorithm, we show that for any policy π, the value of π in the empirical model $\widehat{M}$ is close to that in the true model M. However, standard analysis based on dynamic progarmming shows that the difference between the value of π in $\widehat{M}$ and that in M could be as large as H times the estimation error on $P(s,a)$, which is clearly insufficient for obtaining an algorithm which uses $O(1)$ batches of queries. Our main idea here is to show that for most trajectories T, the probability of T in the empirical model $\widehat{M}$ is a *multiplicative approximation* to that in the true model M with constant approximation ratio.

To establish the multiplicative approximation guarantee, our observation is that one should consider s_1 and s_2, the two states on the support of $P(s,a)$, as a whole. To see this, consider the case where $P(s_1 \mid s,a) = P(s_2 \mid s,a) = 1/2$. Again, the additive estimation errors on both $P(s_1 \mid s,a)$ and $P(s_2 \mid s,a)$ are roughly $O(1/\sqrt{H})$. Now, consider a trajectory that visits both (s,a,s_1) and(s,a,s_2) for $H/2$ times. Note that the multiplicative approximation ratio between $\widehat{P}(s' \mid s,a)^{H/2}$ and $P(s' \mid s,a)^{H/2}$ could be as large as $\exp(\sqrt{H})$, for both $s' = s_1$ and $s' = s_2$. However, since $\widehat{P}(s_1 \mid s,a) + \widehat{P}(s_2 \mid s,a) = 1$ as the empirical estimator $\widehat{P}(s,a)$ is still a probability distribution, it must be the case that $\widehat{P}(s_1 \mid s,a)/P(s_1 \mid s,a) = 1 - 2\delta$ and $\widehat{P}(s_2 \mid s,a)/P(s_2 \mid s,a) = 1 + 2\delta$ where $\delta = P(s_1 \mid s,a) - \widehat{P}(s_1 \mid s,a)$ and thus $|\delta| \leq O(1/\sqrt{H})$. Since $(1+2\delta)^{H/2}(1-2\delta)^{H/2} = (1-4\delta^2)^{H/2}$ is a constant, $(\widehat{P}(s_1 \mid s,a))^{H/2}(\widehat{P}(s_2 \mid s,a))^{H/2}$ is a constant factor approximation to the true probability $(P(s_1 \mid s,a))^{H/2}(P(s_2 \mid s,a))^{H/2}$ due to cancellation.

In our analysis, to formalize the above intuition, for each trajectory T, we take T into consideration only when $|m_T(s,a,s') - P(s' \mid s,a) \cdot m_T(s,a)| \leq O(\sqrt{P(s' \mid s,a) \cdot H})$ for both $s' = s_1$ and $s' = s_2$. Here $m_T(s,a)$ is the number of times that (s,a) is visited on T and $m_T(s,a,s')$ is the number of times that (s,a,s') is visited on T. By Chebyshev's inequality, we only ignore a small subset of trajectories whose total probability can be upper bounded by a constant. For the remaining trajectories, it can be shown that $\widehat{P}(s_1 \mid s,a)^{m_T(s,a,s_1)} \cdot \widehat{P}(s_2 \mid s,a)^{m_T(s,a,s_2)}$ is a constant factor approximation to $P(s_1 \mid s,a)^{m_T(s,a,s_1)} \cdot P(s_2 \mid s,a)^{m_T(s,a,s_2)}$ so long as $|\widehat{P}(s,a,s') - P(s,a,s')| \leq O(\sqrt{P(s,a,s')/H})$ for both $s' = s_1$ and $s' = s_2$ due to the cancellation mentioned above. Note that using $O(H)$ samples, $|\widehat{P}(s,a,s') - P(s,a,s')| \leq O(\sqrt{P(s,a,s')/H})$ holds only when $P(s,a,s') \geq \Omega(1/H)$. On the other hand, we can also ignore trajectories that visit (s,a,s') with $P(s,a,s') \leq O(1/H)$ since such trajectories have negligible cumulative probability by Markov's inequality.

The above analysis can be readily generalized to handle perturbation on the transition probabilities of multiple state-action pairs, and to handle the case when the transition operator $P(\cdot \mid s,a)$ is not supported on two states. In summary, by using $O(H)$ samples for each state-action pair (s,a), the empirical model $\widehat{M}$ provides a constant factor approximation to the probabilities of all trajectories, except for a small subset of them whose cumulative probability can be upper

bounded by a constant. Hence, for all policies, the empirical model provides an accurate estimate to its value and thus, the optimal policy with respect to the empirical model is near-optimal.

Exploration by Stationary Policies. In the discussion above, we heavily rely on the ability of the generative model to obtain $\Omega(H)$ samples for each state-action pair. However, for the RL setting, it is not possible to reach every state-action pair freely. Although each trajectory contains H state-action-state tuples (corresponding to a batch of queries in the generative model), these samples may not cover states that are crucial for learning an optimal policy. Indeed, one could use all possible deterministic non-stationary policies to collect samples, which shall then cover the whole state-action space. Unfortunately, such a naïve method introduces a dependence on the number of non-stationary policies which is exponential in H. The sample complexity of other existing methods in the literature also inevitably depends on H as their sample complexity intrinsically depends on the number of non-stationary policies.

In this work, we adopt a completely different approach for exploration. Our new idea is to show that if there exists a non-stationary policy that visits (s, a) for f times in expectation, then there exists a stationary policy that visit (s, a) for $f/\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$ times in expectation. If the above claim is true, then intuitively, one can simply enumerate all stationary policies and sample $\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$ trajectories using each of them to obtain f samples of (s, a). Note that there are only $|\mathcal{A}|^{|\mathcal{S}|}$ stationary policies, which is completely independent of H. In order to prove the above claim, we show that for any stationary policy π, its value in the infinite-horizon discounted setting is close to that in the finite-horizon undiscounted setting (up to a factor of $\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$) by using a properly chosen discount factor. Note that this implies the correctness of the above claim since there always exists a stationary optimal policy in the infinite-horizon discounted setting.

In order to show the value of a stationary policy in the infinite-horizon discounted setting is close to that in the finite-horizon setting, we study reaching probabilities in time invariant Markov chains. In particular, we show that in a time invariant Markov chain, for any $H \geq |\mathcal{S}|$, the probability of reaching a specific state s within H steps is close the probability of reaching s within $4H$ steps, up to a factor of $\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$. Previous literature on time invariant Markov chains mostly focus on the asymptotic behavior, and as far as we are aware, we are the first to prove the above claim. Note that this claim directly establishes a connection between the value of a stationary policy in the infinite-horizon discounted setting and that in the finite-horizon setting. Moreover, as a direct consequence of the above claim, we can show that if $H > 2|\mathcal{S}|$, the value of a stationary policy within H steps is close to that of the same policy within $H/2$ steps, up to a factor of $\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$. This consequence is crucial for later parts of the analysis.

From Expectation to Quantile. The above analysis shows that if there exists a non-stationary policy that visits (s, a) for f times in expectation, then our algorithm, which uses all stationary policies to collect samples, visits (s, a) for $f/\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$ times in expectation. However, this does not necessarily mean that one can obtain f samples of (s, a) by sampling $\exp(O(|\mathcal{S}|\log|\mathcal{S}|))$ trajectories using our algorithm with good probability. To see this, consider the case when our policy visits (s, a) for H times with probability $1/\sqrt{H}$ and does not visit (s, a) with probability $1 - 1/\sqrt{H}$. In this case, our policy may not obtain even a single sample

for (s,a) unless one rollouts the policy for $O(\sqrt{H})$ times. Therefore, instead of obtaining a visitation frequency guarantee which holds in expectation, it is more desirable to obtain a visitation frequency guarantee that holds with good probability.

To resolve this issue, we establish a connection between the expectation and the ε-quantile of the visitation frequency of a stationary policy. We note that such a connection could not hold without any restriction. To see this, consider a policy that visits (s,a) for H times with probability $\varepsilon/2$. In this case, the expected visitation frequency is $\varepsilon H/2$ while the ε-quantile is zero. On the other hand, suppose the initial state $s_1 = s$ almost surely, then such a connection is easy to establish by using the Martingale Stopping Theorem. In particular, we show that if there exists a non-stationary policy that visits (s,a) for f times with probability ε, then there exists a stationary policy that visits (s,a) for $\varepsilon f/\exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right)$ times with constant probability, when the initial state $s_1 = s$ almost surely.

In general, the initial state s_1 comes from a distribution μ and could be different from s with high probability. To tackle this issue, in our algorithm, we simultaneously enumerate two stationary policies π_1 and π_2. π_1 should be thought as the policy that visits (s,a) with highest probability within $H/2$ steps starting from the initial state distribution μ, and π_2 should be thought as the policy that maximizes the ε-quantile of the visitation frequency of (s,a) within $H/2$ steps when $s_0 = s$. In our algorithm, we execute π_1 before (s,a) is visited for the first time, and switch to π_2 once (s,a) has been visited. Intuitively, we first use π_1 to reach (s,a) for the first time and then use π_2 to collect as many samples as possible. As mentioned above, the value of a stationary policy within H steps is close to the value of the same policy within $H/2$ steps, up to a factor of $\exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right)$. Thus, by sampling the above policy (formed by concatenating π_1 and π_2) for $\exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right)/\varepsilon^2$ times, we obtain at least f samples for (s,a), if there exists a non-stationary policy that visits (s,a) for f times with probability ε.

Perturbation Analysis in the RL Setting. By the above analysis, suppose $m(s,a)$ is the largest integer such that there exists a non-stationary policy that visits (s,a) with probability ε for $m(s,a)$ times, then our dataset contains $\Omega(m(s,a))$ samples of (s,a). However, $m(s,a)$ could be significantly smaller than H and therefore the perturbation analysis established in the generative model no longer applies here. For example, previously we show that if $|m_T(s,a,s') - P(s' \mid s,a)\cdot m_T(s,a)| \le O(\sqrt{P(s' \mid s,a)\cdot H})$, then $\widehat{P}(s_1 \mid s,a)^{m_T(s,a,s_1)}\cdot\widehat{P}(s_2 \mid s,a)^{m_T(s,a,s_2)}$ is a constant factor approximation to $P(s_1 \mid s,a)^{m_T(s,a,s_1)}\cdot P(s_2 \mid s,a)^{m_T(s,a,s_2)}$ when $|\widehat{P}(s,a,s') - P(s,a,s')| \le O(\sqrt{P(s,a,s')/H})$ for both $s' = s_1$ and $s' = s_2$. However, if $m(s,a) \ll H$, it is hopeless to obtain an estimate $\widehat{P}(s,a,s')$ with $|\widehat{P}(s,a,s') - P(s,a,s')| \le O(\sqrt{P(s,a,s')/H})$. Fortunately, our perturbation analysis still goes through so long as $m_T(s,a,s') \le P(s' \mid s,a)\cdot m_T(s,a) + O(\sqrt{P(s' \mid s,a)\cdot m(s,a)})$ and $|\widehat{P}(s,a,s') - P(s,a,s')| \le O(\sqrt{P(s,a,s')/m(s,a)})$, i.e., replacing all H appearances with $m(s,a)$.

The above analysis introduces a final subtlety in our algorithm. In particular, $m(s,a)$ in the empirical model $\widehat{M}$ could be significantly larger than that in the true model. On the other hand, the number of samples of (s,a) in our dataset is at most $O(m(s,a))$ where $m(s,a)$ is defined by the true model. This means the value estimated in the empirical model $\widehat{M}$ could be significantly larger than that in the true model M. To resolve this issue, we employ the principle of "pessimism in the face of uncertainty" and for each policy π, the estimated value of π is set

to be the lowest value among all models that lie the confidence set. Since the true model always lies in the confidence set, the estimated value is then guaranteed to be close to the true value.

3.4 Proof of Theorem 3.1.1

In this section, for the sake of presentation, we assume a fixed initial state s_1. When the initial state is sampled from a distribution μ, we may create a new state s_0 and set s_0 to be the initial state. We set $P(s_0, a) = \mu$ and $r(s_0, a) = 0$ for all $a \in \mathcal{A}$, and increase the planning horizon H by 1. By doing so, now s_1 is sampled from the initial state distribution μ.

3.4.1 An ε-net For Non-Stationary Policies

In this section, we construct a set of polices which contains a near-optimal policy for any MDP. To define these policies, we first define a set of MDPs.

Throughout this section, without loss of genearlity, we assume $1/\varepsilon$ is a positive integer. In general, we may decrease ε by a factor of at most two so that $1/\varepsilon$ is a positive integer.

The following definition is helpful in our analysis.

Definition 3.4.1. For an MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$, we say a pair $(s, h) \in \mathcal{S} \times [H]$ is *admissible* with respect to M if there exists a policy π such that $\Pr[s_h = s \mid \pi] > 0$.

Before we being our analysis, we prove the following property regarding admissible pairs.

Lemma 3.4.1. *For any admissible $(s, h) \in \mathcal{S} \times [H]$, for any $a \in \mathcal{A}$, the following hold:*

- $0 \le R(s, a) \le 1$ *almost surely;*
- $0 \le Q_h^\pi(s, a) \le 1$ *for any policy π;*
- $0 \le V_h^\pi(s) \le 1$ *for any policy π.*

Proof. Here we only prove $0 \le R(s, a) \le 1$. It can be similarly proved that $0 \le Q_h^\pi(s, a) \le 1$ and $0 \le V_h^\pi(s) \le 1$. Suppose $R(s, a) > 1$ or $R(s, a) < 0$ with non-zero probability. Since (s, h) is admissible, there exists a policy π such that $\Pr[s_h = s \mid \pi] > 0$. Consider the policy π' defined to be:

$$\pi'_{h'}(s) = \begin{cases} \pi_{h'}(s) & h' < h \\ a & h' \ge h \end{cases}.$$

Clearly, $r_h > 1$ or $r_h < 0$ with non-zero probability, which violates the assumption that $\sum_{h=1}^H r_h \in [0, 1]$ and $r_h \ge 0$ for all $h \in [H]$ almost surely. □

Definition 3.4.2 (Discretized MDPs)**.** For given $\mathcal{S}$, $\mathcal{A}$, H, s_1 and $\varepsilon > 0$, define $\mathcal{M}_\varepsilon$ to be the set of MDPs $M = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$ such that

- Rewards are deterministic and for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $R(s, a) \in \{0, \varepsilon, 2\varepsilon, 3\varepsilon, \ldots, 1\}$;
- For each $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, $P(s' \mid s, a) \in \{0, \varepsilon, 2\varepsilon, 3\varepsilon, \ldots, 1\}$;

The following lemma gives an upper bound on the size of $\mathcal{M}_\varepsilon$.

Lemma 3.4.2. $|\mathcal{M}_\varepsilon| \le (1/\varepsilon + 1)^{|\mathcal{S}|^2|\mathcal{A}| + |\mathcal{S}||\mathcal{A}|}$.

Proof. Since each $M \in \mathcal{M}_\varepsilon$ is uniquely defined by its R and P, below we count the number of possible R and P respectively.

Since rewards are deterministic and for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, $R(s,a) \in \{0, \varepsilon, 2\varepsilon, \ldots, 1\}$, there are $(1/\varepsilon+1)^{|\mathcal{S}||\mathcal{A}|}$ different rewards in total.

Since for each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, $P(s' \mid s,a) \in \{0, \varepsilon, 2\varepsilon, \ldots, 1\}$, there are at most $(1/\varepsilon+1)^{|\mathcal{S}|^2|\mathcal{A}|}$ different transitions in total.

Therefore, $|\mathcal{M}_\varepsilon| \le (1/\varepsilon+1)^{|\mathcal{S}|^2|\mathcal{A}|+|\mathcal{S}||\mathcal{A}|}$.

Definition 3.4.3 (ε-net for Non-stationary Policies)**.** For given $\mathcal{S}$, $\mathcal{A}$, H and $\varepsilon > 0$, define Π_ε to be the set of polices such that

$$\Pi_\varepsilon = \{\pi_M \mid \pi_M \text{ is an optimal policy for } M \in \mathcal{M}_\varepsilon\}.$$

For each $M \in \mathcal{M}_\varepsilon$, when M has multiple optimal policies, we add an arbitrary one to Π_ε.

By construction of Π_ε and Lemma 3.4.2, it is clear that $|\Pi_\varepsilon| \le (1/\varepsilon+1)^{|\mathcal{S}|^2|\mathcal{A}|+|\mathcal{S}||\mathcal{A}|}$ □

Now we prove that for any MDP M, there is a near-optimal policy $\pi \in \Pi_\varepsilon$.

Lemma 3.4.3. *For any MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, s_1)$, there exists $\pi \in \Pi_\varepsilon$ such that π is $8H|\mathcal{S}|\varepsilon$-optimal.*

Proof. We first show that there exists $\widehat{M} = \left(\mathcal{S}, \mathcal{A}, \widehat{P}, \widehat{R}, H, s_1\right) \in \mathcal{M}_\varepsilon$ such that the following hold:

- For any $(s,h) \in \mathcal{S} \times [H]$ admissible with respect to M, for any $a \in \mathcal{A}$, $|\widehat{R}(s,a) - \mathbb{E}[R(s,a)]| \le \varepsilon$;
- For each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, $\left|P(s' \mid s,a) - \widehat{P}(s' \mid s,a)\right| \le \varepsilon$;
- For each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, if $P(s' \mid s,a) = 0$ then $\widehat{P}(s' \mid s,a) = 0$;

Below we construct such $\widehat{P}$ and $\widehat{R}$. By Lemma 3.4.1 we have $\mathbb{E}[R(s,a)] \in [0,1]$. Therefore, by setting $\widehat{R}(s,a)$ to be closest real number in $\{0, \varepsilon, 2\varepsilon, \ldots, 1\}$, we have $|\widehat{R}(s,a) - \mathbb{E}[R(s,a)]| \le \varepsilon$. Furthermore, for each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we set

$$P'(s' \mid s,a) = \min\{p \in \{0, \varepsilon, 2\varepsilon, \ldots, 1\} \mid p \ge P(s' \mid s,a)\}.$$

Notice that $P'(s,a)$ may not always be a probability distribution. Clearly $P'(s' \mid s,a) \ge P(s' \mid s,a)$ for each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $\sum_{s' \in \mathcal{S}} P'(s' \mid s,a) = 1 + k\varepsilon$ for some positive integer $0 \le k \le |\mathcal{S}|$. Now for each (s,a), we set $\widehat{P}(s' \mid s,a) = P'(s' \mid s,a) - \varepsilon$ for an arbitrary k states $s' \in \mathcal{S}$ with $P(s' \mid s,a) > 0$, and set $\widehat{P}(s' \mid s,a) = P'(s' \mid s,a)$ for all other states s'. Clearly, $P'(s,a)$ is a probability distrbution for any (s,a) and satisfies the desired property.

Now for any policy π, we use V^π to denote the V-value of π on MDP M, and use $\widehat{V}^\pi$ to denote the V-value of π on $\widehat{M}$. Q^π and $\widehat{Q}^\pi$ are defined analogously. We prove that $|V^\pi - \widehat{V}^\pi| \le 4|\mathcal{S}|H\varepsilon$ for any policy π inductively by the following induction hypothesis:

- $|V_h^\pi(s) - \widehat{V}_h^\pi(s)| \le (1 + (H-h)(|\mathcal{S}|+1))\varepsilon$ for any admissible (s,h);
- $|Q_h^\pi(s,a) - \widehat{Q}_h^\pi(s,a)| \le (1 + (H-h)(|\mathcal{S}|+1))\varepsilon$ for any admissible (s,h) and any $a \in \mathcal{A}$.

When $h = H$, $V_H^\pi(s) = Q_H(s, \pi(s)) = \mathbb{E}[R(s, \pi(s))]$ and $\widehat{V}_H^\pi(s) = \widehat{Q}_H(s, \pi(s)) = \widehat{R}(s, \pi(s))$. Therefore, the induction hypothesis holds when $h = H$ since $|\widehat{R}(s,a) - \mathbb{E}[R(s,a)]| \leq \varepsilon$.

Now we show the induction hypothesis holds for any $h < H$. For any $h < H$, consider any state s such that (s, h) is admissible. Notice that $V_h^\pi(s) = Q_h(s, \pi(s))$ and $\widehat{V}_h^\pi(s) = \widehat{Q}_h(s, \pi(s))$, and therefore $|V_h^\pi(s) - \widehat{V}_h^\pi(s)| = |Q_h^\pi(s, \pi(s)) - \widehat{Q}_h^\pi(s, \pi(s))|$. Furthermore,

$$Q_h^\pi(s,a) = \mathbb{E}[R(s,a)] + \sum_{s' \in \mathcal{S}} P(s' \mid s, a) V_{h+1}^\pi(s')$$

and

$$\widehat{Q}_h^\pi(s,a) = \widehat{R}(s,a) + \sum_{s' \in \mathcal{S}} \widehat{P}(s' \mid s, a) V_{h+1}^\pi(s').$$

Therefore,

$$\begin{aligned}
&\left|Q_h^\pi(s,a) - \widehat{Q}_h^\pi(s,a)\right| \\
\leq &\left|\mathbb{E}[R(s,a)] - \widehat{R}(s,a)\right| + \sum_{s' : P(s'|s,a) > 0} \left|P(s' \mid s,a) V_{h+1}^\pi(s') - \widehat{P}(s' \mid s,a) \widehat{V}_{h+1}^\pi(s')\right| \\
\leq &\varepsilon + \sum_{s' : P(s'|s,a) > 0} \left(\left|P(s' \mid s,a) - \widehat{P}(s' \mid s,a)\right| \cdot V_{h+1}^\pi(s') + \widehat{P}(s' \mid s,a) \cdot \left|V_{h+1}^\pi(s') - \widehat{V}_{h+1}^\pi(s')\right|\right) \\
\leq &(|\mathcal{S}| + 1)\varepsilon + \sum_{s' : P(s'|s,a) > 0} \widehat{P}(s' \mid s,a) \cdot \left|V_{h+1}^\pi(s') - \widehat{V}_{h+1}^\pi(s')\right| \quad (V_{h+1}^\pi(s') \leq 1 \text{ by Lemma 3.4.1}) \\
\leq &(|\mathcal{S}| + 1)\varepsilon + (1 + (H - (h+1))(|\mathcal{S}| + 1))\varepsilon \\
&\qquad (\textstyle\sum_{s' \in \mathcal{S}} \widehat{P}(s' \mid s,a) = 1 \text{ and induction hypothesis}) \\
= &(1 + (H - h)(|\mathcal{S}| + 1))\varepsilon.
\end{aligned}$$

Thus, we have

$$\left|V_1^\pi(s_1) - \widehat{V}_1^\pi(s_1)\right| \leq 4|\mathcal{S}|H\varepsilon.$$

Finally, consider any optimal policy $\widehat{\pi}$ of $\widehat{M}$ and any optimal policy π of M, we have

$$V_1^{\widehat{\pi}}(s_1) \geq \widehat{V}_1^{\widehat{\pi}}(s_1) - 4|\mathcal{S}|H\varepsilon \geq \widehat{V}_1^{\pi}(s_1) - 4|\mathcal{S}|H\varepsilon \geq V_1^{\pi}(s_1) - 8|\mathcal{S}|H\varepsilon.$$

Since $\widehat{\pi} \in \Pi_\varepsilon$, the lemma holds. □

3.4.2 Evaluating Policies

As shown in Section 3.4.1, there exists a set of policies Π such that for any MDP M, there exists a near-optimal policy $\pi \in \Pi$. In this section, we show how to approximately evaluate the values of all policies in Π using at most $\mathrm{poly}(|\mathcal{S}|, |\mathcal{A}|, 1/\varepsilon, \log|\Pi|, \log H)$ episodes. We formally describe our simulator in Section 3.4.2.1 and present its analysis in Section 3.4.2.2.

Algorithm 1 `SimAll`

Input: failure probability δ_{sim}, policy set Π, number of trajectories F
$\tau \leftarrow 16|\mathcal{S}|/\delta_{\mathsf{sim}} \cdot \log(4|\mathcal{S}|/\delta_{\mathsf{sim}})$
for $i \in [F]$ **do** ▷ Run F copies of Algorithm 2 in parallel
    Set SO_i to be the i-th independent copy of $\texttt{SimOne}(\tau)$ (Algorithm 2)
for $\pi \in \Pi$ **do**
    for $i \in [F]$ **do**
        $z_i^\pi \leftarrow \mathsf{SO}_i.\textsc{Simulate}(\pi)$
    if $\sum_{i=1}^{F} \mathbb{I}[z_i^\pi \text{ is } \texttt{Fail}] > 3\delta_{\mathsf{sim}}/2 \cdot F$ **then**
        for $i \in [F]$ **do**
            $z_i^\pi \leftarrow \mathsf{SO}_i.\textsc{Rollout}(\pi)$
return $\{z_i^\pi\}_{(i,\pi)\in[F]\times\Pi}$

3.4.2.1 The Trajectory Simulator

In this section, we describe our algorithm for simulating trajectories. The algorithm is formally presented in Algorithm 1 and Algorithm 2. Algorithm 2 receives a parameter τ and uses a replay buffer $\mathcal{D}$ to store samples. Formally, $\mathcal{D} = \{\mathcal{D}_{s,a}\}_{s\in\mathcal{S},a\in\mathcal{A}}$, where each $\mathcal{D}_{s,a}$ contains samples associated with state-action pair (s, a), i.e.,

$$\mathcal{D}_{s,a} = \left[(s_{s,a}^{(1)}, r_{s,a}^{(1)}), (s_{s,a}^{(2)}, r_{s,a}^{(2)}), \ldots\right]$$

and samples are sorted in chronological order. We also maintain $\Pi_{\mathcal{D}}$ in Algorithm 2 which is the set of policies used to generate $\mathcal{D}$. There are two subroutines in Algorithm 2. Subroutine SIMULATE takes an input policy π and outputs either `Fail` or a trajectory for policy π. Subroutine ROLLOUT takes an input policy π, samples τ episodes for π by interacting with the environment and stores all collected samples in the replay buffer $\mathcal{D}$. It also returns one of the τ trajectories sampled for for π. Moreover, whenever Subroutine ROLLOUT is invoked, samples in $\mathcal{D}$ are recollected so that independence among samples in the replay buffer $\mathcal{D}$ is ensured.

Algorithm 1 receives a failure probability δ_{sim} and a policy set Π as inputs. In Algorithm 1, we run F independent copies of Algorithm 2 in parallel. For each policy π, for the F independent copies of Algorithm 2, Algorithm 1 checks whether Subroutine SIMULATE returns `Fail` for too many times. If so, it calls Subroutine ROLLOUT for each copy of Algorithm 2 to collect samples and produce trajectories for π. Otherwise, it directly returns trajectories returned by Subroutine SIMULATE. The formal analysis of our algorithms will presented in Section 3.4.2.2.

3.4.2.2 Analysis

In this section, we present the formal analysis of Algorithm 1 and Algorithm 2. Before we present our analysis, we first introduce some necessary notations.

Definition 3.4.4. For any policy π, for any state-action pair $(s, a) \in \mathcal{S}\times\mathcal{A}$, define $f^\pi(s, a) \in [H]$ to be

$$f^\pi(s, a) = \sum_{h=1}^{H} \mathbb{I}[(s, a) = (s_h, a_h) \mid \pi].$$

Algorithm 2 `SimOne`

Input: number of repetitions τ
function SIMULATE(π):
  for $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
    Mark all elements in $\mathcal{D}_{s,a}$ as `unused`
  for $h \in \{1, 2, \ldots, H-1\}$ **do**
    if all elements in $\mathcal{D}_{s_h, \pi(s_h)}$ are marked as `used` **then**
      return `Fail`
    else
      Set (s_{h+1}, r_h) to be the first element in $\mathcal{D}_{s_h, \pi(s_h)}$ that is marked as `unused`
      Mark (s_{h+1}, r_h) (the first unused element in $\mathcal{D}_{s_h, \pi(s_h)}$) as `used`
  return $(s_1, \pi(s_1), r_1), (s_2, \pi(s_2), r_2), \ldots, (s_H, \pi(s_H), r_H)$
function ROLLOUT(π)
  Set $\mathcal{D}_{s,a}$ to be an empty sequence for all $(s, a) \in \mathcal{S} \times \mathcal{A}$
  $\Pi_{\mathcal{D}} \leftarrow \Pi_{\mathcal{D}} \cup \{\pi\}$
  for $\pi' \in \Pi_{\mathcal{D}}$ **do**
    Sample τ trajectories for π' by interacting with the environment
    Add all collected samples to $\mathcal{D}$
  return one of the τ trajectories sampled for π

I.e., $f^\pi(s, a)$ is the random variable which is the total number of times a trajectory induced by π visits (s, a).

We additionally define the following quantity to characterize the number times a state-action pair is visited by a set of policies. Intuitively, given a success probability δ, this quantity measures the maximum number of times a policy within a given policy set can visit a particular (s, a) pair.

Definition 3.4.5. For a set of policies Π, for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, define

$$\mu_\delta^\Pi(s, a) = \max \left\{ \lambda \mid \lambda \in [0, H], \max_{\pi \in \Pi} \Pr[f^\pi(s, a) \geq \lambda] \geq \delta \right\}.$$

Note that $\mu_\delta^\Pi(s, a)$ is always a non-negative integer since for any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, policy π and real number λ,

$$\Pr\left[f^\pi(s, a) \geq \lambda\right] = \Pr\left[f^\pi(s, a) \geq \lceil \lambda \rceil\right].$$

Our next lemma states that for some policy π, if `SimOne` fails with high probability, then there exists a state-action pair that π visits more frequently than all previous policies.

Lemma 3.4.4. *For a policy $\pi \in \Pi$, suppose Subroutine* SIMULATE *in Algorithm 2 returns `Fail` with probability at least δ_{sim} over the randomness of the generating process of the replay buffer $\mathcal{D}$. There exists $(s, a) \in \mathcal{S} \times \mathcal{A}$ such that*

$$\Pr\left[f^\pi(s, a) > \tau \cdot \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|} \cdot \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s, a)\right] \geq \frac{\delta_{\mathsf{sim}}}{2|\mathcal{S}|}$$

where $\Pi_{\mathcal{D}}$ is the set of policies used to generate $\mathcal{D}$.

Proof. Suppose for the sake of contradiction that for all $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$\Pr\left[f^{\pi}(s,a) > \tau \cdot \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|} \cdot \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)\right] < \frac{\delta_{\mathsf{sim}}}{2|\mathcal{S}|}.$$

Let us denote

$$\Gamma(s,a) = \tau \cdot \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|} \cdot \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a).$$

For all $s \in \mathcal{S}$, we have

$$\Pr\left[f^{\pi}(s,\pi(s)) > \Gamma(s,\pi(s))\right] < \frac{\delta_{\mathsf{sim}}}{2|\mathcal{S}|}.$$

Therefore, by a union bound over all states $\mathcal{S}$, with probability at least $1 - \delta_{\mathsf{sim}}/2$, for all states $s \in \mathcal{S}$,

$$f^{\pi}(s,\pi(s)) \le \Gamma(s,\pi(s)) \tag{3.1}$$

For each $s \in \mathcal{S}$, define $\mathcal{E}_s$ to be the event that

$$\mathcal{E}_s = \left\{\left|\mathcal{D}_{s,\pi(s)}\right| \ge \Gamma(s,\pi(s))\right\}.$$

By Definition 3.4.5, there exists a policy $\pi^*_s \in \Pi_{\mathcal{D}}$ such that

$$\Pr[f^{\pi^*_s}(s,\pi(s)) \ge \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,\pi(s))] \ge \delta_{\mathsf{sim}}/(2|\mathcal{S}|).$$

Now consider Line 16 in Subroutine ROLLOUT in Algortihm 2. Define

$$X_i = \begin{cases} 1 & \text{if } \sum_{h=1}^{H} \mathbb{I}[(s_h,a_h) = (s,\pi(s))] \ge \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,\pi(s)) \text{ for the } i\text{-th trajectory of } \pi^*_s \\ 0 & \text{otherwise} \end{cases}.$$

Note $X_1, \ldots, X_{\delta_{\mathsf{sim}}}$ are i.i.d. random variables. By definition, $\mathbb{E}[X_i] \ge \delta_{\mathsf{sim}}/(2|\mathcal{S}|)$. Therefore, since $\tau = 16|\mathcal{S}|/\delta_{\mathsf{sim}} \cdot \log(4|\mathcal{S}|/\delta_{\mathsf{sim}})$, by Chernoff bound,

$$\Pr\left[\sum_{i=1}^{\tau} X_i \le \tau \cdot \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|}\right] \le \exp\left(-\frac{\tau\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}{8}\right) \le \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|}.$$

Therefore,

$$\Pr[\mathcal{E}_s] \ge \Pr\left[\sum_{i=1}^{\tau} X_i \ge \tau \cdot \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|}\right] \ge 1 - \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|}.$$

It follows that with probability at least $1 - \delta_{\mathsf{sim}}/4$, for all $s \in \mathcal{S}$,

$$\left|\mathcal{D}_{s,\pi(s)}\right| \ge \Gamma(s,\pi(s)). \tag{3.2}$$

By a union bound over (3.1) and (3.2), with probability at least $1 - \frac{3\delta_{\mathsf{sim}}}{4}$, for all $s \in \mathcal{S}$,

$$\left|\mathcal{D}_{s,\pi(s)}\right| \ge \Gamma(s,\pi(s)) \ge f^{\pi}(s,\pi(s)),$$

in which case Subroutine ROLLOUT does not return `Fail`. This contradicts the assumption that Subroutine ROLLOUT returns `Fail` with probability at least δ_{sim}.

□

Now we discuss the implication of Lemma 3.4.4. Note that Algorithm 2 interacts with the environment to sample trajectories only when Subroutine SIMULATE fails with probability at least δ_{sim}. By Lemma 3.4.4, when Algorithm 2 interacts with the environment to sample trajectories, $\mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)$ doubles or changes from 0 to 1 for some $(s,a) \in \mathcal{S} \times \mathcal{A}$ since $\tau\delta_{\mathsf{sim}}/(4|\mathcal{S}|) > 2$. However, $\mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)$ is always upper bounded by H. Therefore, the total number of calls to Subrountine ROLLOUT in Algorithm 1 is upper bounded by $O(|\mathcal{S}||\mathcal{A}|\log H)$. Our next lemma guarantees that whenever Algorithm 1 invokes Subroutine ROLLOUT, the probability that Subroutine SIMULATE returns `Fail` is at least δ_{sim}, and when Subroutine ROLLOUT is not invoked, the probability that Subroutine SIMULATE returns `Fail` is at most $2\delta_{\mathsf{sim}}$.

Lemma 3.4.5. *Suppose $F \geq 24/\delta_{\mathsf{sim}} \cdot \log(2|\Pi|/\delta_{\mathsf{sim}})$. With probability at least $1 - \frac{\delta_{\mathsf{sim}}}{2|\Pi|}$, each time Line 8 in Algorithm 1 is executed, the following hold:*

- *when $\sum_{i=1}^{F} \mathbb{I}[z_i^\pi$ is `Fail`$] > 3\delta_{\mathsf{sim}}/2 \cdot F$, the probability that Subroutine* SIMULATE *returns `Fail` is at least δ_{sim} over the randomness of the generating process of the replay buffer $\mathcal{D}$;*
- *when $\sum_{i=1}^{F} \mathbb{I}[z_i^\pi$ is `Fail`$] \leq 3\delta_{\mathsf{sim}}/2 \cdot F$, the probability that Subroutine* SIMULATE *returns `Fail` is at most $2\delta_{\mathsf{sim}}$ over the randomness of the generating process of the replay buffer $\mathcal{D}$.*

Proof. Let $Y_i = \mathbb{I}[z_i^\pi$ is `Fail`$]$. Note that each time Subroutine ROLLOUT is invoked, all samples in $\mathcal{D}$ are recollected. Therefore, for any given time step of the algorithm, $\{Y_i\}_{i=1}^{F}$ are independent random variables.

If $\Pr[Y_i = 1] < \delta_{\mathsf{sim}}$, by Chernoff bound,

$$\Pr\left[\sum_{i=1}^{F} Y_i \geq 3\delta_{\mathsf{sim}}/2 \cdot F\right] \leq \exp(-\delta_{\mathsf{sim}}F/24) \leq \frac{\delta_{\mathsf{sim}}}{2|\Pi|}.$$

On the other hand, if $\Pr[Y_i = 1] \geq 2\delta_{\mathsf{sim}}$, by Chernoff bound,

$$\Pr\left[\sum_{i=1}^{F} Y_i \leq 3\delta_{\mathsf{sim}}/2 \cdot F\right] \leq \exp(-\delta_{\mathsf{sim}}F/16) \leq \frac{\delta_{\mathsf{sim}}}{2|\Pi|}.$$

Thus the lemma holds. □

Now we bound the overall sample complexity of the algorithm.

Lemma 3.4.6. *Suppose $F \geq 24/\delta_{\mathsf{sim}} \cdot \log(2|\Pi|/\delta_{\mathsf{sim}})$. Let $\Pi_{\mathcal{D}}$ be the set of policies maintained by Algorithm 2 before executing Line 14, and let $\widehat{\Pi}_{\mathcal{D}}$ be the set of policies maintained after executing Line 14, i.e., $\widehat{\Pi}_{\mathcal{D}} = \Pi_{\mathcal{D}} \cup \{\pi\}$. With probability at least $1 - \delta_{\mathsf{sim}}/(2|\Pi|)$, there exists $(s,a) \in \mathcal{S} \times \mathcal{A}$, such that*

$$\mu^{\widehat{\Pi}_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a) \geq \max\left(2 \cdot \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a), 1\right).$$

Proof. By Lemma 3.4.5, with probability at least $1 - \delta_{\mathsf{sim}}/(2|\Pi|)$, for the added policy π, the probability that Subroutine SIMULATE returns `Fail` is at least δ_{sim}. By Lemma 3.4.4, there exists $(s,a) \in \mathcal{S} \times \mathcal{A}$ such that

$$\Pr\left[f^\pi(s,a) > \tau \cdot \frac{\delta_{\mathsf{sim}}}{4|\mathcal{S}|} \cdot \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)\right] \geq \frac{\delta_{\mathsf{sim}}}{2|\mathcal{S}|}.$$

If $\mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a) = 0$, we have

$$\Pr[f^{\pi}(s,a) > 0] = \Pr[f^{\pi}(s,a) \geq 1] \geq \frac{\delta_{\mathsf{sim}}}{2|\mathcal{S}|}.$$

Otherwise, we have

$$\Pr\left[f^{\pi}(s,a) \geq 2 \cdot \mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)\right] \geq \frac{\delta_{\mathsf{sim}}}{2|\mathcal{S}|}.$$

□

Lemma 3.4.7. *Suppose $F \geq 24/\delta_{\mathsf{sim}} \log(2|\Pi|/\delta_{\mathsf{sim}})$. With probability at least $1 - \delta_{\mathsf{sim}}/2$, Algorithm 1 at most interacts*

$$O\left(\frac{|\mathcal{S}|}{\delta_{\mathsf{sim}}} \log(|\mathcal{S}|/\delta_{\mathsf{sim}}) \cdot |\mathcal{S}|^2 |\mathcal{A}|^2 \log^2 H \cdot F\right)$$

episodes with the environment.

Proof. Notice that our algorithm interacts with the environment only when Subroutine ROLLOUT in Algorithm 2 is invoked. By Lemma 3.4.6 and union bound, with probability at least $1 - \delta_{\mathsf{sim}}/2$, whenever Subroutine ROLLOUT is invoked, there exists $(s,a) \in \mathcal{S} \times \mathcal{A}$ such that $\mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)$ is increased from 0 to 1, or $\mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a)$ is increased by a factor of 2. Since $\mu^{\Pi_{\mathcal{D}}}_{\delta_{\mathsf{sim}}/(2|\mathcal{S}|)}(s,a) \leq H$, with probability at least $1 - \delta_{\mathsf{sim}}/2$, Subroutine ROLLOUT is invoked for at most $O(|\mathcal{S}||\mathcal{A}| \log H)$ times. Hence $|\Pi_{\mathcal{D}}| = O(|\mathcal{S}||\mathcal{A}| \log H)$. Finally, whenever Subroutine ROLLOUT is invoked, the algorithm samples at most $F|\Pi_{\mathcal{D}}|\tau$ trajectories by interacting with the environment. Therefore, with probability at least $1 - \delta_{\mathsf{sim}}/2$, the total number of trajectories sampled by the algorithm is upper bounded by $O(F\tau \cdot (|\mathcal{S}||\mathcal{A}| \log H)^2)$. □

3.4.3 The Algorithm

In this section we present our final algorithm. The algorithm description is given in Algorithm 7. Our algorithm invokes Algorithm 1 on the set of policies defined in in Definition 3.4.3 to obtain trajectories for each policy, and simply returns the policy with largest empirical cumulative reward. Now we give the formal analysis of our algorithm.

Lemma 3.4.8. *For each policy $\pi \in \Pi_{\varepsilon/(32H|\mathcal{S}|)}$, for the value $\widehat{r}(\pi)$ calculated in Line 7 of Algorithm 7, with probability at least $1 - \delta_{\mathsf{overall}}/(2|\Pi_{\varepsilon/(32H|\mathcal{S}|)}|)$,*

$$\left|\widehat{r}(\pi) - \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]\right| \leq 5\varepsilon/16.$$

Proof. For those policies $\pi \in \Pi_{\mathcal{D}}$, notice that $\{z_i^{\pi}\}_{i\in[F]}$ are sampled by interacting with the environment. Since all reward values are positive and cumulative reward is upper bounded by 1 almost surely, by Chernoff bound,

$$\Pr\left[\left|\widehat{r}(\pi) - \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]\right| \leq \varepsilon/8\right] \geq 1 - 2\exp(-F\varepsilon^2/64) \geq 1 - \delta/(2|\Pi_{\varepsilon/(32H|\mathcal{S}|)}|).$$

Algorithm 3 `Main`

Input: failure probability $\delta_{\mathsf{overall}}$, accuracy ε
Let $\Pi_{\varepsilon/(32H|\mathcal{S}|)}$ be the set of policies as defined in Definition 3.4.3
Invoke `SimAll` (Algorithm 2) with $\delta_{\mathsf{sim}} = \varepsilon/8$ and

$$F = \max\{64\log(4|\Pi_{\varepsilon/(32H|\mathcal{S}|)}|/\delta_{\mathsf{overall}})/\varepsilon^2, 192/\varepsilon\log(16|\Pi_{\varepsilon/(32H|\mathcal{S}|)}|/\varepsilon)\}$$

for each trajectory $z = (s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_H, a_H, r_H)$ returned by `SimAll` **do**
    Calculate $r(z) = \begin{cases} 0 & z \text{ is } \texttt{Fail} \\ \sum_{h=1}^{H} r_h & \text{otherwise} \end{cases}$
for $\pi \in \Pi_{\varepsilon/(32H|\mathcal{S}|)}$ **do**
    Calculate $\widehat{r}(\pi) = \frac{1}{F}\sum_{i\in[F]} r(z_i^\pi)$
return $\operatorname{argmax}_{\pi\in\Pi_{\varepsilon/(32H|\mathcal{S}|)}} \widehat{r}(\pi)$

For those policies $\pi \notin \Pi_{\mathcal{D}}$, notice that $\{z_i^\pi\}_{i\in[F]}$ have the same distribution as F independent trajectories sampled by interacting with the environment, except that at most $3\delta/2\cdot F = 3\varepsilon/16\cdot F$ trajectories are replaced with `Fail`. If all trajectories are independently sampled by interacting with the environment, by Chernoff bound, with probability at least $1 - \delta_{\mathsf{overall}}/(2|\Pi_{\varepsilon/(32H|\mathcal{S}|)}|)$,

$$\left|\widehat{r}(\pi) - \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]\right| \le \varepsilon/8.$$

Since cumulative reward is in $[0,1]$ almost surely, by replacing at most $3\varepsilon/16\cdot F$ trajectories with `Fail`, $\widehat{r}(\pi)$ is changed by at most $3\varepsilon/16$. Therefore, with probability at least $1 - \delta_{\mathsf{overall}}/(2|\Pi_{\varepsilon/(32H|\mathcal{S}|)}|)$,

$$\left|\widehat{r}(\pi) - \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]\right| \le 5\varepsilon/16.$$

□

Lemma 3.4.9. *With probability at least $1 - \delta_{\mathsf{overall}}/2$, Algorithm 7 returns an ε-optimal policy.*

Proof. By Lemma 3.4.3, there exists a $\varepsilon/4$-optimal policy $\pi' \in \Pi_{\varepsilon/(32H|\mathcal{S}|)}$. By Lemma 3.4.8 and a union bound over Π, with probability at least $1 - \delta_{\mathsf{overall}}/2$, for all policy $\pi \in \Pi_{\varepsilon/(32H|\mathcal{S}|)}$,

$$\left|\widehat{r}(\pi) - \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right]\right| \le 5\varepsilon/16.$$

Let π be the policy returned by algorithm. Conditioned on the event mentioned above, we have

$$\mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi\right] \ge \widehat{r}(\pi) - 5\varepsilon/16 \ge \widehat{r}(\pi') - 5\varepsilon/16 \ge \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi'\right] - 5\varepsilon/8 \ge \mathbb{E}\left[\sum_{h=1}^{H} r_h \mid \pi^*\right] - \varepsilon.$$

□

Our main result, Theorem 3.1.1 is a direct implication of Lemma 3.4.7 and Lemma 3.4.9.

3.5 Proof of Theorem 3.1.2

3.5.1 Properties of Stationary Policies

In this section, we prove several properties of stationary policies. In Section 3.5.1.1, we first prove properties regarding reaching probabilities in Markov chains, and then use them to prove properties for stationary policies in Section 3.5.1.2.

3.5.1.1 Reaching Probabilities in Markov Chains

Let $C = (\mathcal{S}, P, \mu)$ be a Markov chain. For a positive integer L and a sequence of states $T = (s_1, s_1, \ldots, s_L) \in \mathcal{S}^L$, we write

$$p(T, C) = \mu(s_1) \cdot \prod_{h=1}^{L-1} P(s_{h+1} \mid s_h)$$

to denote the probability of T in C. For a state $s \in \mathcal{S}$ and an integer $L \geq 0$, we also write

$$p_L(s, C) = \sum_{(s_1, s_1, \ldots, s_L) \in \mathcal{S}^L} p((s_1, s_2, \ldots, s_L, s), C)$$

to denote the probability of reaching s with exactly L steps.

Our first lemma shows that for any Markov chain C, for any sequence of L states T with $L > |\mathcal{S}|$, there exists a sequence of L' states T' with $L' \leq |\mathcal{S}|$ so that $p(T, C) \leq p(T', C)$.

Lemma 3.5.1. *Let $C = (\mathcal{S}, P, \mu)$ be a Markov chain. For a sequence of L states*

$$T = (s_1, s_1, \ldots, s_L) \in \mathcal{S}^L$$

with $L > |\mathcal{S}|$, there exists a sequence of L' states

$$T' = (s'_1, s'_2, \ldots, s'_{L'}) \in \mathcal{S}^{L'}$$

with $s'_{L'} = s_L$, $L' \leq |\mathcal{S}|$ and $p(T, C) \leq p(T', C)$.

Proof. By pigeonhole principle, since $L > |\mathcal{S}|$, there exists $0 \leq i < j < L$ such that $s_i = s_j$. Consider the sequence induced by removing $s_i, s_{i+1}, s_{i+2}, \ldots, s_{j-1}$ from T, i.e.,

$$T' = (s_1, s_2, \ldots, s_{i-1}, s_j, s_{j+1}, \ldots, s_L).$$

Since $s_i = s_j$, we have

$$p(T, C) = \mu(s_1) \cdot \prod_{h=1}^{L-1} P(s_{h+1} \mid s_h).$$

and

$$p(T', C) = \mu(s_1) \cdot \prod_{h=1}^{i-2} P(s_{h+1} \mid s_h) \cdot \prod_{h=j}^{L-1} P(s_{h+1} \mid s_h).$$

Therefore, we have $p(T, C) \leq p(T', C)$. We continue this process until the length is at most $|\mathcal{S}|$. □

Combining Lemma 3.5.1 with a simple counting argument directly implies the following lemma, which shows that $\sum_{h=0}^{4|\mathcal{S}|-1} p_h(s,C) \le \exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right) \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_h(s,C)$.

Lemma 3.5.2. *Let $C = (\mathcal{S}, P, \mu)$ be a Markov chain. For any $s \in \mathcal{S}$,*

$$\sum_{h=0}^{4|\mathcal{S}|-1} p_h(s,C) \le 4 \cdot |\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_h(s,C).$$

Proof. Consider a sequence of $L+1$ states $T = (s_1, s_2, \ldots, s_{L+1}) \in \mathcal{S}^{L+1}$ with $L \ge |\mathcal{S}|$ and $s_L = s$. By Lemma 3.5.1, there exists another sequence of L' states $T' = (s'_1, s'_2, \ldots, s'_{L'}) \in \mathcal{S}^{L'}$ with $s'_{L'} = s_{L+1} = s$ and $L' \le |\mathcal{S}|$ so that $p(T,C) \le p(T',C)$. Therefore

$$p(T,C) \le p(T',C) \le p_{L'-1}(s,C) \le \sum_{h=0}^{|\mathcal{S}|-1} p_h(s,C),$$

which implies

$$p_L(s,C) = \sum_{(s_1,s_2,\ldots,s_{L-1},s_L)\in\mathcal{S}^L} p((s_1,s_2,\ldots,s_{L-1},s_L,s),C) \le |\mathcal{S}|^L \sum_{h=0}^{|\mathcal{S}|-1} p_h(s,C).$$

Therefore,

$$\sum_{h=0}^{4|\mathcal{S}|-1} p_h(s,C) \le 4\cdot|\mathcal{S}|\cdot|\mathcal{S}|^{4|\mathcal{S}|-1} \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_h(s,C) = 4\cdot|\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_h(s,C).$$

□

By applying Lemma 3.5.2 in a Markov chain C' with modified initial state distribution and transition operator, we can also prove that

$$\sum_{h=0}^{4|\mathcal{S}|-1} p_{\beta h+\alpha}(s,C) \le \exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right) \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_{\beta h+\alpha}(s,C)$$

for any integer $\alpha \ge 0$ and integer $\beta \ge 1$.

Lemma 3.5.3. *Let $C = (\mathcal{S}, P, \mu)$ be a Markov chain. For any integer $\alpha \ge 0$ and integer $\beta \ge 1$, for any $s \in \mathcal{S}$,*

$$\sum_{h=0}^{4|\mathcal{S}|-1} p_{\beta h+\alpha}(s,C) \le 4 \cdot |\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_{\beta h+\alpha}(s,C).$$

Proof. We define a new Markov chain $C' = (\mathcal{S}, P', \mu')$ based on $C = (\mathcal{S}, P, \mu)$. The state space of C' is the same as that of C. The initial state distribution μ' is the same as the distribution of s_α in C, i.e., the distribution after taking α steps in C. The transition operator is defined so that taking one step in C' is equivalent to taking β steps in C, i.e.,

$$P'(s' \mid s) = \sum_{s_1,s_2,\ldots,s_{\beta-1}\in\mathcal{S}^{\beta-1}} P(s' \mid s_{\beta-1}) \cdot P(s_{\beta-1} \mid s_{\beta-2}) \cdot \cdots\cdot P(s_1 \mid s).$$

Clearly, for any state $s \in \mathcal{S}$, $p_L(s, C') = p_{\beta L+\alpha}(s, C)$. By using Lemma 3.5.2 in C', for any $s \in \mathcal{S}$, we have

$$\sum_{h=0}^{4|\mathcal{S}|-1} p_h(s, C') \leq 4 \cdot |\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_h(s, C'),$$

which implies

$$\sum_{h=0}^{4|\mathcal{S}|-1} p_{\beta h+\alpha}(s, C) \leq 4 \cdot |\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{|\mathcal{S}|-1} p_{\beta h+\alpha}(s, C).$$

□

Finally, Lemma 3.5.3 implies the main result of this section, which shows that for any $L \geq |\mathcal{S}|$,

$$\sum_{h=0}^{4L-1} p_h(s, C) \leq \exp\left(O(|\mathcal{S}| \log |\mathcal{S}|)\right) \cdot \sum_{h=0}^{L-1} p_h(s, C).$$

Lemma 3.5.4. *Let $C = (\mathcal{S}, P, \mu)$ be a Markov chain. For any $s \in \mathcal{S}$ and $L \geq |\mathcal{S}|$,*

$$\sum_{h=0}^{2L} p_h(s, C) \leq 4 \cdot |\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{L-1} p_h(s, C).$$

Proof. Clearly,

$$\sum_{h=0}^{L-1} p_h(s, C) \geq \sum_{h=0}^{\lfloor L/|\mathcal{S}| \rfloor \cdot |\mathcal{S}|-1} p_h(s, C) = \sum_{i=0}^{\lfloor L/|\mathcal{S}| \rfloor-1} \sum_{j=0}^{|\mathcal{S}|-1} p_{\lfloor L/|\mathcal{S}| \rfloor \cdot j+i}(s, C).$$

For each $0 \leq i < \lfloor L/|\mathcal{S}| \rfloor$, by Lemma 3.5.3, we have

$$\sum_{j=0}^{|\mathcal{S}|-1} p_{\lfloor L/|\mathcal{S}| \rfloor \cdot j+i}(s, C) \geq \frac{1}{4|\mathcal{S}|^{4|\mathcal{S}|}} \sum_{j=0}^{4|\mathcal{S}|-1} p_{\lfloor L/|\mathcal{S}| \rfloor \cdot j+i}(s, C).$$

On the other hand,

$$\sum_{h=0}^{2L} p_h(s, C) \leq \sum_{i=0}^{\lfloor L/|\mathcal{S}| \rfloor-1} \sum_{j=0}^{\lfloor (2L+1)/\lfloor L/|\mathcal{S}| \rfloor \rfloor-1} p_{\lfloor L/|\mathcal{S}| \rfloor \cdot j+i}(s, C).$$

Note that if $|\mathcal{S}| > L/2$, then $\lfloor (2L+1)/\lfloor L/|\mathcal{S}| \rfloor \rfloor = 2L+1 < 4|\mathcal{S}|$. Moreover, if $|\mathcal{S}| \leq L/2$, then we have $\lfloor L/|\mathcal{S}| \rfloor \geq 2L/3|\mathcal{S}|$, which implies

$$\lfloor (2L+1)/\lfloor L/|\mathcal{S}| \rfloor \rfloor \leq \lfloor (2L+1)/L \cdot 3|\mathcal{S}|/2 \rfloor \leq \lfloor 4|\mathcal{S}| \rfloor = 4|\mathcal{S}|.$$

Hence, we have

$$\sum_{h=0}^{2L} p_h(s, C) \leq \sum_{i=0}^{\lfloor L/|\mathcal{S}| \rfloor-1} \sum_{j=0}^{4|\mathcal{S}|-1} p_{\lfloor L/|\mathcal{S}| \rfloor \cdot j+i}(s, C) \leq 4 \cdot |\mathcal{S}|^{4|\mathcal{S}|} \cdot \sum_{h=0}^{L-1} p_h(s, C).$$

□

3.5.1.2 Implications of Lemma 3.5.4

In this section, we list several implications of Lemma 3.5.4 which would be crucial for the analysis in later sections.

Our first lemma shows that for any MDP M and any stationary policy π, for a properly chosen discount factor γ, $V^{\pi}_{M,\gamma}$ is a multiplicative approximation to $V^{\pi}_{M,H}$ with approximation ratio $\exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right)$.

Lemma 3.5.5. *For any MDP M and any stationary policy π, if $H \geq 2\ln(8\cdot|\mathcal{S}|^{4|\mathcal{S}|})$, by taking $\gamma = 1 - \frac{\ln(8\cdot|\mathcal{S}|^{4|\mathcal{S}|})}{H}$,*

$$\frac{1}{64\cdot|\mathcal{S}|^{8|\mathcal{S}|}}V^{\pi}_{M,H} \leq V^{\pi}_{M,\gamma} \leq 2V^{\pi}_{M,H}.$$

Proof.

$$\begin{aligned}
V^{\pi}_{M,\gamma} &= \sum_{s\in\mathcal{S}}\sum_{h=0}^{\infty}\gamma^h\cdot p_h(s,M^{\pi})\cdot\mathbb{E}[R(s,\pi(s))] \\
&\leq \sum_{s\in\mathcal{S}}\left(\sum_{h=0}^{H-1}p_h(s,M^{\pi}) + \sum_{i=1}^{\infty}\gamma^{H\cdot 2^{i-1}}\left(\sum_{h=0}^{2^i\cdot H-1}p_h(s,M^{\pi})\right)\right)\cdot\mathbb{E}[R(s,\pi(s))].
\end{aligned}$$

For each $i \geq 1$, by Lemma 3.5.4, for any $s\in\mathcal{S}$,

$$\begin{aligned}
\gamma^{H\cdot 2^{i-1}}\left(\sum_{h=0}^{2^i\cdot H-1}p_h(s,M^{\pi})\right) &\leq \gamma^{H\cdot 2^{i-1}}\cdot\left(4\cdot|\mathcal{S}|^{4|\mathcal{S}|}\right)^i\cdot\left(\sum_{h=0}^{H-1}p_h(s,M^{\pi})\right) \\
&\leq \left(8\cdot|\mathcal{S}|^{4|\mathcal{S}|}\right)^{-2^{i-1}}\cdot\left(4\cdot|\mathcal{S}|^{4|\mathcal{S}|}\right)^i\cdot\left(\sum_{h=0}^{H-1}p_h(s,M^{\pi})\right) \\
&\leq 1/2^i\cdot\left(\sum_{h=0}^{H-1}p_h(s,M^{\pi})\right).
\end{aligned}$$

Therefore,

$$V^{\pi}_{M,\gamma} \leq \sum_{s\in\mathcal{S}}2\cdot\left(\sum_{h=0}^{H-1}p_h(s,M^{\pi})\right)\cdot\mathbb{E}[R(s,\pi(s))] = 2V^{\pi}_{M,H}.$$

On the other hand,

$$\begin{aligned}
V^{\pi}_{M,\gamma} &= \sum_{s\in\mathcal{S}}\sum_{h=0}^{\infty}\gamma^h\cdot p_h(s,M^{\pi})\cdot\mathbb{E}[R(s,\pi(s))]\\
&\geq \sum_{s\in\mathcal{S}}\sum_{h=0}^{H-1}\gamma^h\cdot p_h(s,M^{\pi})\cdot\mathbb{E}[R(s,\pi(s))]\\
&\geq \gamma^H\cdot\sum_{s\in\mathcal{S}}\sum_{h=0}^{H-1} p_h(s,M^{\pi})\cdot\mathbb{E}[R(s,\pi(s))]\\
&= \gamma^H\cdot V^{\pi}_{M,H} = \left(1-\frac{\ln(8\cdot|\mathcal{S}|^{4|\mathcal{S}|})}{H}\right)^H\cdot V^{\pi}_{M,H}\\
&\geq (1/4)^{\ln(8\cdot|\mathcal{S}|^{4|\mathcal{S}|})}\cdot V^{\pi}_{M,H} \geq \frac{1}{64\cdot|\mathcal{S}|^{8|\mathcal{S}|}}\cdot V^{\pi}_{M,H}.
\end{aligned}$$

□

As another implication of Lemma 3.5.4, for any MDP M and any stationary policy π, we have

$$V^{\pi}_{M,\lfloor H/2\rfloor}\geq\exp\left(-O(|\mathcal{S}|\log|\mathcal{S}|)\right)V^{\pi}_{M,H}.$$

Lemma 3.5.6. *For any MDP M and any stationary policy $\pi:\mathcal{S}\to\mathcal{A}$, if $H\geq 2|\mathcal{S}|$,*

$$V^{\pi}_{M,\lfloor H/2\rfloor}\geq\frac{1}{4\cdot|\mathcal{S}|^{4|\mathcal{S}|}}V^{\pi}_{M,H}.$$

Proof. Note that

$$V^{\pi}_{M,\lfloor H/2\rfloor}=\sum_{s\in\mathcal{S}}\sum_{h=0}^{\lfloor H/2\rfloor-1}p_h(s,M^{\pi})\cdot\mathbb{E}[R(s,\pi(s))].$$

Since $H\geq 2|\mathcal{S}|$, by Lemma 3.5.4, for any $s\in\mathcal{S}$,

$$\sum_{h=0}^{\lfloor H/2\rfloor-1}p_h(s,M^{\pi})\geq\frac{1}{4\cdot|\mathcal{S}|^{4|\mathcal{S}|}}\sum_{h=0}^{2\lfloor H/2\rfloor}p_h(s,M^{\pi})\geq\frac{1}{4\cdot|\mathcal{S}|^{4|\mathcal{S}|}}\sum_{h=0}^{H-1}p_h(s,M^{\pi}).$$

Therefore,

$$V^{\pi}_{M,\lfloor H/2\rfloor}\geq\frac{1}{4\cdot|\mathcal{S}|^{4|\mathcal{S}|}}\sum_{s\in\mathcal{S}}\sum_{h=0}^{H-1}p_h(s,M^{\pi})\cdot\mathbb{E}[R(s,\pi(s))]=\frac{1}{4\cdot|\mathcal{S}|^{4|\mathcal{S}|}}V^{\pi}_{M,H}.$$

□

As a corollary of Lemma 3.5.5, we show that for any episodic MDP M, there always exists a stationary policy whose value is as large as the best non-stationary policy up to a factor of $\exp\left(O(|\mathcal{S}|\log|\mathcal{S}|)\right)$.

Corollary 3.5.7. *For any MDP M, if $H \geq 2\ln(8 \cdot |\mathcal{S}|^{4|\mathcal{S}|})$, then there exists a stationary policy π such that*

$$V^{\pi}_{M,H} \geq \frac{1}{128 \cdot |\mathcal{S}|^{8|\mathcal{S}|}} V^{\pi^*}_{M,H}.$$

Proof. In this proof we fix $\gamma = 1 - \frac{\ln(8 \cdot |\mathcal{S}|^{4|\mathcal{S}|})}{H}$. We also use $\widetilde{\pi}^*$ to denote a non-stationary policy such that $\widetilde{\pi}^*_h = \pi^*_h$ when $h \in [H]$ and $\widetilde{\pi}^*_h$ is defined arbitrarily when $h \geq H$.

Clearly, there exists a stationary policy π such that for any (possibly non-stationary) policy π',

$$V^{\pi}_{M,\gamma} \geq V^{\pi'}_{M,\gamma}.$$

For a proof, see Theorem 5.5.3 in [67]. Clearly,

$$V^{\widetilde{\pi}^*}_{M,\gamma} \geq \gamma^H \cdot V^{\pi^*}_{M,H}.$$

Moreover, by Lemma 3.5.5,

$$V^{\pi}_{M,H} \geq \frac{1}{2} V^{\pi}_{M,\gamma} \geq \frac{1}{2} V^{\widetilde{\pi}^*}_{M,\gamma} \geq \frac{1}{2} \cdot \gamma^H \cdot V^{\pi^*}_{M,H} \geq \frac{1}{128 \cdot |\mathcal{S}|^{8|\mathcal{S}|}} \cdot V^{\pi^*}_{M,H}.$$

□

By applying Corollary 3.5.7 in an MDP with an extra terminal state s_{terminal}, we can show that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, there always exists a stationary policy that visits (s, a) in the first $H/2$ time steps with probability as large as the probability that the best non-stationary policy visits (s, a) in all the H time steps, up to a factor of $\exp\left(O(|\mathcal{S}| \log |\mathcal{S}|)\right)$.

Corollary 3.5.8. *For any MDP M, if $H \geq 2\ln(8 \cdot (|\mathcal{S}| + 1)^{4(|\mathcal{S}|+1)})$, then for any $z \in \mathcal{S} \times \mathcal{A}$, there exists a stationary policy π, such that for any (possibly non-stationary) policy π',*

$$\Pr\left[\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h, a_h) = z] \geq 1\right] \geq \frac{1}{512 \cdot (|\mathcal{S}| + 1)^{12(|\mathcal{S}|+1)}} \Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s'_h, a'_h) = z] \geq 1\right],$$

where

$$(s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}$$

is a random trajectory induced by executing π in M and

$$(s'_1, a'_1), (s'_2, a'_2), \ldots, (s'_H, a'_H), s'_{H+1}$$

is a random trajectory induced by executing π' in M.

Proof. For the given MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$, we create a new MDP

$$M' = \left(\mathcal{S} \cup \{s_{\text{terminal}}\}, \mathcal{A}, P', R', H, \mu\right),$$

where s_{terminal} is a state such that $s_{\text{terminal}} \notin \mathcal{S}$. Moreover,

$$P'(s, a) = \begin{cases} P(s, a) & s \neq s_{\text{terminal}} \text{ and } (s, a) \neq z \\ s_{\text{terminal}} & s = s_{\text{terminal}} \text{ or } (s, a) = z \end{cases}$$

and

$$R'(s,a) = \mathbb{I}[(s,a) = z].$$

Clearly, for any policy π,

$$V^{\pi}_{M',H} = \Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s_h,a_h) = (s,a)] \geq 1\right]$$

where

$$(s_1,a_1),(s_2,a_2),\ldots,(s_H,a_H),s_{H+1}$$

is a random trajectory induced by executing π in M. Therefore, by Corollary 3.5.7, there exists a stationary policy π such that for any (possibly non-stationary) policy π',

$$V^{\pi}_{M',H} \geq \frac{1}{128 \cdot (|\mathcal{S}|+1)^{8(|\mathcal{S}|+1)}} V^{\pi'}_{M',H}.$$

Moreover, by Lemma 3.5.6, for any (possibly non-stationary) policy π',

$$V^{\pi}_{M',\lfloor H/2 \rfloor} \geq \frac{1}{512 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}} V^{\pi'}_{M',H},$$

which implies the desired result. □

Finally, by combining Lemma 3.5.6 and Corollary 3.5.7, we can show that for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, if the initial state distribution μ always takes s and there exists a non-stationary policy that visits (s,a) for f times with probability ε in all the H steps, then there exists a stationary policy that visits (s,a) for $\exp\left(-O(|\mathcal{S}|\log|\mathcal{S}|)\right) \cdot \varepsilon \cdot f$ times with constant probability in the first $H/2$ steps.

Corollary 3.5.9. *For a given MDP M and a state-action pair $z = (s_z, a_z) \in \mathcal{S} \times \mathcal{A}$, suppose the initial state distribution $\mu = s_z$ and $H \geq 2\ln(8 \cdot |\mathcal{S}|^{4|\mathcal{S}|})$. If there exists a (possibly non-stationary) policy π' such that $\mathcal{Q}^{\pi'}_{\varepsilon}(s_z,a_z) \geq f$ for some integer $0 \leq f \leq H$, then there exists a stationary policy π such that*

$$\mathcal{Q}^{\pi}_{1/2}\left(\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h,a_h) = z]\right) \geq \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot f \right\rfloor$$

where

$$(s_1,a_1,r_1),(s_2,a_2,r_2)\ldots,(s_H,a_H,r_H),s_{H+1}$$

is a trajectory induced by executing π in M.

Proof. If $f = 0$ then the lemma is clearly true. No consider the case $f > 0$. Consider a new MDP $M' = (\mathcal{S},\mathcal{A},P,R',H,\mu)$ where $R'(s,a) = \mathbb{I}[(s,a) = z]$. Clearly, $V^{\pi'}_{M',H} \geq \varepsilon \cdot f$. By Corollary 3.5.7, there exists a stationary policy π such that

$$V^{\pi}_{M',H} \geq \frac{1}{128 \cdot |\mathcal{S}|^{8|\mathcal{S}|}} \cdot \varepsilon \cdot f.$$

By Lemma 3.5.6,

$$V^{\pi}_{M',\lfloor H/2 \rfloor} \geq \frac{1}{512 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot f.$$

This implies $\pi(s_z) = a_z$.

Now we use X to denote a random variable which is defined to be

$$X = \min\{h \geq 1 \mid (s_{h+1}, a_{h+1}) = z\}.$$

Here the trajectory

$$(s_1, a_1), (s_2, a_2), \ldots$$

is induced by executing the stationary policy π in M'. We also write $\widehat{X} = \min\{\lfloor H/2 \rfloor, X\}$. We use $\{X_i\}_{i=1}^{\infty}$ to denote a sequence of i.i.d. copies of $\widehat{X}$. We use τ to denote a random variable which is defined to be

$$\tau = \min\left\{ i \geq 1 \mid \sum_{j=1}^{i} X_j \geq \lfloor H/2 \rfloor \right\}.$$

Clearly, $\tau \leq H/2$ almost surely. Moreover, π is a stationary policy, the initial state distribution $\mu = s_z$ deterministically and $\pi(s_z) = a_z$, which implies τ and $\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h, a_h) = z]$ have the same distribution. Indeed, whenever the trajectory $(s_1, a_1), (s_2, a_2) \ldots$ visits z, it corresponds to a new copy of $\widehat{X}$.

Now for each $i > 0$, we define $Y_i = X_i - \mathbb{E}[\widehat{X}]$. Clearly $\mathbb{E}[Y_i] = 0$. Let $S_i = 0$ and $S_i = \sum_{j=1}^{i} Y_j$ for all $i > 0$. Clearly τ is a stopping time, and

$$\sum_{j=1}^{\tau} X_j \leq H$$

since $X_i \leq \lfloor H/2 \rfloor$ for all $i > 0$. By Martingale Stopping Theorem, we have

$$\mathbb{E}[S_\tau] = \sum_{j=1}^{\tau} \mathbb{E}[X_j] - \mathbb{E}[\tau] \cdot \mathbb{E}[\widehat{X}] = 0,$$

which implies $\mathbb{E}[\tau] \cdot \mathbb{E}[\widehat{X}] \leq H$ and therefore

$$\mathbb{E}[\widehat{X}] \leq H/\mathbb{E}[\tau] = H/V^{\pi}_{M',\lfloor H/2 \rfloor} \leq 512 \cdot |\mathcal{S}|^{12|\mathcal{S}|} H/(\varepsilon \cdot f),$$

where we use the fact that

$$V^{\pi}_{M',\lfloor H/2 \rfloor} = \mathbb{E}\left[\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h, a_h) = z] \right] = \mathbb{E}[\tau].$$

Let $\tau' = \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot f \right\rfloor$. By Markov's inequality, with probability at least $1/2$,

$$\sum_{i=1}^{\tau'} X_i \leq 2\tau' \mathbb{E}[\widehat{X}] \leq H/2,$$

in which case $\tau \geq \tau'$. Consequently,

$$\mathcal{Q}^{\pi}_{1/2}\left(\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h, a_h) = z]\right) = \mathcal{Q}_{1/2}(\tau) \geq \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot f \right\rfloor.$$

□

3.5.2 The Algorithm

In this section, we present our algorithm together with its analysis. Our algorithm is divided into two parts. In Section 3.5.2.1, we first present the algorithm for collecting samples together with its analysis. In Section 3.5.2.2, we establish a perturbation analysis on the value functions which is crucial for the analysis in later proofs. Finally, in Section 3.5.2.3, we present the algorithm for finding near-optimal policies based on the dataset found by the algorithm in Section 3.5.2.1, together with its analysis based on the machinery developed in Section 3.5.2.2.

3.5.2.1 Collecting Samples

In this section, we present our algorithm for collecting samples. The algorithm is formally presented in Algorithm 4. The dataset D returned by Algorithm 4 consists of N lists, where for each list, elements in the list are tuples of the form $(s, a, r, s') \in \mathcal{S} \times \mathcal{A} \times [0, 1] \times \mathcal{S}$. To construct these lists, Algorithm 4 enumerates a state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ and a pair of stationary policies (π_1, π_2), and then collects a trajectory using π_1 and π_2. More specifically, π_1 is executed until the trajectory visits (s, a), at which point π_2 is executed until the last step.

Algorithm 4 Collect Samples

Input: number of repetitions N
Output: Dataset D where $D = \left(\left(\left(s_{i,t}, a_{i,t}, r_{i,t}, s'_{i,t}\right)\right)_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\right)_{i=1}^{N}$
for $i \in [N]$ **do**
    Let T_i be an empty list
    for $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
        for $(\pi_1, \pi_2) \in \Pi_{\mathsf{st}} \times \Pi_{\mathsf{st}}$ **do**
            Receive $s_1 \sim \mu$
            for $h \in [H]$ **do**
                if $(s, a) = (s_{h'}, a_{h'})$ for some $h' < h$ **then**
                    Take $a_h = \pi_2(s_h)$
                else
                    Take $a_h = \pi_1(s_h)$
                Receive $r_h \sim R(s_h, a_h)$ and $s_{h+1} \sim P(s_h, a_h)$
                Append (s_h, a_h, r_h, s_{h+1}) to the end of T_i
return D where $D = (T_i)_{i=1}^{N}$

Throughout this section, we use $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ to denote the underlying MDP that the agent interacts with. For each $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $(\pi_1, \pi_2) \in \Pi_{\mathsf{st}} \times \Pi_{\mathsf{st}}$, let

$$(s_1^{s,a,\pi_1,\pi_2}, a_1^{s,a,\pi_1,\pi_2}, r_1^{s,a,\pi_1,\pi_2}), (s_2^{s,a,\pi_1,\pi_2}, a_2^{s,a,\pi_1,\pi_2}, r_2^{s,a,\pi_1,\pi_2}), \ldots, (s_H^{s,a,\pi_1,\pi_2}, a_H^{s,a,\pi_1,\pi_2}, r_H^{s,a,\pi_1,\pi_2}), s_{H+1}^{s,a,\pi_1,\pi_2}$$

by a trajectory where $s_1^{s,a,\pi_1,\pi_2} \sim \mu$ and $s_{h+1}^{s,a,\pi_1,\pi_2} \sim P(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2})$ for all $1 \le h \le H$, $r_h^{s,a,\pi_1,\pi_2} \sim R(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2})$ for all $h \in [H]$, and

$$a_h^{s,a,\pi_1,\pi_2} = \begin{cases} \pi_2(s_h^{s,a,\pi_1,\pi_2}) & (s,a) = (s_{h'}^{s,a,\pi_1,\pi_2}, a_{h'}^{s,a,\pi_1,\pi_2}) \text{ for some } h' < h \\ \pi_1(s_h^{s,a,\pi_1,\pi_2}) & \text{otherwise} \end{cases}$$

for all $h \in [H]$. Note that the above trajectory is the one collected by Algorithm 4 when a specific state-action pair (s, a) and a specific pair of policies (π_1, π_2) are used.

For any $\varepsilon \in (0, 1]$, define

$$\mathcal{Q}_\varepsilon^{\mathsf{st}}(s,a) = \mathcal{Q}_\varepsilon \left(\sum_{(s',a') \in \mathcal{S} \times \mathcal{A}} \sum_{\pi_1 \in \Pi_{\mathsf{st}}} \sum_{\pi_2 \in \Pi_{\mathsf{st}}} \sum_{h=1}^{H} \mathbb{I}[(s_h^{s',a',\pi_1,\pi_2}, a_h^{s',a',\pi_1,\pi_2}) = (s,a)] \right).$$

Clearly, $\mathcal{Q}_\varepsilon^{\mathsf{st}}(s, a)$ is the ε-quantile of the frequency that (s, a) appears in each T_i.

In Lemma 3.5.10, we first show that for each $(s, a) \in \mathcal{S} \times \mathcal{A}$, if there exists a policy π that visits (s, a) for $m(s, a)$ times with probability at least ε, then

$$\mathcal{Q}_{\varepsilon/\exp(O(|\mathcal{S}| \log |\mathcal{S}|))}^{\mathsf{st}}(s,a) \ge m(s,a)/\exp(O(|\mathcal{S}| \log |\mathcal{S}|)).$$

Lemma 3.5.10. *Let $\varepsilon \in (0, 1]$ be a given real number. For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, let $m_\varepsilon(s, a)$ be the largest integer such that there exists a (possibly non-stationary) policy $\pi_{s,a}$ such that $\mathcal{Q}_\varepsilon^{\pi_{s,a}}(s, a) \ge m_\varepsilon(s, a)$. For each $(s, a) \in \mathcal{S} \times \mathcal{A}$,*

$$\mathcal{Q}_{\varepsilon(|\mathcal{S}|+1)^{-12(|\mathcal{S}|+1)}/1024}^{\mathsf{st}}(s,a) \ge \frac{1}{4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a).$$

Proof. For each $(s, a) \in \mathcal{S} \times \mathcal{A}$, there exists a (possibly non-stationary) policy $\pi_{s,a}$ such that $\mathcal{Q}_\varepsilon^{\pi_{s,a}}(s, a) \ge m_\varepsilon(s, a)$. Here we consider the case that $m_\varepsilon(s, a) \ge 1$, since otherwise the lemma clearly holds. By Corollary 3.5.8, there exists a stationary policy $\pi'_{s,a}$ such that

$$\Pr\left[\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h, a_h) = (s, a)] \ge 1 \right] \ge \frac{\varepsilon}{512 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}},$$

where

$$(s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}$$

is a random trajectory induced by executing $\pi'_{s,a}$ in M.

In the remaining part of the analysis, we consider two cases.

Case I: $m_\varepsilon(s,a) \geq 4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}/\varepsilon$**.** Let

$$(s_1,a_1),(s_2,a_2),\ldots,(s_H,a_H),s_{H+1}$$

be a random trajectory induced by executing $\pi_{s,a}$ in M. Let $X_{s,a}$ be the random variable which is defined to be

$$X_{s,a} = \begin{cases} \min\{h \in [H] \mid (s_h,a_h) = (s,a)\} & \text{if there exists } h \in [H] \text{ such that } (s_h,a_h)=(s,a) \\ H+1 & \text{otherwise} \end{cases}.$$

Clearly,

$$\begin{aligned} &\sum_{h'=1}^{H} \Pr[X_{s,a}=h'] \cdot \Pr\left[\sum_{h=h'}^{H} \mathbb{I}[(s_h,a_h)=(s,a)] \geq m_\varepsilon(s,a) \mid (s_{h'},a_{h'})=(s,a)\right] \\ =&\Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s_h,a_h)=(s,a)] \geq m_\varepsilon(s,a)\right] \geq \varepsilon. \end{aligned}$$

Therefore, there exists $h' \in [H]$ such that

$$\Pr[X_{s,a}=h'] > 0$$

and

$$\Pr\left[\sum_{h=h'}^{H} \mathbb{I}[(s_h,a_h)=(s,a)] \geq m_\varepsilon(s,a) \mid (s_{h'},a_{h'})=(s,a)\right] \geq \varepsilon.$$

Note that we must have $\pi_{h'}(s)=a$, since otherwise $\Pr[X_{s,a}=h']=0$.

Now we consider a new MDP $M_s=(\mathcal{S},\mathcal{A},P,R,H,\mu_s)$ where $\mu_s=s$. Let $\widetilde{\pi}$ be an arbitrary policy so that $\widetilde{\pi}_h=(\pi_{s,a})_{h'+h}$ for all $h \in [H-h']$. Clearly,

$$\Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s'_h,a'_h)=(s,a)] \geq m_\varepsilon(s,a)\right] \geq \Pr\left[\sum_{h=1}^{H-h'} \mathbb{I}[(s'_h,a'_h)=(s,a)] \geq m_\varepsilon(s,a)\right] \geq \varepsilon$$

where

$$(s'_1,a'_1),(s'_2,a'_2),\ldots,(s'_H,a'_H),s'_{H+1}$$

is a random trajectory induced by executing $\widetilde{\pi}$ in M_s. Therefore, by Corollary 3.5.9, there exists a stationary policy $\widetilde{\pi}_{s,a}$ such that

$$\Pr\left[\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s''_h,a''_h)=(s,a)] \geq \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a) \right\rfloor\right] \geq 1/2$$

where

$$(s''_1,a''_1),(s''_2,a''_2),\ldots,(s''_H,a''_H),s''_{H+1}$$

is a random trajectory induced by executing $\widetilde{\pi}_{s,a}$ in M_s. Since $m_\varepsilon(s,a) \geq 4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}/\varepsilon$ and thus $\left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a) \right\rfloor \geq 1$, we must have $\widetilde{\pi}_{s,a}(s) = a$.

Now we consider the case when $\pi_1 = \pi'_{s,a}$ and $\pi_2 = \widetilde{\pi}_{s,a}$. Since $\pi_1 = \pi'_{s,a}$,

$$\Pr\left[\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq 1\right] \geq \frac{\varepsilon}{512 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}}.$$

Therefore, let $X'_{s,a}$ be a random variable which is defined to be

$$X'_{s,a} = \begin{cases} \min\{h \in [\lfloor H/2 \rfloor] \mid (s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)\} & \text{if } (s_h,a_h) = (s,a) \text{ for some } h \in [\lfloor H/2 \rfloor] \\ \lfloor H/2 \rfloor + 1 & \text{otherwise} \end{cases}.$$

We have that

$$\Pr[X'_{s,a} \in [\lfloor H/2 \rfloor]] \geq \frac{\varepsilon}{512 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}}.$$

Moreover, for each $h' \in [\lfloor H/2 \rfloor]$, since $\pi_2 = \widetilde{\pi}_{s,a}$,

$$\Pr\left[\sum_{h=h'}^{H} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a) \right\rfloor \mid (s_{h'}^{s,a,\pi_1,\pi_2}, a_{h'}^{s,a,\pi_1,\pi_2}) = (s,a)\right]$$
$$\geq 1/2.$$

Therefore,

$$\Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a) \right\rfloor\right]$$
$$\geq \sum_{h'=1}^{\lfloor H/2 \rfloor} \Pr[X'_{s,a} = h']$$
$$\cdot \Pr\left[\sum_{h=h'}^{H} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq \left\lfloor \frac{1}{2048 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a) \right\rfloor \mid (s_{h'}^{s,a,\pi_1,\pi_2}, a_{h'}^{s,a,\pi_1,\pi_2}) = (s,a)\right]$$
$$\geq \frac{\varepsilon}{1024 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}}.$$

Since $m_\varepsilon(s,a) \geq 4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}/\varepsilon$, we have

$$\Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2} a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq \frac{1}{4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a)\right] \geq \frac{\varepsilon}{1024 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}}$$

and thus

$$\mathcal{Q}^{\mathsf{st}}_{\varepsilon(|\mathcal{S}|+1)^{-12(|\mathcal{S}|+1)}/1024}(s,a) \geq \frac{1}{4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a).$$

Case II: $m_\varepsilon(s,a) < 4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}/\varepsilon$**.** Consider the case when $\pi_1 = \pi_2 = \pi'_{s,a}$. Clearly,

$$\begin{aligned}&\Pr\left[\sum_{h=1}^{H} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq 1\right] \\ \geq &\Pr\left[\sum_{h=1}^{\lfloor H/2 \rfloor} \mathbb{I}[(s_h^{s,a,\pi_1,\pi_2}, a_h^{s,a,\pi_1,\pi_2}) = (s,a)] \geq 1\right] \geq \frac{\varepsilon}{512 \cdot (|\mathcal{S}|+1)^{12(|\mathcal{S}|+1)}}\end{aligned}$$

and thus

$$\mathcal{Q}^{\mathsf{st}}_{\varepsilon(|\mathcal{S}|+1)^{-12(|\mathcal{S}|+1)}/1024}(s,a) \geq \frac{1}{4096 \cdot |\mathcal{S}|^{12|\mathcal{S}|}} \cdot \varepsilon \cdot m_\varepsilon(s,a).$$

□

Now we show that for a given percentile ε, for the dataset D returned by Algorithm 4, for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, (s,a) appears for at least $\mathcal{Q}^{\mathsf{st}}_{\varepsilon/4}(s,a)$ times for at least $\Omega(N \cdot \varepsilon)$ lists out of the N lists returned by Algorithm 4.

Lemma 3.5.11. *Let $\varepsilon, \delta \in (0,1]$ be a given real number. Let D be the dataset returned by Algorithm 4 where*

$$D = \left(\left(\left(s_{i,t}, a_{i,t}, r_{i,t}, s'_{i,t}\right)\right)_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\right)_{i=1}^{N}.$$

Suppose $N \geq 16/\varepsilon \cdot \log(3|\mathcal{S}||\mathcal{A}|/\delta)$. With probability at least $1-\delta/3$, for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\sum_{i=1}^{N} \mathbb{I}\left[\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} \mathbb{I}[(s_{i,t}, a_{i,t}) = (s,a)] \geq \mathcal{Q}^{\mathsf{st}}_{\varepsilon/4}(s,a)\right] \geq N\varepsilon/8.$$

Proof. For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, by the definition of $\mathcal{Q}^{\mathsf{st}}_{\varepsilon/4}(s,a)$, for each $i \in [N]$, we have

$$\mathbb{E}\left[\mathbb{I}\left[\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} \mathbb{I}[(s_{i,t}, a_{i,t}) = (s,a)] \geq \mathcal{Q}^{\mathsf{st}}_{\varepsilon/4}(s,a)\right]\right] \geq \varepsilon/4.$$

Hence, the desired result follows by Chernoff bound and a union bound over all $(s,a) \in \mathcal{S} \times \mathcal{A}$. □

We also need a subroutine to estimate $\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)$ for some ε_{est} to be decided. Such estimates are crucial for building estimators for the transition probabilities and the rewards with bounded variance, which we elaborate in later parts of this section.

Our algorithm for estimating $\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)$ is described in Algorithm 5. Algorithm 5 collects N lists, where for each list, elements in the list are tuples of the form $(s,a,r,s') \in \mathcal{S} \times \mathcal{A} \times [0,1] \times \mathcal{S}$. These N lists are collected using the same approach as in Algorithm 4. Once these N lists are collected, for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, our estimate (denoted as $\overline{m}^{\mathsf{st}}(s,a)$) is then set to be the $\lceil N \cdot \varepsilon_{\text{est}}/2 \rceil$-th largest element in $F_{s,a}$, where $F_{s,a}$ is the set of the number of times (s,a) appear in each of the N lists.

We now show that for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, $\overline{m}^{\mathsf{st}}(s,a)$ is an accurate estimate of $\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)$.

Algorithm 5 Estimate Quantiles

Input: Percentile ε_{est}, failure probability δ_{est}
Output: Estimates $\overline{m}^{\mathsf{st}} : \mathcal{S} \times \mathcal{A} \to \mathbb{N}$
Let $N = \lceil 300 \log(6|\mathcal{S}||\mathcal{A}|/\delta_{\text{est}})/\varepsilon_{\text{est}} \rceil$
Let $F_{s,a}$ be an empty multiset for all $(s,a) \in \mathcal{S} \times \mathcal{A}$
for $i \in [N]$ **do**
    Let T_i be an empty list
    for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
        for $(\pi_1, \pi_2) \in \Pi_{\mathsf{st}} \times \Pi_{\mathsf{st}}$ **do**
            Receive $s_1 \sim \mu$
            for $h \in [H]$ **do**
                if $(s,a) = (s_{h'}, a_{h'})$ for some $0 \le h' < h$ **then**
                    Take $a_h = \pi_2(s_h)$
                else
                    Take $a_h = \pi_1(s_h)$
                Receive $r_h \sim R(s_h, a_h)$ and $s_{h+1} \sim P(s_h, a_h)$
                Append (s_h, a_h, r_h, s_{h+1}) to the end of T_i
    for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
        Add $\sum_{t=1}^{|T_i|} \mathbb{I}[(s_t, a_t) = (s,a)]$ into $F_{s,a}$ where

$$T_i = ((s_1, a_1, r_1, s_1'), (s_2, a_2, r_2, s_2'), \ldots, (s_{|T_i|}, a_{|T_i|}, r_{|T_i|}, s_{|T_i|}'))$$

for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
    Set $\overline{m}^{\mathsf{st}}(s,a)$ be the $\lceil N \cdot \varepsilon_{\text{est}}/2 \rceil$-th largest element in $F_{s,a}$
return $\overline{m}^{\mathsf{st}}$

Lemma 3.5.12. *Let $\overline{m}^{\mathsf{st}}$ be the function returned by Algorithm 5. With probability at least $1 - \delta_{\text{est}}/3$, for all $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a) \le \overline{m}^{\mathsf{st}}(s,a) \le \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}/4}(s,a).$$

Proof. Fix a state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. For each $i \in [N]$, define

$$\underline{X}_i = \mathbb{I}\left[\sum_{t=1}^{|T_i|} \mathbb{I}[(s_t, a_t) = (s,a)] > \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}/4}(s,a)\right]$$

where

$$T_i = ((s_1, a_1, r_1, s_1'), (s_2, a_2, r_2, s_2'), \ldots, (s_{|T_i|}, a_{|T_i|}, r_{|T_i|}, s_{|T_i|}')).$$

For each $i \in [N]$, by the definition of $\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}/4}(s,a)$, we have $\mathbb{E}[\underline{X}_i] \le \varepsilon_{\text{est}}/4$ and thus $\sum_{i=1}^{N} \mathbb{E}[\underline{X}_i] \le N \cdot \varepsilon_{\text{est}}/4$. By Chernoff bound, with probability at most $\delta_{\text{est}}/(6|\mathcal{S}||\mathcal{A}|)$,

$$\sum_{i=1}^{N} \underline{X}_i \ge N \cdot \varepsilon_{\text{est}}/3.$$

On the other hand, for each $i \in [N]$, define

$$\overline{X}_i = \mathbb{I}\left[\sum_{t=1}^{|T_i|} \mathbb{I}[(s_{i,t}, a_{i,t}) = (s,a)] \geq \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)\right].$$

where

$$T_i = ((s_1, a_1, r_1, s_1'), (s_2, a_2, r_2, s_2'), \ldots, (s_{|T_i|}, a_{|T_i|}, r_{|T_i|}, s_{|T_i|}')).$$

For each $i \in [N]$, by the definition of $\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)$, we have $\mathbb{E}[\overline{X}_i] \geq \varepsilon_{\text{est}}$ and thus $\sum_{i=1}^N \mathbb{E}[\overline{X}_i] \geq N \cdot \varepsilon_{\text{est}}$. By Chernoff bound, with probability at most $\delta_{\text{est}}/(6|\mathcal{S}||\mathcal{A}|)$,

$$\sum_{i=1}^N \overline{X}_i \leq 2N \cdot \varepsilon_{\text{est}}/3.$$

Hence, by union bound, with probability at least $1 - \delta_{\text{est}}/(3|\mathcal{S}||\mathcal{A}|)$,

$$\sum_{i=1}^N \underline{X}_i < N \cdot \varepsilon_{\text{est}}/3.$$

and

$$\sum_{i=1}^N \overline{X}_i > 2N \cdot \varepsilon_{\text{est}}/3,$$

in which case the $\lceil N \cdot \varepsilon_{\text{est}}/2 \rceil$-th largest element in $F_{s,a}$ is in $\left[\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a), \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}/4}(s,a)\right]$. We finish the proof by a union bound over all $(s,a) \in \mathcal{S} \times \mathcal{A}$. □

In Lemma 4.4.5, we show that using the dataset D returned by Algorithm 4, and the estimates of quantiles returned by Algorithm 5, we can compute accurate estimates of the transition probabilities and rewards. The estimators used in Lemma 4.4.5 are the empirical estimators, with proper truncation if a list T_i contains too many samples (i.e., more than $\overline{m}^{\mathsf{st}}(\cdot,\cdot)$). As will be made clear in the proof, such truncation is crucial for obtaining estimators with bounded variance.

Lemma 3.5.13. *Suppose Algorithm 5 is invoked with the percentile set to be ε_{est} and the failure probability set to be δ, and Algorithm 4 is invoked with $N \geq 16/\varepsilon_{\text{est}} \cdot \log(3|\mathcal{S}||\mathcal{A}|/\delta)$. Let $\overline{m}^{\mathsf{st}} : \mathcal{S} \times \mathcal{A} \to \mathbb{N}$ be the estimates returned by Algorithm 5. Let D be the dataset returned by Algorithm 4 where*

$$D = \left(\left((s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t}')\right)_{t=1}^{|\mathcal{S}||\mathcal{A}| \cdot |\mathcal{A}|^{2|\mathcal{S}|} \cdot H}\right)_{i=1}^{N}.$$

For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, for each $i \in [N]$ and $t \in \left[|\mathcal{S}||\mathcal{A}| \cdot |\mathcal{A}|^{2|\mathcal{S}|} \cdot H\right]$, define

$$\mathsf{Trunc}_{i,t}(s,a) = \mathbb{I}\left[\sum_{t'=1}^{t} \mathbb{I}\left[(s_{i,t'}, a_{i,t'}) = (s,a)\right] < \overline{m}^{\mathsf{st}}(s,a)\right].$$

For each $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$, define

$$m_D(s,a) = \sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} \mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot \mathsf{Trunc}_{i,t}(s,a),$$

$$\widehat{P}(s' \mid s,a) = \frac{\sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\mathbb{I}\left[(s_{i,t},a_{i,t},s'_{i,t})=(s,a,s')\right]\cdot \mathsf{Trunc}_{i,t}(s,a)}{\max\{1,m_D(s,a)\}},$$

$$\widehat{R}(s,a) = \frac{\sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot r_{i,t}\cdot \mathsf{Trunc}_{i,t}(s,a)}{\max\{1,m_D(s,a)\}}$$

and

$$\widehat{\mu}(s) = \frac{\sum_{i=1}^{N}\mathbb{I}[s_{i,1}=s]}{N}.$$

Then with probability at least $1-\delta$, for all $(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}$ with $\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)>0$, we have

$$\begin{aligned}\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| &\le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\varepsilon_{\text{est}}}, 32\sqrt{\frac{\widehat{P}(s' \mid s,a)\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\varepsilon_{\text{est}}}}\right\}\\ &\le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)\cdot N\cdot\varepsilon_{\text{est}}}, 64\sqrt{\frac{P(s' \mid s,a)\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)\cdot N\cdot\varepsilon_{\text{est}}}}\right\},\end{aligned}$$

$$\left|\widehat{R}(s' \mid s,a) - \mathbb{E}[R(s,a)]\right| \le 8\sqrt{\frac{\mathbb{E}\left[(R(s,a))^2\right]\cdot\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)\cdot N\cdot\varepsilon_{\text{est}}}} + \frac{8\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a)\cdot N\cdot\varepsilon_{\text{est}}},$$

and

$$|\widehat{\mu}(s)-\mu(s)| \le \sqrt{\frac{\log(18|\mathcal{S}|/\delta)}{N}}.$$

Proof. Fix a state-action pair $(s,a)\in\mathcal{S}\times\mathcal{A}$ and $s'\in\mathcal{S}$. For each $i\in[N]$ and $t\in\left[|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H\right]$, let $\mathcal{F}_{i,t}$ be the filtration induced by

$$\left\{\left(s_{i,t'},a_{i,t'},r_{i,t'},s'_{i,t'}\right)\right\}_{t'=1}^{t-1}.$$

For each $i\in[N]$ and $t\in\left[|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H\right]$, define

$$X_{i,t} = \left(\mathbb{I}\left[(s_{i,t},a_{i,t},s'_{i,t})=(s,a,s')\right] - P(s' \mid s,a)\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\right)\cdot\mathsf{Trunc}_{i,t}(s,a)$$

and

$$Y_{i,t} = \mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot(r_{i,t}-\mathbb{E}[R(s,a)])\cdot\mathsf{Trunc}_{i,t}(s,a).$$

Clearly,

$$\begin{aligned}&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t},s'_{i,t})=(s,a,s')\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\mid\mathcal{F}_{i,t}\right]\\ =&P(s' \mid s,a)\cdot\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\mid\mathcal{F}_{i,t}\right]\end{aligned}$$

and

$$\begin{aligned}&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot r_{i,t}\cdot\mathsf{Trunc}_{i,t}(s,a)\mid\mathcal{F}_{i,t}\right]\\=&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathbb{E}[R(s,a)]\cdot\mathsf{Trunc}_{i,t}(s,a)\mid\mathcal{F}_{i,t}\right],\end{aligned}$$

which implies

$$\begin{aligned}&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t},s'_{i,t})=(s,a,s')\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\right]\\=&P(s'\mid s,a)\cdot\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\right],\end{aligned}$$

and

$$\begin{aligned}&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot r_{i,t}\cdot\mathsf{Trunc}_{i,t}(s,a)\right]\\=&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathbb{E}[R(s,a)]\cdot\mathsf{Trunc}_{i,t}(s,a)\right],\end{aligned}$$

and thus

$$\mathbb{E}\left[X_{i,t}\right]=\mathbb{E}\left[Y_{i,t}\right]=0.$$

Moreover, for any $i\in[N]$ and $1\le t'<t\le|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H$, we have

$$\mathbb{E}\left[X_{i,t'}\cdot X_{i,t}\right]=\mathbb{E}\left[\mathbb{E}\left[X_{i,t'}\cdot X_{i,t}\mid\mathcal{F}_{i,t}\right]\right]=\mathbb{E}\left[X_{i,t'}\cdot\mathbb{E}\left[X_{i,t}\mid\mathcal{F}_{i,t}\right]\right]=0$$

and

$$\mathbb{E}\left[Y_{i,t'}\cdot Y_{i,t}\right]=\mathbb{E}\left[\mathbb{E}\left[Y_{i,t'}\cdot Y_{i,t}\mid\mathcal{F}_{i,t}\right]\right]=\mathbb{E}\left[Y_{i,t'}\cdot\mathbb{E}\left[Y_{i,t}\mid\mathcal{F}_{i,t}\right]\right]=0.$$

Note that for each $i\in[N]$,

$$\mathbb{E}\left[\left(\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}X_{i,t}\right)^2\right]=\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\mathbb{E}\left[\left(X_{i,t}\right)^2\right]$$

and

$$\mathbb{E}\left[\left(\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}Y_{i,t}\right)^2\right]=\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\mathbb{E}\left[\left(Y_{i,t}\right)^2\right].$$

Furthermore, for each $i\in[N]$ and $t\in\left[|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H\right]$,

$$\begin{aligned}\mathbb{E}\left[X_{i,t}^2\right]\le&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t},s'_{i,t})=(s,a,s')\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\right]\\&+\mathbb{E}\left[\left(P(s'\mid s,a)\right)^2\cdot\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\right]\\\le&2P(s'\mid s,a)\cdot\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\right]\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\left[Y_{i,t}^2\right]\le&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\left(r_{i,t}-\mathbb{E}[R(s,a)]\right)^2\cdot\mathsf{Trunc}_{i,t}(s,a)\right]\\\le&\mathbb{E}\left[\mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot\mathbb{E}\left[\left(R(s,a)\right)^2\right]\cdot\mathsf{Trunc}_{i,t}(s,a)\right].\end{aligned}$$

Since

$$\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} \mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot \mathsf{Trunc}_{i,t}(s,a) \le \overline{m}^{\mathsf{st}}(s,a),$$

we have

$$\mathbb{E}\left[\left(\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} X_{i,t}\right)^2\right] \le 2P(s' \mid s,a)\cdot \overline{m}^{\mathsf{st}}(s,a)$$

and

$$\mathbb{E}\left[\left(\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} Y_{i,t}\right)^2\right] \le \mathbb{E}\left[(R(s,a))^2\right]\cdot \overline{m}^{\mathsf{st}}(s,a).$$

Now, for each $i \in [N]$, define

$$\mathcal{X}_i = \sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} X_{i,t}$$

and

$$\mathcal{Y}_i = \sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} Y_{i,t}.$$

We have $\mathbb{E}[\mathcal{X}_i] = \mathbb{E}[\mathcal{Y}_i] = 0$,

$$\mathbb{E}[\mathcal{X}_i^2] \le 2P(s' \mid s,a)\cdot \overline{m}^{\mathsf{st}}(s,a)$$

and

$$\mathbb{E}[\mathcal{Y}_i^2] \le \mathbb{E}\left[(R(s,a))^2\right]\cdot \overline{m}^{\mathsf{st}}(s,a).$$

Also note that

$$\sum_{i=1}^{N}\mathcal{X}_i = \sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} \mathbb{I}\left[(s_{i,t},a_{i,t},s'_{i,t})=(s,a,s')\right]\cdot \mathsf{Trunc}_{i,t}(s,a) - P(s' \mid s,a)\cdot m_D(s,a)$$

and

$$\sum_{i=1}^{N}\mathcal{Y}_i = \sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H} \mathbb{I}\left[(s_{i,t},a_{i,t})=(s,a)\right]\cdot r_{i,t}\cdot \mathsf{Trunc}_{i,t}(s,a) - \mathbb{E}[R(s,a)]\cdot m_D(s,a).$$

By Bernstein's inequality,

$$\Pr\left[\left|\sum_{i=1}^{N}\mathcal{X}_i\right| \ge t\right] \le 2\exp\left(\frac{-t^2}{2\cdot \overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot P(s' \mid s,a) + t/3}\right).$$

Thus, by setting $t = 2\sqrt{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot P(s' \mid s,a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)} + \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)$, we have

$$\Pr\left[\left|\sum_{i=1}^{N} \mathcal{X}_i\right| \geq t\right] \leq \delta/(9|\mathcal{S}|^2|\mathcal{A}|).$$

By applying a union bound over all $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, with probability at least $1 - \delta/9$, for all $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$,

$$\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| \leq \frac{2\sqrt{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot P(s' \mid s,a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}}{m_D(s,a)} + \frac{\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{m_D(s,a)},$$

which we define to be event $\mathcal{E}_P$. Note that conditioned on $\mathcal{E}_P$ and the events in Lemma 3.5.12 and Lemma 3.5.11, we have

$$\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a) \leq \overline{m}^{\mathsf{st}}(s,a) \leq \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}/4}(s,a),$$

which implies

$$m_D(s,a) \geq N \cdot \varepsilon_{\text{est}}/8 \cdot \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}/4}(s,a) \geq N \cdot \varepsilon_{\text{est}}/8 \cdot \overline{m}^{\mathsf{st}}(s,a) \geq N \cdot \varepsilon_{\text{est}}/8 \cdot \mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a),$$

and thus

$$\begin{aligned}
\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| \leq & 8\sqrt{\frac{P(s' \mid s,a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}}} + \frac{8\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}} \\
\leq & 8\sqrt{\frac{P(s' \mid s,a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}}} + \frac{64\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}} \\
\leq & \max\left\{16\sqrt{\frac{P(s' \mid s,a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}}}, \frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}}\right\}
\end{aligned}$$

for all $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$.

When

$$P(s' \mid s,a) \leq \frac{1024\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}},$$

we have

$$16\sqrt{\frac{P(s' \mid s,a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}}} \leq \frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}},$$

and therefore

$$\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| \leq \frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}} \leq \frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\text{est}}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}}.$$

When

$$P(s' \mid s,a) \geq \frac{1024\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a) \cdot N \cdot \varepsilon_{\text{est}}},$$

we have

$$\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| \le 16\sqrt{\frac{P(s' \mid s,a)\cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot \varepsilon_{\mathrm{est}}}} \le P(s' \mid s,a)/2$$

and thus

$$P(s' \mid s,a)/2 \le \widehat{P}(s' \mid s,a) \le 2P(s' \mid s,a),$$

which implies

$$\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| \le 32\sqrt{\frac{\widehat{P}(s' \mid s,a)\cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot \varepsilon_{\mathrm{est}}}} \le 64\sqrt{\frac{P(s' \mid s,a)\cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot \varepsilon_{\mathrm{est}}}}.$$

Hence, conditioned on $\mathcal{E}_P$ and the events in Lemma 3.5.12 and Lemma 3.5.11, for all $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$,

$$\begin{aligned}\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| &\le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}, 32\sqrt{\frac{\widehat{P}(s' \mid s,a)\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}}\right\}\\ &\le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}, 64\sqrt{\frac{P(s' \mid s,a)\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}}\right\}.\end{aligned}$$

By Bernstein's inequality,

$$\Pr\left[\left|\sum_{i=1}^{N}\mathcal{Y}_i\right| \ge t\right] \le 2\exp\left(\frac{-t^2}{\mathbb{E}\left[(R(s,a))^2\right]\cdot\overline{m}^{\mathsf{st}}(s,a)\cdot N + t/3}\right).$$

Thus, by setting $t = 2\sqrt{\mathbb{E}\left[(R(s,a))^2\right]\cdot\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\log(18|\mathcal{S}||\mathcal{A}|/\delta)} + \log(18|\mathcal{S}||\mathcal{A}|/\delta)$, we have

$$\Pr\left[\left|\sum_{i=1}^{N}\mathcal{Y}_i\right| \ge t\right] \le \delta/(9|\mathcal{S}||\mathcal{A}|).$$

By applying a union bound over all $(s,a,s') \in \mathcal{S}\times\mathcal{A}$, with probability at least $1-\delta/9$, for all $(s,a)\in\mathcal{S}\times\mathcal{A}$,

$$\left|\widehat{R}(s' \mid s,a) - \mathbb{E}[R(s,a)]\right| \le \frac{2\sqrt{\mathbb{E}\left[(R(s,a))^2\right]\cdot\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\log(18|\mathcal{S}||\mathcal{A}|/\delta)}}{m_D(s,a)} + \frac{\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{m_D(s,a)},$$

which we define to be event $\mathcal{E}_R$. Note that conditioned on $\mathcal{E}_R$ and the events in Lemma 3.5.12 and Lemma 3.5.11, we have

$$\left|\widehat{R}(s' \mid s,a) - \mathbb{E}[R(s,a)]\right| \le 8\sqrt{\frac{\mathbb{E}\left[(R(s,a))^2\right]\cdot\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}} + \frac{8\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}.$$

Finally, for each $s\in\mathcal{S}$, for each $i\in[N]$, define

$$\mathcal{Z}_i = \mathbb{I}[s_{i,1} = s] - \mu(s).$$

Note that

$$\sum_{i=1}^{N} \mathcal{Z}_i = \sum_{i=1}^{N} \mathbb{I}[s_{1,i} = s] - N \cdot \mu(s).$$

Therefore, by Chernoff bound, with probability at least $1 - \delta/(9|\mathcal{S}|)$ we have

$$|\widehat{\mu}(s) - \mu(s)| \leq \sqrt{\frac{\log(18|\mathcal{S}|/\delta)}{K}}.$$

Hence, with probability at least $1 - \delta/9$, for all $s \in \mathcal{S}$, we have

$$|\widehat{\mu}(s) - \mu(s)| \leq \sqrt{\frac{\log(18|\mathcal{S}|/\delta)}{N}}$$

which we define to be event $\mathcal{E}_\mu$.

We finish the proof by applying a union bound over $\mathcal{E}_P$, $\mathcal{E}_R$, $\mathcal{E}_\mu$ and the events in Lemma 3.5.12 and Lemma 3.5.11. □

3.5.2.2 Perturbation Analysis

In this section, we establish a perturbation analysis on the value functions which is crucial for the analysis in the next section. We first recall a few basic facts.

Fact 3.5.1. *Let $|x| \leq 1/2$ be a real number, we have*

1. $x - x^2 \leq \log(1 + x) \leq x$;
2. $1 + x \leq e^x \leq 1 + 2|x|$.

We now prove the following lemma using the above facts.

Lemma 3.5.14. *Let $\overline{m} \geq 1$, $\bar{n} \geq n \geq 1$ be positive integers. Let $\varepsilon \in [0, 1/(8\bar{n})]$ be some real numbers. Let $p \in [1/\overline{m}, 1]^n$ be a vector with $\sum_{i=1}^n p_i \leq 1$. Let $\delta \in \mathbb{R}^n$ be a vector such that for each $1 \leq i \leq n$, $|\delta_i| \leq \varepsilon\sqrt{p_i/\overline{m}}$ and $|\sum_{i=1}^n \delta_i| \leq \varepsilon\bar{n}/\overline{m}$. For every $m \in [0, \overline{m}]$ and every $\Gamma \in \mathbb{R}^n$ such that $|\Gamma_i| \in [-\sqrt{p_i\overline{m}}, \sqrt{p_i\overline{m}}]$ for all $1 \leq i \leq n$, we have*

$$(1 - 8\bar{n}\varepsilon) \prod_{i=1}^{n} p_i^{p_i m + \Gamma_i} \leq \prod_{i=1}^{n} (p_i + \delta_i)^{p_i m + \Gamma_i} \leq (1 + 8\bar{n}\varepsilon) \prod_{i=1}^{n} p_i^{p_i m + \Gamma_i}.$$

Proof. Note that

$$\prod_{i=1}^{n} (p_i + \delta_i)^{p_i m + \Gamma_i} = \prod_{i=1}^{n} (p_i)^{p_i m + \Gamma_i} \cdot F$$

where

$$F = \prod_{i=1}^{n} \left(1 + \frac{\delta_i}{p_i}\right)^{p_i m + \Gamma_i}.$$

Clearly,

$$\log F = \sum_{i=1}^{n} (p_i m + \Gamma_i) \log\left(1 + \frac{\delta_i}{p_i}\right).$$

By the choice of δ, we have

$$\left|\frac{\delta_i}{p_i}\right| \le \varepsilon \le \frac{1}{2}.$$

Using Fact 3.5.1, for all $1 \le i \le n$, we have

$$\frac{\delta_i}{p_i} - \frac{\delta_i^2}{p_i^2} \le \log\left(1 + \frac{\delta_i}{p_i}\right) \le \frac{\delta_i}{p_i}.$$

Hence we,

$$|\log F| \le \left|\sum_{i=1}^{n} m\delta_i\right| + \sum_{i=1}^{n}\left(\frac{|\Gamma_i||\delta_i|}{p_i} + \frac{|\Gamma_i|\delta_i^2}{p_i^2} + \frac{m\delta_i^2}{p_i}\right).$$

Note that $|\sum_{i=1}^{n} m\delta_i| \le \varepsilon\bar{n}$, $|\Gamma_i||\delta_i| \le \varepsilon p_i$, $|\Gamma_i|\delta_i^2 \le \varepsilon p_i \cdot \varepsilon\sqrt{p_i/\overline{m}} \le \varepsilon^2 p_i^2$, and $m\delta_i^2 \le \varepsilon^2 p_i$. We have,

$$|\log F| \le \varepsilon\bar{n} + \varepsilon n + \varepsilon^2 n + \varepsilon^2 n \le 4\bar{n}\varepsilon.$$

By the choice of ε, we have $4\bar{n}\varepsilon \le 1/2$, and therefore

$$1 - 8\bar{n}\varepsilon \le \exp(\log F) \le 1 + 8\bar{n}\varepsilon.$$

□

In the following lemma, we show that for any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, with probability at least $1 - \delta$, the number of times (s, a, s') is visited can be upper bounded in terms of the $\delta/2$-quantile of the number of times (s, a) is visited and $P(s' \mid s, a)$.

Lemma 3.5.15. *For a given MDP M. Suppose a random trajectory*

$$T = ((s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_H, a_H, r_H), s_{H+1})$$

is obtained by executing a (possibly non-stationary) policy π in M. For any $(s, a, s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$, with probability at least $1 - \delta$, we have

$$\sum_{h=1}^{H} \mathbb{I}\left[(s, a, s') = (s_h, a_h, s_{h+1})\right] \le \frac{2\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s, a) = (s_h, a_h)\right]\right) + 4}{\delta} \cdot P(s' \mid s, a).$$

Proof. For each $h \in [H]$, define

$$I_h = \begin{cases} 1 & h = 1 \\ \mathbb{I}\left[\sum_{t=1}^{h-1} \mathbb{I}\left[(s_t, a_t) = (s, a)\right] \le \mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s_h, a_h) = (s, a)\right]\right) + 1\right] & h > 1 \end{cases}.$$

Let $\mathcal{E}_1$ be the event that for all $h \in [H]$, $I_h = 1$. By definition of $\mathcal{Q}_{\delta/2}$, we have $\Pr[\mathcal{E}_1] \ge 1 - \delta/2$.

For each $h \in [H]$, let $\mathcal{F}_h$ be the filtration induced by $\{(s_1, a_1, r_1), \ldots, (s_h, a_h, r_h)\}$. For each $h \in [H]$, define

$$X_h = \mathbb{I}\left[(s, a, s') = (s_h, a_h, s_{h+1})\right] \cdot I_h$$

and

$$Y_h = \mathbb{I}\left[(s,a) = (s_h, a_h)\right] \cdot I_h.$$

When $h = 1$, we have

$$\mathbb{E}[X_h] = \mathbb{E}[\mathbb{I}\left[(s,a) = (s_1, a_1)\right] \cdot P(s' \mid s, a) = \mathbb{E}[Y_h] \cdot P(s' \mid s, a).$$

When $h \in [H] \setminus \{1\}$, we have

$$\begin{aligned}\mathbb{E}[X_h \mid \mathcal{F}_{h-1}] =& \mathbb{E}[\mathbb{I}\left[(s,a,s') = (s_h, a_h, s_{h+1})\right] \cdot I_h \mid \mathcal{F}_{h-1}] \\ =& \mathbb{E}[\mathbb{I}\left[(s,a) = (s_h, a_h)\right] \cdot I_h \mid \mathcal{F}_{h-1}] \cdot P(s' \mid s, a),\end{aligned}$$

which implies

$$\mathbb{E}[X_h] = \mathbb{E}[\mathbb{I}\left[(s,a) = (s_h, a_h)\right] \cdot I_h] \cdot P(s' \mid s, a) = \mathbb{E}[Y_h] \cdot P(s' \mid s, a).$$

Note that

$$\sum_{h=1}^{H} Y_h \le \mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_h, a_h)\right]\right) + 2,$$

which implies

$$\mathbb{E}\left[\sum_{h=1}^{H} X_h\right] \le \left(\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_h, a_h)\right]\right) + 2\right) \cdot P(s' \mid s, a).$$

By Markov's inequality, with probability at least $1 - \delta/2$,

$$\sum_{h=1}^{H} X_h \le \frac{2\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_h, a_h)\right]\right) + 4}{\delta} \cdot P(s' \mid s, a).$$

which we denote as event $\mathcal{E}_2$.

Conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$ which happens with probability $1 - \delta$, we have

$$\sum_{h=1}^{H} \mathbb{I}\left[(s,a,s') = (s_h, a_h, s_{h+1})\right] = \sum_{h=1}^{H} X_h \le \frac{2\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_h, a_h)\right]\right) + 4}{\delta} \cdot P(s' \mid s, a).$$

□

In the following lemma, we show that for any $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, the number of times (s,a,s') is visited should be close to the number of times (s,a) is visited times $P(s' \mid s,a)$.

Lemma 3.5.16. *For a given MDP M. Suppose a random trajectory*

$$T = ((s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_H, a_H, r_H), s_{H+1})$$

is obtained by executing a policy π in M. For any $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$, with probability at least $1-\delta$, we have

$$\left|\sum_{h=1}^{H}\mathbb{I}\left[(s,a,s')=(s_t,a_t,s_{t+1})\right]-P(s'\mid s,a)\cdot\sum_{h=1}^{H}\mathbb{I}\left[(s,a)=(s_t,a_t)\right]\right| \leq\sqrt{\frac{4\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H}\mathbb{I}\left[(s,a)=(s_t,a_t)\right]\right)+8}{\delta}\cdot P(s'\mid s,a)}.$$

Proof. For each $h\in[H]$, define

$$I_h=\begin{cases}1 & h=1\\ \mathbb{I}\left[\sum_{t=1}^{h-1}\mathbb{I}\left[(s,a)=(s_t,a_t)\right]\leq\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H}\mathbb{I}\left[(s,a)=(s_h,a_h)\right]\right)+1\right] & h>1\end{cases}.$$

Let $\mathcal{E}_1$ be the event that for all $h\in[H]$, $I_h=1$. By definition of $\mathcal{Q}_{\delta/2}$, we have $\Pr\left[\mathcal{E}_1\right]\geq 1-\delta/2$.

For each $h\in[H]$, let $\mathcal{F}_h$ be the filtration induced by $\{(s_1,a_1,r_1),\ldots,(s_h,a_h,r_h)\}$. For each $h\in[H]$, define

$$X_h=\mathbb{I}\left[(s,a,s')=(s_h,a_h,s_{h+1})\right]\cdot I_h-P(s'\mid s,a)\mathbb{I}\left[(s,a)=(s_h,a_h)\right]\cdot I_h.$$

As we have shown in the proof of Lemma 3.5.15, for each $h\in[H]$, $\mathbb{E}[X_h]=0$. Moreover, for any $1\leq h'<h\leq H$, we have

$$\mathbb{E}[X_hX_{h'}]=\mathbb{E}[\mathbb{E}[X_hX_{h'}|\mathcal{F}_{h-1}]]=\mathbb{E}[X_{h'}\mathbb{E}[X_h|\mathcal{F}_{h-1}]]=0.$$

Therefore,

$$\mathbb{E}\left[\left(\sum_{h=1}^{H}X_h\right)^2\right]=\mathbb{E}\left[\sum_{h=1}^{H}X_h^2\right].$$

Note that for each $h\in[H]$.

$$\begin{aligned}X_h^2&=\left(\mathbb{I}\left[(s,a,s')=(s_h,a_h,s_{h+1})\right]\cdot I_h-P(s'\mid s,a)\mathbb{I}\left[(s,a)=(s_h,a_h)\right]\cdot I_h\right)^2\\&\leq I_h\cdot\left(\mathbb{I}\left[(s,a,s')=(s_h,a_h,s_{h+1})\right]+\left(P(s'\mid s,a)\right)^2\cdot\mathbb{I}\left[(s,a)=(s_h,a_h)\right]\right).\end{aligned}$$

As we have shown in the proof of Lemma 3.5.15, for each $h\in[H]$,

$$\mathbb{E}[I_h\cdot\mathbb{I}\left[(s,a,s')=(s_h,a_h,s_{h+1})\right]]=\mathbb{E}[I_h\cdot\mathbb{I}\left[(s,a)=(s_h,a_h)\right]]\cdot P(s'\mid s,a),$$

which implies

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_{h=1}^{H} X_h\right)^2\right] &= \mathbb{E}\left[\sum_{h=1}^{H} X_h^2\right] \\
&\leq \sum_{h=1}^{H} \mathbb{E}\left[I_h \cdot \left(\mathbb{I}\left[(s,a,s') = (s_h, a_h, s_{h+1})\right] + \left(P(s' \mid s,a)\right)^2 \cdot \mathbb{I}\left[(s,a) = (s_h,a_h)\right]\right)\right] \\
&\leq 2P(s' \mid s,a) \cdot \sum_{h=1}^{H} \mathbb{E}\left[I_h \cdot \mathbb{I}\left[(s,a) = (s_h,a_h)\right]\right] \\
&\leq 2P(s' \mid s,a) \cdot \left(\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_h,a_h)\right]\right) + 2\right).
\end{aligned}$$

By Chebyshev's inequality, we have with probability at least $1 - \delta/2$,

$$\left|\sum_{h=1}^{H} X_h\right| \leq \sqrt{\frac{4\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_h,a_h)\right]\right) + 8}{\delta} \cdot P(s' \mid s,a)},$$

which we denote as event $\mathcal{E}_2$.

Conditioned on $\mathcal{E}_1 \cap \mathcal{E}_2$ which happens with probability $1 - \delta$, we have

$$\begin{aligned}
&\left|\sum_{h=1}^{H} \mathbb{I}\left[(s,a,s') = (s_t, a_t, s_{t+1})\right] - P(s' \mid s,a) \cdot \sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_t,a_t)\right]\right| \\
\leq &\sqrt{\frac{4\mathcal{Q}_{\delta/2}\left(\sum_{h=1}^{H} \mathbb{I}\left[(s,a) = (s_t,a_t)\right]\right) + 8}{\delta} \cdot P(s' \mid s,a)}.
\end{aligned}$$

□

Using Lemma 3.5.14, Lemma 3.5.15 and Lemma 3.5.16, we now present the main result in this section, which shows that for two MDPs M and $\widehat{M}$ that are close enough in terms of rewards and transition probabilities, for any policy π, its value in $\widehat{M}$ should be lower bounded by that in M up to an error of ε.

Lemma 3.5.17. *Let $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ be an MDP and π be a policy. Let $0 < \varepsilon \leq 1/2$ be a parameter. For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, define $\overline{m}(s,a) = \mathcal{Q}^{\pi}_{\varepsilon/(12|\mathcal{S}||\mathcal{A}|)}(s,a)$. Let $\widehat{M} = (\mathcal{S}, \mathcal{A}, \widehat{P}, \widehat{R}, H, \widehat{\mu})$ be another MDP. If for all $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ with $\overline{m}(s,a) \geq 1$, we have*

$$|\widehat{P}(s'|s,a) - P(s'|s,a)| \leq \frac{\varepsilon}{96|\mathcal{S}|^2|\mathcal{A}|} \cdot \max\left(\sqrt{\frac{\varepsilon P(s'|s,a)}{72 \cdot \overline{m}(s,a) \cdot |\mathcal{S}||\mathcal{A}|}}, \frac{\varepsilon}{72 \cdot \overline{m}(s,a) \cdot |\mathcal{S}||\mathcal{A}|}\right),$$

$$\left|\mathbb{E}[\widehat{R}|s,a] - \mathbb{E}[R|s,a]\right| \leq \frac{\varepsilon}{24|\mathcal{S}||\mathcal{A}|} \cdot \max\left\{\sqrt{\frac{\mathbb{E}[(R(s,a))^2]}{\overline{m}(s,a)}}, \frac{1}{\overline{m}(s,a)}\right\}$$

and

$$|\mu(s) - \widehat{\mu}(s)| \leq \varepsilon/(6|\mathcal{S}|),$$

then

$$V^{\pi}_{\widehat{M},H} \geq V^{\pi}_{M,H} - \varepsilon.$$

Proof. Define $\mathcal{T} = (\mathcal{S} \times \mathcal{A})^H \times \mathcal{S}$ be set of all possible trajectories, where for each $T \in \mathcal{T}$, T has the form

$$((s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}).$$

For a trajectory $T \in \mathcal{T}$, for each $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we write

$$m_T(s, a) = \sum_{h=1}^{H} \mathbb{I}[(s_h, a_h) = (s, a)]$$

as the number times (s, a) is visited and

$$m_T(s, a, s') = \sum_{h=1}^{H} \mathbb{I}[(s_h, a_h, s_{h+1}) = (s, a, s')]$$

as the number of times (s, a, s') is visited. We say a trajectory

$$T = ((s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}) \in \mathcal{T}$$

is *compatible* with a (possibly non-stationary) policy π if for all $h \in [H]$,

$$a_h = \pi_h(s_h).$$

For a (possibly non-stationary) policy π, we use $\mathcal{T}^{\pi} \subseteq \mathcal{T}$ to denote the set of all trajectories that are compatible with π.

For an MDP $M = (\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ and a (possibly non-stationary) policy π, for a trajectory T that is compatible with π, we write

$$p(T, M, \pi) = \mu(s_1) \cdot \prod_{h=1}^{H} P(s_{h+1} \mid s_h, a_h) = \mu(s_1) \cdot \prod_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}} P(s' \mid s, a)^{m_T(s,a,s')}$$

to be the probability of T when executing π in M. Here we assume $0^0 = 1$.

Using these definitions, we have

$$V^{\pi}_{M,H} = \sum_{T \in \mathcal{T}^{\pi}} p(T, M, \pi) \cdot \left(\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} m_T(s, a) \cdot \mathbb{E}[R(s, a)] \right).$$

Note that for any trajectory $T = ((s_1, a_1), (s_2, a_2), \ldots, (s_H, a_H), s_{H+1}) \in \mathcal{T}^{\pi}$, if $p(T, M, \pi) > 0$, by Assumption 3.2.1,

$$\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} m_T(s, a) \cdot \mathbb{E}[R(s, a)] \leq 1.$$

We define $\mathcal{T}_1^\pi \subseteq \mathcal{T}^\pi$ be the set of trajectories that for each $T \in \mathcal{T}_1^\pi$, for each $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$m_T(s,a) \leq \overline{m}(s,a).$$

By a union bound over all $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\sum_{T \in \mathcal{T}^\pi \setminus \mathcal{T}_1^\pi} p(T, M, \pi) \leq \varepsilon/6.$$

We also define $\mathcal{T}_2^\pi \subseteq \mathcal{T}^\pi$ be the set of trajectories that for each $T \in \mathcal{T}_2^\pi$, for each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$,

$$|m_T(s,a,s') - m_T(s,a) \cdot P(s'|s,a)| \leq \sqrt{\frac{6P(s'|s,a)(4\overline{m}(s,a)+8)|\mathcal{S}||\mathcal{A}|}{\varepsilon}}.$$

By Lemma 3.5.16 and a union bound over all $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\sum_{T \in \mathcal{T}^\pi \setminus \mathcal{T}_2^\pi} p(T, M, \pi) \leq \varepsilon/6.$$

Finally, we define $\mathcal{T}_3^\pi \subseteq \mathcal{T}^\pi$ be the set of trajectories such that for each $T \in \mathcal{T}_3^\pi$, for each $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$,

$$m_T(s,a,s') \leq \frac{6|\mathcal{S}||\mathcal{A}|(2\overline{m}(s,a)+4)}{\varepsilon} \cdot P(s'|s,a).$$

By Lemma 3.5.15 and a union bound over all $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\sum_{T \in \mathcal{T}^\pi \setminus \mathcal{T}_3^\pi} p(T, M, \pi) \leq \varepsilon/6.$$

Thus, by defining $\mathcal{T}_{\text{pruned}}^\pi = \mathcal{T}_1^\pi \cap \mathcal{T}_2^\pi \cap \mathcal{T}_3^\pi$, we have

$$\sum_{T \in \mathcal{T}_{\text{pruned}}^\pi} p(T, M, \pi) \cdot \left(\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} m_T(s,a) \cdot \mathbb{E}[R(s,a)] \right) \geq V_{M,H}^\pi - \varepsilon/2.$$

Note that for each $T \in \mathcal{T}_{\text{pruned}}^\pi$ with

$$T = ((s_1,a_1),(s_2,a_2),\ldots,(s_H,a_H),s_{H+1}),$$

for each state-action $(s,a) \in \mathcal{S} \times \mathcal{A}$, we must have $m_T(s,a) \leq \overline{m}(s,a)$. This is because $T \in \mathcal{T}_1^\pi$ Moreover, for any $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, if $m_T(s,a,s') \geq 1$, then

$$P(s' \mid s,a) \geq \frac{\varepsilon}{36|\mathcal{S}||\mathcal{A}|\overline{m}(s,a)}.$$

This is because $T \in \mathcal{T}_3^\pi$, and if

$$P(s' \mid s,a) < \frac{\varepsilon}{36|\mathcal{S}||\mathcal{A}|\overline{m}(s,a)},$$

then

$$m_T(s,a,s') \le \frac{6|\mathcal{S}||\mathcal{A}|(2\overline{m}(s,a)+4)}{\varepsilon} \cdot P(s'|s,a) \le \frac{36|\mathcal{S}||\mathcal{A}|\overline{m}(s,a)}{\varepsilon} \cdot P(s'|s,a) < 1.$$

For each $(s,a) \in \mathcal{S} \times \mathcal{A}$, define

$$\mathcal{S}_{s,a} = \left\{ s' \in \mathcal{S} \mid P(s'|s,a) \ge \frac{\varepsilon}{36|\mathcal{S}||\mathcal{A}|\overline{m}(s,a)} \right\}.$$

Therefore, for each $(s,a) \in \mathcal{S} \times \mathcal{A}$,

$$\prod_{s' \in \mathcal{S}} P(s' \mid s,a)^{m_T(s,a,s')} = \prod_{s' \in \mathcal{S}_{s,a}} P(s' \mid s,a)^{m_T(s,a,s')}$$

and

$$\prod_{s' \in \mathcal{S}} \widehat{P}(s' \mid s,a)^{m_T(s,a,s')} = \prod_{s' \in \mathcal{S}_{s,a}} \widehat{P}(s' \mid s,a)^{m_T(s,a,s')}$$

with

$$\begin{aligned} |m_T(s,a,s') - m_T(s,a) \cdot P(s'|s,a)| &\le \sqrt{\frac{6P(s'|s,a)(4\overline{m}(s,a)+8)|\mathcal{S}||\mathcal{A}|}{\varepsilon}} \\ &\le \sqrt{\frac{72P(s'|s,a)\overline{m}(s,a)|\mathcal{S}||\mathcal{A}|}{\varepsilon}}. \end{aligned}$$

Note that $\sum_{s' \in \mathcal{S}} \widehat{P}(s'|s,a) - P(s'|s,a) = 0$, which implies

$$\left| \sum_{s' \in \mathcal{S}_{s,a}} \widehat{P}(s'|s,a) - P(s'|s,a) \right| = \left| \sum_{s' \notin \mathcal{S}_{s,a}} P(s'|s,a) - \widehat{P}(s'|s,a) \right| \le \frac{\varepsilon}{96|\mathcal{S}||\mathcal{A}|} \cdot \frac{\varepsilon}{72 \cdot \overline{m}(s,a) \cdot |\mathcal{S}||\mathcal{A}|}.$$

By applying Lemma 3.5.14 and settting $\bar{n}$ to be $|\mathcal{S}|$, n to be $|\mathcal{S}_{s,a}|$, ε to be $\varepsilon/(96|\mathcal{S}|^2|\mathcal{A}|)$, and $\overline{m}$ to be $72 \cdot \overline{m}(s,a) \cdot |\mathcal{S}||\mathcal{A}|/\varepsilon$, we have

$$\prod_{s' \in \mathcal{S}_{s,a}} \widehat{P}(s' \mid s,a)^{m_T(s,a,s')} \ge \left(1 - \frac{\varepsilon}{12|\mathcal{S}||\mathcal{A}|}\right) \prod_{s' \in \mathcal{S}_{s,a}} P(s' \mid s,a)^{m_T(s,a,s')}.$$

Therefore,

$$\prod_{(s,a) \in \mathcal{S} \times \mathcal{A}} \prod_{s' \in \mathcal{S}} \widehat{P}(s' \mid s,a)^{m_T(s,a,s')} \ge \left(1 - \frac{\varepsilon}{12|\mathcal{S}||\mathcal{A}|}\right)^{|\mathcal{S}||\mathcal{A}|} \prod_{(s,a) \in \mathcal{S} \times \mathcal{A}} \prod_{s' \in \mathcal{S}} P(s' \mid s,a)^{m_T(s,a,s')},$$

which implies

$$\prod_{(s,a) \in \mathcal{S} \times \mathcal{A}} \prod_{s' \in \mathcal{S}} \widehat{P}(s' \mid s,a)^{m_T(s,a,s')} \ge (1 - \varepsilon/6) \prod_{(s,a) \in \mathcal{S} \times \mathcal{A}} \prod_{s' \in \mathcal{S}} P(s' \mid s,a)^{m_T(s,a,s')}.$$

For the summation of rewards, we have

$$\begin{aligned}
&\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right| \\
=&\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)=1} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right| \\
+&\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)>1} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|.
\end{aligned}$$

For those $(s,a)\in\mathcal{S}\times\mathcal{A}$ with $m_T(s,a)>1$, we have $\overline{m}(s,a)>1$. By Lemma 3.2.1, we have

$$\begin{aligned}
&|\mathbb{E}[\widehat{R}(s,a)]-\mathbb{E}[R(s,a)]| \\
\leq&\frac{\varepsilon}{24|\mathcal{S}||\mathcal{A}|}\cdot\max\left\{\sqrt{\frac{\mathbb{E}[(R(s,a))^2]}{\overline{m}(s,a)}},\frac{1}{\overline{m}(s,a)}\right\}\leq\max\left\{\frac{\varepsilon}{12H},\frac{\varepsilon}{24|\mathcal{S}||\mathcal{A}|\overline{m}(s,a)}\right\}.
\end{aligned}$$

Since $\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\leq H$ and $m_T(s,a)\leq\overline{m}(s,a)$, we have

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)>1} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|\leq\frac{\varepsilon}{8}.$$

For those $(s,a)\in\mathcal{S}\times\mathcal{A}$ with $m_T(s,a)=1$, we have

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)=1} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|\leq\frac{\varepsilon}{24}.$$

Thus,

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|\leq\frac{\varepsilon}{6}.$$

For each $T\in\mathcal{T}^{\pi}_{\text{pruned}}$ with

$$T=((s_1,a_1),(s_2,a_2),\ldots,(s_H,a_H),s_{H+1}),$$

we have

$$\begin{aligned}
&p(T,\widehat{M},\pi)\cdot\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\mathbb{E}[\widehat{R}(s,a)]\right) \\
=&\widehat{\mu}(s_0)\cdot\prod_{(s,a)\in\mathcal{S}\times\mathcal{A}}\prod_{s'\in\mathcal{S}}\widehat{P}(s'\mid s,a)^{m_T(s,a,s')}\cdot\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\mathbb{E}[\widehat{R}(s,a)]\right) \\
\geq&(\mu(s_0)-\varepsilon/(6|\mathcal{S}|))\cdot(1-\varepsilon/6)\prod_{(s,a)\in\mathcal{S}\times\mathcal{A}}\prod_{s'\in\mathcal{S}}P(s'\mid s,a)^{m_T(s,a,s')}\cdot\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\mathbb{E}[R(s,a)]-\varepsilon/6\right).
\end{aligned}$$

Since

$$\sum_{T\in\mathcal{T}^{\pi}_{\text{pruned}}} p(T,M,\pi)\cdot\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\mathbb{E}[R(s,a)]\right)\geq V^{\pi}_{M,H}-\varepsilon/2,$$

we have

$$V^{\pi}_{\widehat{M},H}\geq\sum_{T\in\mathcal{T}^{\pi}_{\text{pruned}}} p(T,\widehat{M},\pi)\cdot\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\mathbb{E}[\widehat{R}(s,a)]\right)\geq V^{\pi}_{M,H}-\varepsilon.$$

□

Now we show that for two MDPs M and $\widehat{M}$ with the same transition probabilities and close enough rewards, for any policy π, its value in $\widehat{M}$ should be upper bounded by that in $\widehat{M}$ up to an error of ε.

Lemma 3.5.18. *Let $M=(\mathcal{S},\mathcal{A},P,R,H,\mu)$ be an MDP and π be a policy. Let $0<\varepsilon\leq 1/2$ be a parameter. For each $(s,a)\in\mathcal{S}\times\mathcal{A}$, define $\overline{m}(s,a)=\mathcal{Q}^{\pi}_{\varepsilon/(12|\mathcal{S}||\mathcal{A}|)}(s,a)$. Let $\widehat{M}=(\mathcal{S},\mathcal{A},P,\widehat{R},H,\mu)$ be another MDP. If for all $(s,a)\in\mathcal{S}\times\mathcal{A}$ with $\overline{m}(s,a)\geq 1$, we have*

$$\left|\mathbb{E}[\widehat{R}|s,a]-\mathbb{E}[R|s,a]\right|\leq\frac{\varepsilon}{24|\mathcal{S}||\mathcal{A}|}\cdot\max\left\{\sqrt{\frac{\mathbb{E}[(R(s,a))^2]}{\overline{m}(s,a)}},\frac{1}{\overline{m}(s,a)}\right\}.$$

then

$$V^{\pi}_{\widehat{M},H}\leq V^{\pi}_{M,H}+\varepsilon.$$

Proof. We adopt the same notations as in the proof of Lemma 3.5.17. Recall that

$$V^{\pi}_{M,H}=\sum_{T\in\mathcal{T}^{\pi}} p(T,M,\pi)\cdot\left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\mathbb{E}[R(s,a)]\right).$$

and

$$\sum_{T\in\mathcal{T}^{\pi}\setminus\mathcal{T}^{\pi}_1} p(T,M,\pi)\leq\varepsilon/6.$$

As in the proof of Lemma 3.5.17, for the summation of rewards, we have

$$\begin{aligned}
&\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|\\
=&\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)=1} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|\\
+&\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)>1} m_T(s,a)\cdot\left|\mathbb{E}\left[R(s,a)\right]-\mathbb{E}\left[\widehat{R}(s,a)\right]\right|.
\end{aligned}$$

For each $T \in \mathcal{T}_1^\pi$, for each $(s,a) \in \mathcal{S} \times \mathcal{A}$, we must have $m_T(s,a) \leq \overline{m}(s,a)$. Therefore, for those $(s,a) \in \mathcal{S} \times \mathcal{A}$ with $m_T(s,a) > 1$, by Lemma 3.2.1, we have

$$\begin{aligned}
&|\mathbb{E}[\widehat{R}(s,a)] - \mathbb{E}[R(s,a)]| \\
\leq &\frac{\varepsilon}{24|\mathcal{S}||\mathcal{A}|} \cdot \max\left\{\sqrt{\frac{\mathbb{E}[(R(s,a))^2]}{\overline{m}(s,a)}}, \frac{1}{\overline{m}(s,a)}\right\} \leq \max\left\{\frac{\varepsilon}{12H}, \frac{\varepsilon}{24|\mathcal{S}||\mathcal{A}|\overline{m}(s,a)}\right\}.
\end{aligned}$$

Since $\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a) \leq H$ and $m_T(s,a) \leq \overline{m}(s,a)$, we have

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)>1} m_T(s,a) \cdot \left|\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\widehat{R}(s,a)\right]\right| \leq \frac{\varepsilon}{8}.$$

For those $(s,a) \in \mathcal{S} \times \mathcal{A}$ with $m_T(s,a) = 1$, we have

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}|m_T(s,a)=1} m_T(s,a) \cdot \left|\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\widehat{R}(s,a)\right]\right| \leq \frac{\varepsilon}{24}.$$

Thus,

$$\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a) \cdot \left|\mathbb{E}\left[R(s,a)\right] - \mathbb{E}\left[\widehat{R}(s,a)\right]\right| \leq \frac{\varepsilon}{6}.$$

Hence,

$$\begin{aligned}
V_{\widehat{M},H}^\pi &\leq \sum_{T\in\mathcal{T}^\pi\setminus\mathcal{T}_1^\pi} p(T,M,\pi) + \sum_{T\in\mathcal{T}_1^\pi} p(T,M,\pi) \cdot \left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a) \cdot \mathbb{E}[\widehat{R}(s,a)]\right) \\
&\leq \varepsilon/6 + \sum_{T\in\mathcal{T}_1^\pi} p(T,M,\pi) \cdot \left(\sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} m_T(s,a) \cdot \mathbb{E}[R(s,a)] + \varepsilon/6\right) \\
&\leq V_{M,H}^\pi + \varepsilon.
\end{aligned}$$

□

3.5.2.3 Pessimistic Planning

We now present our final algorithm in the RL setting. The formal description is provided in Algorithm 6. In our algorithm, we first invoke Algorithm 4 to collect a dataset D and then invoke Algorithm 5 to estimate $\mathcal{Q}_\varepsilon^{\mathsf{st}}(s,a)$ for some properly chosen ε. We then use the estimators in Lemma 4.4.5 to define $\widehat{P}$, $\widehat{R}$ and $\widehat{\mu}$. Note that Lemma 4.4.5 not only provides an estimator but also provides a computable confidence interval for $\widehat{P}$ and $\widehat{\mu}$, which we also utilize in our algorithm.

At this point, a natural idea is to find the optimal policy with respect to the MDP $\widehat{M}$ defined by $\widehat{P}$, $\widehat{R}$ and $\widehat{\mu}$. However, our Lemma 3.5.17 only provides a lower bound guarantee for

Algorithm 6 Pessimistic Planning

1: **Input:** desired accuracy ε, failure probability δ
2: **Output:** ε-optimal policy π
3: Invoke Algorithm 4 with

$$N = 2^{66} \cdot (|\mathcal{S}| + 1)^{24(|\mathcal{S}|+1)} \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta) \cdot |\mathcal{S}|^7|\mathcal{A}|^5/\varepsilon^5$$

and receive

$$D = \left(\left(\left(s_{i,t}, a_{i,t}, r_{i,t}, s'_{i,t}\right)\right)_{t=1}^{|\mathcal{S}||\mathcal{A}|\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot H}\right)_{i=1}^{N}$$

4: Invoke Algorithm 5 with

$$\varepsilon_{\text{est}} = \frac{\varepsilon}{32768 \cdot |\mathcal{S}||\mathcal{A}|(|\mathcal{S}| + 1)^{12(|\mathcal{S}|+1)}}$$

and $\delta = \delta$, and receive estimates $\overline{m}^{\mathsf{st}} : \mathcal{S} \times \mathcal{A} \to \mathbb{N}$
5: For each $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, for each $i \in [N]$ and $t \in \left[|\mathcal{S}||\mathcal{A}| \cdot |\mathcal{A}|^{2|\mathcal{S}|} \cdot H\right]$. define $\mathsf{Trunc}_{i,t}(s, a)$, $\widehat{P}(s' \mid s, a)$, $\widehat{R}(s, a)$ and $\widehat{\mu}(s)$ as in Lemma 3.5.13
6: Define $\mathcal{M}$ to be a set of MDPs where for each $M = (\mathcal{S}, \mathcal{A}, \widetilde{P}, \widehat{R}, H, \widetilde{\mu}) \in \mathcal{M}$,

$$\left|\widehat{P}(s' \mid s, a) - \widetilde{P}(s' \mid s, a)\right| \le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s, a) \cdot N \cdot \varepsilon_{\text{est}}}, 32\sqrt{\frac{\widehat{P}(s' \mid s, a) \cdot \log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s, a) \cdot N \cdot \varepsilon_{\text{est}}}}\right\}$$

and

$$|\widehat{\mu}(s) - \widetilde{\mu}(s)| \le \sqrt{\frac{\log(18|\mathcal{S}|/\delta)}{N}}$$

for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$
7: For each (possibly non-stationary) policy π, define $\underline{V}^{\pi} = \min_{M\in\mathcal{M}} V^{\pi}_{M,H}$
8: **return** $\operatorname{argmax}_{\pi}\underline{V}^{\pi}$

$V^{\pi}_{\widehat{M},H}$ without any upper bound guarantee. We resolve this issue by pessimistic planning. More specifically, for any policy π, we define its pessimistic value to be

$$\underline{V}^{\pi} = \min_{M\in\mathcal{M}} V^{\pi}_{M,H}$$

where $\mathcal{M}$ includes all MDPs whose transition probabilities are within the confidence interval provided in Lemma 4.4.5. We simply return the policy π that maximizes $\underline{V}^{\pi}$. Since the true MDP lies in $\mathcal{M}$, $\underline{V}^{\pi}$ is never an overestimate. On the other hand, Lemma 3.5.17 guarantees that $\underline{V}^{\pi}$ is also lower bounded by $V^{\pi}_{M,H}$ up to an error of ε. Therefore, $\underline{V}^{\pi}$ provides an accurate estimate to the true value of π. However, note that Lemma 4.4.5 does not provide a computable confidence interval for the rewards. Fortunately, as we have shown in Lemma 3.5.18, perturbation on the rewards will not significantly increase the value of the policy and thus the estimate is still accurate.

We now present the formal analysis for our algorithm.

Theorem 3.5.19. *With probability at least $1-\delta$, for any (possibly non-stationary) policy π,*

$$\left|\underline{V}^{\pi} - V^{\pi}_{M,H}\right| \le \varepsilon/2$$

where $\underline{V}^{\pi}$ is defined in Line 7 of Algorithm 6. Moreover, Alogrithm 6 samples at most

$$2^{66} \cdot (|\mathcal{S}|+1)^{24(|\mathcal{S}|+1)} \cdot |\mathcal{A}|^{2|\mathcal{S}|} \cdot \log(12|\mathcal{S}|^2|\mathcal{A}|/\delta) \cdot |\mathcal{S}|^8|\mathcal{A}|^6/\varepsilon^5.$$

trajectories.

Proof. By Lemma 3.5.13, with probability at least $1-\delta$, for all $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$, we have

$$\begin{aligned}\left|\widehat{P}(s' \mid s,a) - P(s' \mid s,a)\right| &\le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}, 32\sqrt{\frac{\widehat{P}(s' \mid s,a)\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\overline{m}^{\mathsf{st}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}}\right\} \\ &\le \max\left\{\frac{512\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}, 64\sqrt{\frac{P(s' \mid s,a)\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}}\right\},\end{aligned}$$

$$\left|\widehat{R}(s' \mid s,a) - \mathbb{E}[R(s,a)]\right| \le 8\sqrt{\frac{\mathbb{E}\left[(R(s,a))^2\right]\cdot\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}}} + \frac{8\log(18|\mathcal{S}||\mathcal{A}|/\delta)}{\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a)\cdot N\cdot\varepsilon_{\mathrm{est}}},$$

and

$$|\widehat{\mu}(s) - \mu(s)| \le \sqrt{\frac{\log(18|\mathcal{S}|/\delta)}{N}}.$$

In the remaining part of the analysis, we condition on the above event.

by Lemma 3.5.10, for any (possibly non-stationary) policy π,

$$\mathcal{Q}^{\mathsf{st}}_{\varepsilon_{\mathrm{est}}}(s,a) \ge \frac{1}{4096\cdot|\mathcal{S}|^{12|\mathcal{S}|}}\cdot\varepsilon/(24|\mathcal{S}||\mathcal{A}|)\cdot\mathcal{Q}^{\pi}_{\varepsilon/(24|\mathcal{S}||\mathcal{A}|)}(s,a).$$

Let $M=(\mathcal{S},\mathcal{A},P,R,H,\mu)$ be the true MDP. By Lemma 3.5.17, for any (possibly non-stationary) policy π, for any $\overline{M}\in\mathcal{M}$, we have

$$V^{\pi}_{\overline{M},H} \ge V^{\pi}_{M,H} - \varepsilon/2.$$

Moreover, $M'=(\mathcal{S},\mathcal{A},P,\widehat{R},H,\mu)\in\mathcal{M}$. Therefore, by Lemma 3.5.18,

$$V^{\pi}_{M',H} \le V^{\pi}_{M,H} + \varepsilon/2.$$

Consequently,

$$\left|\underline{V}^{\pi} - V^{\pi}_{M,H}\right| \le \varepsilon/2.$$

Finally, the algorithm samples at most

$$\begin{aligned}&(N + \lceil 300\log(6|\mathcal{S}||\mathcal{A}|/\delta)/\varepsilon_{\mathrm{est}}\rceil)\times|\mathcal{S}|\times|\mathcal{A}|\times|\mathcal{A}|^{2|\mathcal{S}|} \\ \le& 2^{66}\cdot(|\mathcal{S}|+1)^{24(|\mathcal{S}|+1)}\cdot|\mathcal{A}|^{2|\mathcal{S}|}\cdot\log(18|\mathcal{S}|^2|\mathcal{A}|/\delta)\cdot|\mathcal{S}|^8|\mathcal{A}|^6/\varepsilon^5.\end{aligned}$$

trajectories. □

Theorem 3.5.19 immediately implies Theorem 3.1.2.

Part II

RL with Large State Spaces

Chapter 4

RL with Large State Spaces: Upper Bound in the Online Setting

4.1 Introduction

In online RL, the agent interacts with the environment episodically, where each episode consists of H steps. The goal of the agent is to interact with the environment strategically such that after a certain number of interactions, sufficient information is collected so that the agent can act nearly optimally afterward. The performance of an agent is measured by the *regret*, which is defined as the difference between the total rewards collected by the agent and those a best possible agent would collect.

Without additional assumptions on the structure of the MDP, the best possible algorithm achieves a regret bound of $\widetilde{\Theta}(\sqrt{H|\mathcal{S}||\mathcal{A}|T})$ [11], where T is the total number of steps the agent interacts with the environment. In other words, the algorithm learns to interact with the environment nearly as well as an optimal agent after roughly $H|\mathcal{S}||\mathcal{A}|$ steps. This regret bound, however, can be unacceptably large in practice. E.g., the game of Go has a state space with size 3^{361}, and the state space of certain robotics applications can even be continuous. Practitioners apply function approximation schemes to tackle this issue, i.e., the value of a state-action pair is approximated by a function which is able to predict the value of unseen state-action pairs given a few training samples. The most commonly used function approximators are deep neural networks (DNN) which have achieved remarkable success in playing video games [58], the game of Go [77], and controlling robots [4]. Nevertheless, despite the outstanding achievements in solving real-world problems, no convincing theoretical guarantees were known about RL with general value function approximators like DNNs.

Recently, there is a line of research trying to understand RL with simple function approximators, e.g. linear functions. For instance, given a feature extractor which maps state-action pairs to d-dimensional feature vectors, [15, 25, 26, 40, 59, 96, 102, 103, 108, 109] developed algorithms with regret bound proportional to $\mathrm{poly}(dH)\sqrt{T}$ which is independent of the size of $\mathcal{S} \times \mathcal{A}$. Although being much more efficient than algorithms for the tabular setting, these algorithms require a well-designed feature extractor and also make restricted assumptions on the transition model. This severely limits the scope that these approaches can be applied to, since

obtaining a good feature extractor is by no means easy.

In this section, we develop a provably efficient (both computationally and statistically) Q-learning algorithm that works with general value function approximators. We show that our algorithm enjoys a regret bound of $\widetilde{O}(\mathrm{poly}(dH)\sqrt{T})$ where d is a complexity measure of the function class that depends on the *eluder dimension* [73] and log-covering numbers. Our theory generalizes the linear MDP assumption in [40, 102] to general function classes, and our algorithm provides comparable regret bounds when applied to the linear case.

4.2 Notations and Assumptions

Episodic Markov Decision Process. In this section, we assume the reward $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is deterministic. In the online RL setting, the agent aims to learn the optimal policy by interacting with the environment during a number of episodes. For each $k \in [K]$, at the beginning of the k-th episode, the agent chooses a policy π^k which induces a trajectory, based on which the agent chooses policies for later episodes. Throughout this section, we define $T := KH$ to be the total number of steps that the agent interacts with the environment.

We adopt the following regret definition in this section.

Definition 4.2.1. The regret of an algorithm $\mathcal{A}$ after K episodes is defined as

$$\mathrm{Reg}(K) = \sum_{k=1}^{K} V_1^* \left(s_1^k\right) - V_1^{\pi^k} \left(s_1^k\right)$$

where π^k is the policy played by algorithm $\mathcal{A}$ at the k-th episode.

Additional Notations. For a function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, define

$$\|f\|_\infty = \max_{(s,a)\in\mathcal{S}\times\mathcal{A}} |f(s, a)|.$$

Similarly, for a function $v : \mathcal{S} \to \mathbb{R}$, define

$$\|v\|_\infty = \max_{s\in\mathcal{S}} |v(s)|.$$

Given a dataset

$$\mathcal{D} = \{(s_i, a_i, q_i)\}_{i=1}^{|\mathcal{D}|} \subseteq \mathcal{S} \times \mathcal{A} \times \mathbb{R},$$

for a function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, define

$$\|f\|_{\mathcal{D}} = \left(\sum_{t=1}^{|\mathcal{D}|} \left(f(s_t, a_t) - q_t\right)^2\right)^{1/2}.$$

For a set of state-action pairs $\mathcal{Z} \subseteq \mathcal{S} \times \mathcal{A}$, for a function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, define

$$\|f\|_{\mathcal{Z}} = \left(\sum_{(s,a)\in\mathcal{Z}} \left(f(s, a)\right)^2\right)^{1/2}.$$

For a set of functions $\mathcal{F} \subseteq \{f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}\}$, we define the width function of a state-action pair (s, a) as

$$w(\mathcal{F}, s, a) = \max_{f, f' \in \mathcal{F}'} f(s, a) - f'(s, a).$$

Our Assumptions. We make the following assumption throughout this chapter.

Assumption 4.2.1. *There exists a set of functions $\mathcal{F} \subseteq \{f : \mathcal{S} \times \mathcal{A} \to [0, H+1]\}$, such that for any $V : \mathcal{S} \to [0, H]$, there exists $f_V \in \mathcal{F}$ which satisfies*

$$f_V(s, a) = R(s, a) + \sum_{s' \in \mathcal{S}} P(s' \mid s, a) V(s') \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}. \tag{4.1}$$

Intuitively, Assumption 4.2.1 requires that for any $V : \mathcal{S} \to [0, H]$, after applying the Bellman backup operator, the resulting function lies in the function class $\mathcal{F}$. We note that Assumption 4.2.1 is very general and includes many previous assumptions as special cases. For instance, for the tabular RL setting, $\mathcal{F}$ can be the entire function space of $\mathcal{S} \times \mathcal{A} \to [0, H+1]$. For linear MDPs [40, 96, 102, 103] where both the reward function $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$ and the transition operator $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ are linear functions of a given feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, $\mathcal{F}$ can be defined as the class of linear functions with respect to ϕ. In practice, when $\mathcal{F}$ is a function class with sufficient expressive power (e.g. deep neural networks), Assumption 4.2.1 (approximately) holds.

The complexity of $\mathcal{F}$ determines the learning complexity of the RL problem under consideration. To characterize the complexity of $\mathcal{F}$, we use the following definition of eluder dimension which was first introduced in [73] to characterize the complexity of function classes in bandits problems.

Definition 4.2.2 (Eluder dimension)**.** Let $\varepsilon \geq 0$ and $\mathcal{Z} = \{(s_i, a_i)\}_{i=1}^n \subseteq \mathcal{S} \times \mathcal{A}$ be a sequence of state-action pairs.

- A state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ is *ε-dependent* on $\mathcal{Z}$ with respect to $\mathcal{F}$ if any $f, f' \in \mathcal{F}$ satisfying $\|f - f'\|_{\mathcal{Z}} \leq \varepsilon$ also satisfies $|f(s, a) - f'(s, a)| \leq \varepsilon$.
- An (s, a) is *ε-independent* of $\mathcal{Z}$ with respect to $\mathcal{F}$ if (s, a) is not ε-dependent on $\mathcal{Z}$.
- The *ε-eluder dimension* $\dim_E(\mathcal{F}, \varepsilon)$ of a function class $\mathcal{F}$ is the length of the longest sequence of elements in $\mathcal{S} \times \mathcal{A}$ such that, for some $\varepsilon' \geq \varepsilon$, every element is ε'-independent of its predecessors.

It has been shown in [73] that $\dim_E(\mathcal{F}, \varepsilon) \leq |\mathcal{S}||\mathcal{A}|$ when $\mathcal{S}$ and $\mathcal{A}$ are finite. When $\mathcal{F}$ is the class of linear functions, i.e., $f_\theta(s, a) = \theta^\top \phi(s, a)$ for a given feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, $\dim_E(\mathcal{F}, \varepsilon) = O(d \log(1/\varepsilon))$. When $\mathcal{F}$ is the class generalized linear functions of the form $f_\theta(s, a) = g(\theta^\top \phi(s, a))$ where g is an increasing continuously differentiable function, $\dim_E(\mathcal{F}, \varepsilon) = O(dr^2 \log(\overline{h}/\varepsilon))$ where

$$r = \frac{\sup_{\theta, (s,a) \in \mathcal{S} \times \mathcal{A}} g'(\theta^\top \phi(s, a))}{\inf_{\theta, (s,a) \in \mathcal{S} \times \mathcal{A}} g'(\theta^\top \phi(s, a))}$$

and

$$\overline{h} = \sup_{\theta, (s,a) \in \mathcal{S} \times \mathcal{A}} g'(\theta^\top \phi(s, a)).$$

In [63], it has been shown that when $\mathcal{F}$ is the class of quadratic functions, i.e., $f_\Lambda(s,a) = \phi(s,a)^\top \Lambda \phi(s,a)$ where $\Lambda \in \mathbb{R}^{d\times d}$, $\dim_E(\mathcal{F},\varepsilon) = O(d^2\log(1/\varepsilon))$.

We further assume the function class $\mathcal{F}$ and the state-action pairs $\mathcal{S}\times\mathcal{A}$ have bounded complexity in the following sense.

Assumption 4.2.2. *For any $\varepsilon > 0$, the following holds:*

1. *there exists an ε-cover $\mathcal{C}(\mathcal{F},\varepsilon) \subseteq \mathcal{F}$ with size $|\mathcal{C}(\mathcal{F},\varepsilon)| \le \mathcal{N}(\mathcal{F},\varepsilon)$, such that for any $f \in \mathcal{F}$, there exists $f' \in \mathcal{C}(\mathcal{F},\varepsilon)$ with $\|f - f'\|_\infty \le \varepsilon$;*
2. *there exists an ε-cover $\mathcal{C}(\mathcal{S}\times\mathcal{A},\varepsilon)$ with size $|\mathcal{C}(\mathcal{S}\times\mathcal{A},\varepsilon)| \le \mathcal{N}(\mathcal{S}\times\mathcal{A},\varepsilon)$, such that for any $(s,a) \in \mathcal{S}\times\mathcal{A}$, there exists $(s',a') \in \mathcal{C}(\mathcal{S}\times\mathcal{A},\varepsilon)$ with $\max_{f\in\mathcal{F}} |f(s,a) - f(s',a')| \le \varepsilon$.*

Assumption 4.2.2 requires both the function class $\mathcal{F}$ and the state-action pairs $\mathcal{S}\times\mathcal{A}$ have bounded covering numbers. Since our regret bound depends *logarithmically* on $\mathcal{N}(\mathcal{F},\cdot)$ and $\mathcal{N}(\mathcal{S}\times\mathcal{A},\cdot)$, it is acceptable for the covers to have exponential size. In particular, when $\mathcal{S}$ and $\mathcal{A}$ are finite, it is clear that $\log\mathcal{N}(\mathcal{F},\varepsilon) = \widetilde{O}(|\mathcal{S}||\mathcal{A}|)$ and $\log\mathcal{N}(\mathcal{S}\times\mathcal{A},\varepsilon) = \log(|\mathcal{S}||\mathcal{A}|)$. For the case of d-dimensional linear functions and generalized linear functions, $\log\mathcal{N}(\mathcal{F},\varepsilon) = \widetilde{O}(d)$ and $\log\mathcal{N}(\mathcal{S}\times\mathcal{A},\varepsilon) = \widetilde{O}(d)$. For quadratic functions, $\log\mathcal{N}(\mathcal{F},\varepsilon) = \widetilde{O}(d^2)$ and $\log\mathcal{N}(\mathcal{S}\times\mathcal{A},\varepsilon) = \widetilde{O}(d)$.

4.3 Algorithm

Overview. The algorithm is formally presented in Algorithm 7. At the beginning of each episode $k \in [K]$, we maintain a replay buffer $\{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{(h,\tau)\in[H]\times[k-1]}$ which contains all existing samples. We set $Q_{H+1}^k = 0$, and calculate $Q_H^k, Q_{H-1}^k, \ldots, Q_1^k$ iteratively as follows. For each $h = H, H-1, \ldots, 1$,

$$f_h^k(\cdot,\cdot) \leftarrow \operatorname{argmin}_{f\in\mathcal{F}} \sum_{\tau=1}^{k-1}\sum_{h'=1}^{H} \left(f(s_{h'}^\tau, a_{h'}^\tau) - \left(r_{h'}^\tau + \max_{a\in\mathcal{A}} Q_{h+1}^k(s_{h'+1}^\tau, a) \right) \right)^2 \tag{4.2}$$

and

$$Q_h^k(\cdot,\cdot) = \min\left\{ f_h^k(\cdot,\cdot) + b_h^k(\cdot,\cdot), H \right\}.$$

Here, $b_h^k(\cdot,\cdot)$ is a bonus function to be defined shortly.

Stable Upper-Confidence Bonus Function. With more collected data, the least squares predictor is expected to return a better approximate the true Q-function. To encourage exploration, we carefully design a bonus function $b_h^k(\cdot,\cdot)$ which guarantees that, with high probability, $Q_{h+1}^k(s,a)$ is an overestimate of the one-step backup. The bonus function $b_h^k(\cdot,\cdot)$ is guaranteed to tightly characterize the estimation error of the one-step backup

$$R(\cdot,\cdot) + \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(s'),$$

where $V_{h+1}^k(\cdot) = \max_{a\in\mathcal{A}} Q_{h+1}^k(\cdot,a)$ is the value function of the next step.

Algorithm 7 $\mathcal{F}$-LSVI(δ)

1: **Input**: failure probability $\delta \in (0,1)$ and number of episodes K
2: **for** episode $k = 1, 2, \ldots, K$ **do**
3: Receive initial state $s_1^k \sim \mu$
4: $Q_{H+1}^k(\cdot,\cdot) \leftarrow 0$ and $V_{H+1}^k(\cdot) \leftarrow 0$
5: $\mathcal{Z}^k \leftarrow \{(s_{h'}^\tau, a_{h'}^\tau)\}_{(\tau,h')\in[k-1]\times[H]}$
6: **for** $h = H, \ldots, 1$ **do**
7: $\mathcal{D}_h^k \leftarrow \left\{\left(s_{h'}^\tau, a_{h'}^\tau, r_{h'}^\tau + V_{h+1}^k(s_{h'+1}^\tau, a)\right)\right\}_{(\tau,h')\in[k-1]\times[H]}$
8: $f_h^k \leftarrow \operatorname{argmin}_{f\in\mathcal{F}} \|f\|_{\mathcal{D}_h^k}^2$
9: $b_h^k(\cdot,\cdot) \leftarrow$ Bonus($\mathcal{F}$, f_h^k, $\mathcal{Z}^k$, δ) (Algorithm 9)
10: $Q_h^k(\cdot,\cdot) \leftarrow \min\{f_h^k(\cdot,\cdot) + b_h^k(\cdot,\cdot), H\}$ and $V_h^k(\cdot) = \max_{a\in\mathcal{A}} Q_h^k(\cdot,a)$
11: $\pi_h^k(\cdot) \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} Q_h^k(\cdot,a)$
12: **for** $h = 1, 2, \ldots, H$ **do**
13: Take action $a_h^k \leftarrow \pi_h^k(s_h^k)$ and observe $s_{h+1}^k \sim P(\cdot \mid s_h^k, a_h^k)$ and $r_h^k = R(s_h^k, a_h^k)$

4.3.1 Stable UCB via Importance Sampling

In this section, we formally define the bonus function $b_h^k(\cdot,\cdot)$ used in Algorithm 7. The bonus function is designed to estimate the confidence interval of our estimate of the Q-function. In our algorithm, we define the bonus function to be the width function $b_h^k(\cdot,\cdot) = w(\mathcal{F}_h^k, \cdot, \cdot)$ where the confidence region $\mathcal{F}_h^k$ is defined so that $R(\cdot,\cdot) + \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(s') \in \mathcal{F}_h^k$ with high probability. By definition of the width function, $b_h^k(\cdot,\cdot)$ gives an upper bound on the confidence interval of the estimate of the Q-function, since the width function *maximizes* the difference between all pairs of Q-functions that lie in the confidence region.

To define the confidence region $\mathcal{F}_h^k$, a natural definition would be

$$\mathcal{F}_h^k = \left\{f \in \mathcal{F} \mid \|f - f_h^k\|_{\mathcal{Z}^k}^2 \le \beta\right\}$$

where β is defined so that $R(\cdot,\cdot) + \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(s') \in \mathcal{F}_h^k$ with high probability, and recall that $\mathcal{Z}^k = \{(s_{h'}^\tau, a_{h'}^\tau)\}_{(\tau,h')\in[k-1]\times[H]}$ is the set of state-action pairs defined in Line 5. However, as one can observe, the complexity of such a bonus function could be extremely high as it is defined by a dataset $\mathcal{Z}^k$ whose size can be as large as $T = KH$. A high-complexity bonus function could potentially introduce *instability* issues in the algorithm. Technically, we require a stable bonus function to allow for highly concentrated estimate of the one-step backup so that the confidence region $\mathcal{F}_h^k$ is accurate. Our strategy to "stabilize" the bonus function is to reduce the size of the dataset by importance sampling, so that only important state-action pairs are kept.

Sensitivity Sampling. Here we present a framework to subsample a dataset, so that the confidence region is approximately preserved while the size of the dataset is reduced. Our framework is built upon the *sensitivity sampling* technique introduced in a different context [28, 29, 49].

Definition 4.3.1. For a given set of state-action pairs $\mathcal{Z} \subseteq \mathcal{S}\times\mathcal{A}$ and a function class $\mathcal{F}$, for

Algorithm 8 `Sensitivity-Sampling`($\mathcal{F}$, $\mathcal{Z}$, λ, ε, δ)

1: **Input**: function class $\mathcal{F}$, set of state-action pairs $\mathcal{Z} \subseteq \mathcal{S} \times \mathcal{A}$, accuracy parameters $\lambda, \varepsilon > 0$ and failure probability $\delta \in (0, 1)$
2: Initialize $\mathcal{Z}' \leftarrow \{\}$
3: For each $z \in \mathcal{Z}$, let p_z to be smallest real number such that $1/p_z$ is an integer and

$$p_z \geq \min\{1, \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \cdot 72 \ln(4\mathcal{N}(\mathcal{F}, \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta)/\varepsilon^2\} \tag{4.3}$$

4: For each $z \in \mathcal{Z}$, independently add $1/p_z$ copies of z into $\mathcal{Z}'$ with probability p_z
5: **return** $\mathcal{Z}'$

each $z \in \mathcal{Z}$, define the λ-sensitivity of (s, a) with respect to $\mathcal{Z}$ and $\mathcal{F}$ to be

$$\mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(s, a) = \max_{\substack{f,f' \in F \\ \|f-f'\|_{\mathcal{Z}}^2 \geq \lambda}} \frac{(f(s, a) - f'(s, a))^2}{\|f - f'\|_{\mathcal{Z}}^2}.$$

Sensitivity measures the importance of each data point z in $\mathcal{Z}$ by considering the pair of functions $f, f' \in \mathcal{F}$ such that z contributes the most to $\|f - f'\|_{\mathcal{Z}}^2$. In Algorithm 8, we define a procedure to sample each state-action pair with sampling probability proportional to the sensitivity. In the analysis, we show that after applying Algorithm 8 on the input dataset $\mathcal{Z}$, the confidence region $\left\{f \in \mathcal{F} \mid \|f - f_h^k\|_{\mathcal{Z}}^2 \leq \beta\right\}$ is approximately preserved, while the size of the subsampled dataset is upper bounded by the eluder dimension of $\mathcal{F}$ times the log-covering number of $\mathcal{F}$.

The Stable Bonus Function. With the above sampling procedure, we are now ready to obtain a stable bonus function. In Algorithm 9, we first subsample the given dataset $\mathcal{Z}$ and then round the reference function $\bar{f}$ and all data points in the subsampled dataset $\overline{\mathcal{Z}}$ to their nearest neighbors in a $1/(8\sqrt{4T/\delta})$-cover. We discard the subsampled dataset if its size is too large, and then define the confidence region using the new dataset and the rounded reference function.

4.4 Theoretical Guarantee

In this section we provide the theoretical guarantee of Algorithm 7, which is stated in Theorem 8.3.1.

Theorem 4.4.1. *Under Assumption 4.2.1, after interacting with the environment for $T = KH$ steps, with probability $1 - \delta$, Algorithm 7 achieves a a regret bound of*

$$\mathrm{Reg}(K) \leq \sqrt{\iota \cdot H^2 \cdot T},$$

where

$$\iota \leq C \cdot \log^2(T/\delta) \cdot \dim_E^2\left(\mathcal{F}, \delta/T^3\right) \cdot \ln\left(\mathcal{N}\left(\mathcal{F}, \delta/T^2\right)/\delta\right) \cdot \log\left(\mathcal{N}\left(\mathcal{S} \times \mathcal{A}, \delta/T\right) \cdot T/\delta\right)$$

for some constant $C > 0$.

Algorithm 9 `Bonus`$(\mathcal{F}, \bar{f}, \mathcal{Z}, \delta)$

Input: function class $\mathcal{F}$, reference function $\bar{f} \in \mathcal{F}$, state-action pairs $\mathcal{Z} \subseteq \mathcal{S} \times \mathcal{A}$ and failure probability $\delta \in (0,1)$

$\overline{\mathcal{Z}} \leftarrow$ `Sensitivity-Sampling`$\big(\mathcal{F}, \mathcal{Z}, \delta/(16T), 1/2, \delta\big)$ ▷ Subsample the dataset

$\overline{\mathcal{Z}} \leftarrow \{\}$ if $|\overline{\mathcal{Z}}| \geq 4T/\delta$ or the number of distinct elements in $\overline{\mathcal{Z}}$ exceeds

$$6912 \dim_E(\mathcal{F}, \delta/(16T^2)) \log(64H^2T^2/\delta) \ln T \ln(4\mathcal{N}(\mathcal{F}, \delta/(566T))/\delta)$$

Let $\widehat{f} \in \mathcal{C}(\mathcal{F}, 1/(8\sqrt{4T/\delta}))$ be such that $\|\bar{f} - \widehat{f}\|_\infty \leq 1/(8\sqrt{4T/\delta})$ ▷ Round $\bar{f}$

$\widehat{\mathcal{Z}} \leftarrow \{\}$

for $z \in \overline{\mathcal{Z}}$ **do** ▷ Round state-action pairs

  Let $\widehat{z} \in \mathcal{C}(\mathcal{S} \times \mathcal{A}, 1/(8\sqrt{4T/\delta}))$ be such that $\sup_{f \in \mathcal{F}} |f(z) - f(\widehat{z})| \leq 1/(8\sqrt{4T/\delta})$

  $\widehat{\mathcal{Z}} \leftarrow \widehat{\mathcal{Z}} \cup \{\widehat{z}\}$

return $\widehat{w}(\cdot,\cdot) := w(\widehat{\mathcal{F}}, \cdot, \cdot)$, where $\widehat{\mathcal{F}} = \left\{ f \in \mathcal{F} \mid \|f - \widehat{f}\|_{\widehat{\mathcal{Z}}}^2 \leq 3\beta(\mathcal{F}, \delta) + 2 \right\}$ and

$$\beta(\mathcal{F}, \delta) = c' H^2 \cdot \log^2(T/\delta) \cdot \dim_E(\mathcal{F}, \delta/T^3) \cdot \ln(\mathcal{N}(\mathcal{F}, \delta/T^2)/\delta) \cdot \log\left(\mathcal{N}(\mathcal{S} \times \mathcal{A}, \delta/T)\right) \cdot T/\delta) \tag{4.4}$$

for some absolute constants $c' > 0$.

Remark 4.4.1. For the tabular setting, we may set $\mathcal{F}$ to be the entire function space of $\mathcal{S} \times \mathcal{A} \to [0, H+1]$. Recall that when $\mathcal{S}$ and $\mathcal{A}$ are finite, for any $\varepsilon > 0$, $\dim_E(\mathcal{F}, \varepsilon) \leq |\mathcal{S}||\mathcal{A}|$, $\log(\mathcal{N}(\mathcal{F}, \varepsilon)) = \widetilde{O}(|\mathcal{S}||\mathcal{A}|)$ and $\log(\mathcal{N}(\mathcal{S} \times \mathcal{A}, \varepsilon)) = O(\log(|\mathcal{S}||\mathcal{A}|))$, and thus the regret bound in Theorem 8.3.1 is $\widetilde{O}(\sqrt{|\mathcal{S}|^3|\mathcal{A}|^3H^2T})$ which is worse than the near-optimal bound in [11]. However, when applied to the tabular setting, our algorithm is similar to the algorithm in [11]. By a more refined analysis specialized to the tabular setting, the regret bound of our algorithm can be improved using techniques in [11]. We would like to stress that our algorithm and analysis tackle a much more general setting and recovering the optimal regret bound for the tabular setting is not the focus of this chapter.

Remark 4.4.2. When $\mathcal{F}$ is the class of d-dimensional linear functions, we have $\dim_E(\mathcal{F}, \varepsilon) = \widetilde{O}(d)$, $\log(\mathcal{N}(\mathcal{F}, \varepsilon)) = \widetilde{O}(d)$ and $\log(\mathcal{N}(\mathcal{S} \times \mathcal{A}, \varepsilon)) = \widetilde{O}(d)$ and thus the regret bound in Theorem 8.3.1 is $\widetilde{O}(\sqrt{d^4H^2T})$, which is worse by a $\widetilde{O}(\sqrt{d})$ factor when compared to the bound in [40, 96], and is worse by a $\widetilde{O}(d)$ factor when compared to the bound in [108]. Note that for our algorithm, a regret bound of $\widetilde{O}(\sqrt{d^3H^2T})$ is achievable using a more refined analysis (see Remark 4.4.3). Moreover, unlike our algorithm, the algorithm in [108] requires solving the Planning Optimization Program and is thus computationally intractable. Finally, we would like to stress that our algorithm and analysis tackle the case that $\mathcal{F}$ is a general function class which contains the linear case studied in [40, 96, 108] as a special case.

Here we provide an overview of the proof to highlight the technical novelties in the analysis.

The Stable Bonus Function. Similar to the analysis in [40, 96], to account for the dependency structure in the data sequence, we need to bound the complexity of the bonus function $b_h^k(\cdot,\cdot)$.

When $\mathcal{F}$ is the class of d-dimensional linear functions (as in [40, 96]), $b(\cdot,\cdot) = \|\phi(\cdot,\cdot)\|_{\Lambda^{-1}}$ for a covariance matrix $\Lambda \in \mathbb{R}^{d\times d}$, whose complexity is upper bounded by d^2 which is the number of entries in the covariance matrix Λ. However, such simple complexity upper bound is no longer available for the class of general functions considered in this chapter. Instead, we bound the complexity of the bonus function by relying on the fact that the subsampled dataset has bounded size. Scrutinizing the sampling algorithm (Algorithm 8), it can be seen that the size of the subsampled dataset is upper bounded by the sum of the sensitivity of the data points in the given dataset times the log-convering number of the function class $\mathcal{F}$. To upper bound the sum of the sensitivity of the data points in the given dataset, we rely on a novel combinatorial argument which establishes a surprising connection between the sum of the sensitivity and the eluder dimension of the function class $\mathcal{F}$. We show that the sum of the sensitivity of data points is upper bounded by the eluder dimension of the dataset up to logarithm factors. Hence, the complexity of the subsampled dataset, and therefore, the complexity of the bonus function, is upper bound by the log-covering number of $\mathcal{S}\times\mathcal{A}$ (the complexity of each state-action pair) times the product of the eluder dimension of the function class and the log-covering number of the function class (the number of data points in the subsampled dataset).

In order to show that the confidence region is approximately preserved when using the subsampled dataset $\mathcal{Z}'$, we show that for any $f, f' \in \mathcal{F}$, $\|f - f'\|^2_{\mathcal{Z}'}$ is a good approximation to $\|f - f'\|^2_{\mathcal{Z}}$. To show this, we apply a union bound over all pairs of functions on the cover of $\mathcal{F}$ which allows us to consider fixed $f, f' \in \mathcal{F}$. For fixed $f, f' \in \mathcal{F}$, note that $\|f - f'\|^2_{\mathcal{Z}'}$ is an unbiased estimate of $\|f - f'\|^2_{\mathcal{Z}}$, and importance sampling proportinal to the sensitivity implies an upper bound on the variance of the estimator which allows us to apply concentration bounds to prove the desired result. We note that the sensitivity sampling framework used here is very crucial to the theoreical guarantee of the algorithm. If one replaces sensitivity sampling with more naïve sampling approaches (e.g. uniform sampling), then the required sampling size would be much larger, which does not give any meaningful reduction on the size of the dataset and also leads to a high complexity bonus function.

Remark 4.4.3. When $\mathcal{F}$ is the class of d-dimensional linear functions, our upper bound on the size of the subsampled dataset is $\widetilde{O}(d^2)$. However, in this case, our sampling algorithm (Algorithm 8) is equivalent to the leverage score sampling [21] and therefore the sample complexity can be further improved to $\widetilde{O}(d)$ using a more refined analysis [80]. Therefore, our regret bound can be improved to $\widetilde{O}(\sqrt{d^3H^2T})$, which matches the bounds in [40, 96]. However, the $\widetilde{O}(d)$ sample bound is specialized to the linear case and heavily relies on the matrix Chernoff bound which is unavailable for the class of general functions considered in this chapter. This also explains why our regret bound in Theorem 8.3.1, when applied to the linear case, is larger by a $\sqrt{d}$ factor when compared to those in [40, 96]. We leave it as an open question to obtain more refined bound on the size of the subsampled dataset and improve the overall regret bound of our algorithm.

The Confidence Region. Our algorithm applies the principle of optimism in the face of uncertainty (OFU) to balance exploration and exploitation. Note that V^k_{h+1} is the value function estimated at step $h+1$. In our analysis, we require the Q-function Q^k_h estimated at level h to

satisfy

$$Q_h^k(\cdot,\cdot) \geq R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s'|\cdot,\cdot) V_{h+1}^k(s')$$

with high probability. To achieve this, we optimize the least squares objective to find a solution $f_h^k \in \mathcal{F}$ using collected data. We then show that f_h^k is close to $R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s'|\cdot,\cdot) V_{h+1}^k(s')$. This would follow from standard analysis if the collected samples were independent of V_{h+1}^k. However, V_{h+1}^k is calculated using the collected samples and thus they are subtly dependent on each other. To tackle this issue, we notice that V_{h+1}^k is computed by using f_{h+1}^k and the bonus function b_{h+1}^k, and both f_{h+1}^k and the bonus function b_{h+1}^k have bounded complexity, thanks to the design of bonus function. Hence, we can construct a $1/T$-cover to approximate V_{h+1}^k. By doing so, we can now bound the fitting error of f_h^k by replacing V_{h+1}^k with its closest neighbor in the $1/T$-cover which is independent of the dataset. By a union bound over all functions in the $1/T$-cover, it follows that with high probability,

$$R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s'|\cdot,\cdot) V_{h+1}^k(s') \in \left\{ f \in \mathcal{F} \mid \|f - f_h^k\|_{\mathcal{Z}^k}^2 \leq \beta \right\}$$

for some β that depends only on the complexity of the bonus function and the function class $\mathcal{F}$.

Regret Decomposition and the Eluder Dimension. By standard regret decomposition for optimistic algorithms, the total regret is upper bounded by the summation of the bonus function $\sum_{k=1}^K \sum_{h=1}^H b_h^k\left(s_h^k, a_h^k\right)$. To bound the summation of the bonus function, we use an argument similar to that in [73], which shows that the summation of the bonus function can be upper bounded in terms of the eluder dimension of the function class $\mathcal{F}$, if the confidence region is defined using the original dataset. In the formal analysis, we adapt the argument in [73] to show that even if the confidence region is defined using the subsampled dataset, the summation of the bonus function can be bounded in a similar manner.

4.4.1 Analysis of the Stable Bonus Function

Our first lemma gives an upper bound on the sum of the sensitivity in terms of the eluder dimension of the function class $\mathcal{F}$.

Lemma 4.4.2. *For a given set of state-action pairs $\mathcal{Z}$,*

$$\sum_{z \in \mathcal{Z}} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \leq 4\dim_E(\mathcal{F}, \lambda/|\mathcal{Z}|) \log((H+1)^2|\mathcal{Z}|/\lambda) \ln|\mathcal{Z}|.$$

Proof. For each $z \in \mathcal{Z}$, let $f, f' \in F$ be an arbitrary pair of functions such that $\|f - f'\|_{\mathcal{Z}}^2 \geq \lambda$ and

$$\frac{(f(z) - f'(z))^2}{\|f - f'\|_{\mathcal{Z}}^2}$$

is maximized, and we define $L(z) = (f(z) - f'(z))^2$ for such f and f'. Note that $0 \leq L(z) \leq (H+1)^2$. Let $\mathcal{Z} = \bigcup_{\alpha=0}^{\log((H+1)^2|\mathcal{Z}|/\lambda)-1} \mathcal{Z}^\alpha \cup \mathcal{Z}^\infty$ be a dyadic decomposition with respect to $L(\cdot)$, where for each $0 \leq \alpha < \log((H+1)^2|\mathcal{Z}|/\lambda)$, define

$$\mathcal{Z}^\alpha = \{z \in \mathcal{Z} \mid L(z) \in ((H+1)^2 \cdot 2^{-\alpha-1}, (H+1)^2 \cdot 2^{-\alpha}]\}$$

and

$$\mathcal{Z}^\infty = \{z \in \mathcal{Z} \mid L(z) \le \lambda/|\mathcal{Z}|\}.$$

Clearly, for any $z \in \mathcal{Z}^\infty$, $\mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \le 1/|\mathcal{Z}|$ and thus

$$\sum_{z\in\mathcal{Z}^\infty} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \le 1.$$

Now we bound $\sum_{z\in\mathcal{Z}^\alpha} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z)$ for each $0 \le \alpha < \log((H+1)^2|\mathcal{Z}|/\lambda)$ separately. For each α, let

$$N_\alpha = |\mathcal{Z}^\alpha|/\dim_E(\mathcal{F}, (H+1)^2 \cdot 2^{-\alpha-1})$$

and we decompose $\mathcal{Z}^\alpha$ into $N_\alpha+1$ disjoint subsets, i.e., $\mathcal{Z}^\alpha = \bigcup_{j=1}^{N_\alpha+1} \mathcal{Z}_j^\alpha$, by using the following procedure. Let $\mathcal{Z}^\alpha = \{z_1, z_2, \ldots, z_{|\mathcal{Z}^\alpha|}\}$ and we consider each z_i sequentially. Initially $\mathcal{Z}_j^\alpha = \{\}$ for all j. Then, for each z_i, we find the largest $1 \le j \le N_\alpha$ such that z_i is $(H+1)^2 \cdot 2^{-\alpha-1}$-independent of $\mathcal{Z}_j^\alpha$ with respect to $\mathcal{F}$. We set $j = N_\alpha + 1$ if such j does not exist, and use $j(z_i) \in [N_\alpha + 1]$ to denote the choice of j for z_i. By the design of the algorithm, for each z_i, it is clear that z_i is dependent on each of $\mathcal{Z}_1^\alpha, \mathcal{Z}_2^\alpha, \ldots, \mathcal{Z}_{j(z_i)-1}^\alpha$.

Now we show that for each $z_i \in \mathcal{Z}^\alpha$,

$$\mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z_i) \le 2/j(z_i).$$

For any $z_i \in \mathcal{Z}^\alpha$, we use $f, f' \in F$ to denote the pair of functions in $\mathcal{F}$ such that $\|f - f'\|_{\mathcal{Z}}^2 \ge \lambda$ and

$$\frac{(f(z_i) - f'(z_i))^2}{\|f - f'\|_{\mathcal{Z}}^2}$$

is maximized. Since $z_i \in \mathcal{Z}^\alpha$, we must have $(f(z_i) - f'(z_i))^2 > (H+1)^2 \cdot 2^{-\alpha-1}$. Since z_i is dependent on each of $\mathcal{Z}_1^\alpha, \mathcal{Z}_2^\alpha, \ldots, \mathcal{Z}_{j(z_i)-1}^\alpha$, for each $1 \le k < j(z_i)$, we have

$$\|f - f'\|_{\mathcal{Z}_k^\alpha} \ge (H+1)^2 \cdot 2^{-\alpha-1},$$

which implies

$$\begin{aligned}
&\mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z_i) = \frac{(f(z_i) - f'(z_i))^2}{\|f - f'\|_{\mathcal{Z}}^2} \le \frac{(H+1)^2 \cdot 2^{-\alpha}}{\|f - f'\|_{\mathcal{Z}}^2} \\
\le &\frac{(H+1)^2 \cdot 2^{-\alpha}}{\sum_{k=1}^{j(z_i)-1} \|f - f'\|_{\mathcal{Z}_k^\alpha} + (f(z_i) - f'(z_i))^2} \le 2/j(z_i).
\end{aligned}$$

Moreover, by the definition of $(H+1)^2 \cdot 2^{-\alpha-1}$-independence, we have $|\mathcal{Z}_j^\alpha| \le \dim_E(\mathcal{F}, (H+1)^2 \cdot 2^{-\alpha-1})$ for all $1 \le j \le N_\alpha$. Therefore,

$$\begin{aligned}
&\sum_{z\in\mathcal{Z}^\alpha} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \le \sum_{1\le j\le N_\alpha} |\mathcal{Z}_j^\alpha| \cdot 2/j + \sum_{z\in\mathcal{Z}_{N_\alpha+1}^\alpha} 2/N_\alpha \\
\le &2\dim_E(\mathcal{F}, (H+1)^2 \cdot 2^{-\alpha-1}) \ln(N_\alpha) + |\mathcal{Z}^\alpha| \cdot \frac{2\dim_E(\mathcal{F}, (H+1)^2 \cdot 2^{-\alpha-1})}{|\mathcal{Z}^\alpha|} \\
\le &3\dim_E(\mathcal{F}, (H+1)^2 \cdot 2^{-\alpha-1}) \ln(|\mathcal{Z}|).
\end{aligned}$$

By the monotonicity of eluder dimension, it follows that

$$\begin{aligned}
&\sum_{z\in\mathcal{Z}} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \\
\leq &\sum_{\alpha=0}^{\log((H+1)^2|\mathcal{Z}|/\lambda)-1} \sum_{z\in\mathcal{Z}^\alpha} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) + \sum_{z\in\mathcal{Z}^\infty} \mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z) \\
\leq &3\log((H+1)^2|\mathcal{Z}|/\lambda)\dim_E(\mathcal{F},\lambda/|\mathcal{Z}|)\ln(|\mathcal{Z}|)+1 \\
\leq &4\log((H+1)^2|\mathcal{Z}|/\lambda)\dim_E(\mathcal{F},\lambda/|\mathcal{Z}|)\ln(|\mathcal{Z}|).
\end{aligned}$$

□

Using Lemma 4.4.2, we can prove an upper bound on the number of distinct elements in $\mathcal{Z}'$ returned by the sampling algorithm (Algorithm 8).

Lemma 4.4.3. *With probability at least $1-\delta/4$, the number of distinct elements in $\mathcal{Z}'$ returned by Algorithm 8 is at most*

$$1728\dim_E(\mathcal{F},\lambda/|\mathcal{Z}|)\log((H+1)^2|\mathcal{Z}|/\lambda)\ln(|\mathcal{Z}|)\ln(4\mathcal{N}(\mathcal{F},\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta)/\varepsilon^2.$$

Proof. Note that

$$p_z \leq \min\{1, 2\cdot\mathsf{sensitivity}_{\mathcal{Z},\mathcal{F},\lambda}(z)\cdot 72\ln(4\mathcal{N}(\mathcal{F},\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta)/\varepsilon^2\},$$

since for any real number $x<1$, there always exists $\widehat{x}\in[x,2x]$ such that $1/\widehat{x}$ is an integer. Let X_z be a random variable defined as

$$X_z = \begin{cases} 1 & z\in Z' \\ 0 & z\notin Z' \end{cases}.$$

Clearly, the number of distinct elements in $\mathcal{Z}'$ is upper bounded by $\sum_{z\in\mathcal{Z}} X_z$ and $\mathbb{E}[X_z]=p_z$. By Lemma 4.4.2,

$$\sum_{z\in\mathcal{Z}}\mathbb{E}[X_z] \leq 576\dim_E(\mathcal{F},\lambda/|\mathcal{Z}|)\log((H+1)^2|\mathcal{Z}|/\lambda)\ln(|\mathcal{Z}|)\ln(4\mathcal{N}(\mathcal{F},\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta)/\varepsilon^2.$$

By Chernoff bound, with probability at least $1-\delta/4$, we have

$$\sum_{z\in\mathcal{Z}} X_z \geq 1728\dim_E(\mathcal{F},\lambda/|\mathcal{Z}|)\log((H+1)^2|\mathcal{Z}|/\lambda)\ln(|\mathcal{Z}|)\ln(4\mathcal{N}(\mathcal{F},\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta)/\varepsilon^2.$$

□

Our second lemma upper bounds the number of elements in $\mathcal{Z}'$ returned by Algorithm 8.

Lemma 4.4.4. *With probability at least $1-\delta/4$, $|\mathcal{Z}'|\leq 4|\mathcal{Z}|/\delta$.*

Proof. Let X_z be the random variable which is defined as

$$X_z = \begin{cases} 1/p_z & z \text{ is added into } \mathcal{Z}' \\ 0 & \text{otherwise} \end{cases}.$$

Note that $|\mathcal{Z}'| = \sum_{z\in\mathcal{Z}} X_z$ and $\mathbb{E}[X_z] = 1$. By Markov inequality, with probability $1 - \delta/4$, $|\mathcal{Z}'| \le 4|\mathcal{Z}|/\delta$. □

Our third lemma shows that for the given set of state-action pairs $\mathcal{Z}$ and function class $\mathcal{F}$, Algorithm 8 returns a set of state-action pairs $\mathcal{Z}'$ so that $\|f - f'\|_{\mathcal{Z}}^2$ is approximately preserved for all $f, f' \in \mathcal{F}$.

Lemma 4.4.5. *With probability at least $1 - \delta/2$, for any $f, f' \in \mathcal{F}$,*

$$(1-\varepsilon)\|f - f'\|_{\mathcal{Z}}^2 - 2\lambda \le \|f - f'\|_{\mathcal{Z}'}^2 \le (1+\varepsilon)\|f - f'\|_{\mathcal{Z}}^2 + 8|\mathcal{Z}|\lambda/\delta.$$

Proof. In our proof, we separately consider two cases: $\|f - f'\|_{\mathcal{Z}}^2 < 2\lambda$ and $\|f - f'\|_{\mathcal{Z}}^2 \ge 2\lambda$.

Case I: $\|f - f'\|_{\mathcal{Z}}^2 < 2\lambda$. Consider $f, f' \in \mathcal{F}$ with $\|f - f'\|_{\mathcal{Z}}^2 < 2\lambda$. Conditioned on the event defined in Lemma 4.4.4 which holds with probability at least $1 - \delta/4$, we have $\|f - f'\|_{\mathcal{Z}'}^2 \le |\mathcal{Z}'| \cdot \|f - f'\|_{\mathcal{Z}}^2 \le 8|\mathcal{Z}|\lambda/\delta$. Moreover, we always have $\|f - f'\|_{\mathcal{Z}'} \ge 0$. In summary, we have

$$\|f - f'\|_{\mathcal{Z}}^2 - 2\lambda \le \|f - f'\|_{\mathcal{Z}'}^2 \le \|f - f'\|_{\mathcal{Z}}^2 + 8|\mathcal{Z}|\lambda/\delta.$$

Case II: $\|f - f'\|_{\mathcal{Z}}^2 \ge 2\lambda$. We first show that for any fixed $f, f' \in \mathcal{F}$ with $\|f - f'\|_{\mathcal{Z}}^2 \ge \lambda$, with probability at least $1 - \delta/(4\mathcal{N}(\mathcal{F}, \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)}))$, we have

$$(1-\varepsilon/4)\|f - f'\|_{\mathcal{Z}}^2 \le \|f - f'\|_{\mathcal{Z}'}^2 \le (1+\varepsilon/4)\|f - f'\|_{\mathcal{Z}}^2.$$

To prove this, for each $z \in \mathcal{Z}$, define

$$X_z = \begin{cases} \frac{1}{p_z}(f(z) - f'(z))^2 & z \text{ is added into } \mathcal{Z}' \text{ for } 1/p_z \text{ times} \\ 0 & \text{otherwise} \end{cases}.$$

Clearly, $\|f - f'\|_{\mathcal{Z}'} = \sum_{z\in\mathcal{Z}} X_z$ and $\mathbb{E}[X_z] = (f(z) - f'(z))^2$. Moreover, since $\|f - f'\|_{\mathcal{Z}}^2 \ge \lambda$, by (4.3) and Definition 4.3.1, we have

$$\max_{z\in\mathcal{Z}} X_z \le \|f - f'\|_{\mathcal{Z}}^2 \cdot \varepsilon^2/(72\ln(4\mathcal{N}(\mathcal{F}, \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta).$$

Moreover, $\mathbb{E}[X_z^2] \le (f(z) - f'(z))^4/p_z$. Therefore, by Hölder's inequality,

$$\begin{aligned}\sum_{z\in\mathcal{Z}} \mathrm{Var}[X_z] \le \sum_{z\in\mathcal{Z}} \mathbb{E}[X_z^2] \le \sum_{z\in\mathcal{Z}} (f(z) - f'(z))^2 \cdot \max_{z\in\mathcal{Z}} (f(z) - f'(z))^2/p_z \\ \le \|f - f'\|_{\mathcal{Z}}^4 \cdot \varepsilon^2/(72\ln(4\mathcal{N}(\mathcal{F}, \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta).\end{aligned}$$

Therefore, by Bernstein inequality,

$$\begin{aligned}
&\Pr\left[|\|f-f'\|_{\mathcal{Z}}^2-\|f-f'\|_{\mathcal{Z}'}^2|\geq\varepsilon/4\cdot\|f-f'\|_{\mathcal{Z}}^2\right]\\
=&\Pr\left[\left|\sum_{z\in\mathcal{Z}}\mathbb{E}[X_z]-\sum_{z\in\mathcal{Z}}X_z\right|\geq\varepsilon/4\cdot\|f-f'\|_{\mathcal{Z}}^2\right]\\
\leq&2\exp\left(-\frac{\varepsilon^2/16\cdot\|f-f'\|_{\mathcal{Z}}^4}{2\sum_{z\in\mathcal{Z}}\mathrm{Var}[X_z]+2\max_{z\in\mathcal{Z}}X_z\cdot\varepsilon/4\cdot\|f-f'\|_{\mathcal{Z}}^2/3}\right)\\
\leq&(\delta/4)/\left(\mathcal{N}(\mathcal{F},\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})\right)^2.
\end{aligned}$$

By union bound, the above inequality implies that with probability at least $1-\delta/4$, for any $(f,f')\in\mathcal{C}(F,\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})\times\mathcal{C}(F,\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})$ with $\|f-f'\|_{\mathcal{Z}}^2\geq\lambda$,

$$(1-\varepsilon/4)\|f-f'\|_{\mathcal{Z}}^2\leq\|f-f'\|_{\mathcal{Z}'}^2\leq(1+\varepsilon/4)\|f-f'\|_{\mathcal{Z}'}^2.$$

Now we condition on the event defined above and the event defined in Lemma 4.4.4. Consider $f,f'\in\mathcal{F}$ with $\|f-f'\|_{\mathcal{Z}}^2\geq2\lambda$. Recall that there exists

$$(\widehat{f},\widehat{f'})\in\mathcal{C}(F,\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})\times\mathcal{C}(F,\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)})$$

such that $\|f-\widehat{f}\|_\infty\leq\sqrt{\lambda/(25|\mathcal{Z}|)}$ and $\|f'-\widehat{f'}\|_\infty\leq\sqrt{\lambda/(25|\mathcal{Z}|)}$. Therefore,

$$\begin{aligned}
&\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}}^2=\sum_{z\in\mathcal{Z}}(\widehat{f}(z)-\widehat{f'}(z))^2\\
=&\sum_{z\in\mathcal{Z}}(f(z)-f'(z)+(\widehat{f}(z)-f(z))+(f'(z)-\widehat{f'}(z)))^2\\
\geq&\left(\|f-f'\|_{\mathcal{Z}}-\|\widehat{f}-f\|_{\mathcal{Z}}-\|f'-\widehat{f'}\|_{\mathcal{Z}}\right)^2\\
\geq&\left(\sqrt{2\lambda}-2\sqrt{\lambda/25}\right)^2\geq\lambda.
\end{aligned}$$

Therefore, conditioned on the event defined above, we have

$$(1-\varepsilon/4)\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}}^2\leq\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}'}^2\leq(1+\varepsilon/4)\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}'}^2.$$

Conditioned on the event defined in Lemma 4.4.4 which holds with probability at least $1-\delta/4$, we have

$$\begin{aligned}
&\|f-f'\|_{\mathcal{Z}'}^2\leq\left(\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}'}+\|f-\widehat{f}\|_{\mathcal{Z}'}+\|f'-\widehat{f'}\|_{\mathcal{Z}'}\right)^2\\
\leq&\left(\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}'}+2\sqrt{|\mathcal{Z}'|}\cdot\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)}\right)^2\\
\leq&\left((1+\varepsilon/6)\|\widehat{f}-\widehat{f'}\|_{\mathcal{Z}}+2\sqrt{|\mathcal{Z}'|}\cdot\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)}\right)^2\\
\leq&\left((1+\varepsilon/6)\|f-f\|_{\mathcal{Z}}+2\sqrt{|\mathcal{Z}'|}\cdot\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)}+4\sqrt{|\mathcal{Z}|}\cdot\varepsilon/72\cdot\sqrt{\lambda\delta/(|\mathcal{Z}|)}\right)^2\\
\leq&(1+\varepsilon)\|f-f\|_{\mathcal{Z}}^2,
\end{aligned}$$

where the last inequality holds since $\|f - f\|_{\mathcal{Z}} \geq \sqrt{\lambda}$.

Similarly,

$$\begin{aligned}
&\|f - f'\|_{\mathcal{Z}'}^2 \geq \left(\|\widehat{f} - \widehat{f}'\|_{\mathcal{Z}'} - \|f - \widehat{f}\|_{\mathcal{Z}'} - \|f' - \widehat{f}'\|_{\mathcal{Z}'}\right)^2 \\
\geq &\left(\|\widehat{f} - \widehat{f}'\|_{\mathcal{Z}'} - 2\sqrt{|\mathcal{Z}'|} \cdot \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)}\right)^2 \\
\geq &\left((1-\varepsilon/6)\|\widehat{f} - \widehat{f}'\|_{\mathcal{Z}} - 2\sqrt{|\mathcal{Z}'|} \cdot \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)}\right)^2 \\
\geq &\left((1-\varepsilon/6)\|f - f\|_{\mathcal{Z}} - 2\sqrt{|\mathcal{Z}'|} \cdot \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)} - 2\sqrt{|\mathcal{Z}|} \cdot \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)}\right)^2 \\
\geq &(1-\varepsilon)\|f - f\|_{\mathcal{Z}}^2.
\end{aligned}$$

□

Combining Lemma 4.4.3, Lemma 4.4.4 and Lemma 4.4.5 with a union bound, we have the following proposition.

Proposition 4.4.6. *With probability at least $1-\delta$, the size of $\mathcal{Z}'$ returned by Algorithm 8 satisfies $|\mathcal{Z}'| \leq 4|\mathcal{Z}|/\delta$, the number of distinct elements in $\mathcal{Z}$ is at most*

$$1728\dim_E(\mathcal{F}, \lambda/|\mathcal{Z}|)\log((H+1)^2|\mathcal{Z}|/\lambda)\ln(|\mathcal{Z}|)\ln(4\mathcal{N}(\mathcal{F}, \varepsilon/72 \cdot \sqrt{\lambda\delta/(|\mathcal{Z}|)})/\delta)/\varepsilon^2,$$

and for any $f, f' \in \mathcal{F}$,

$$(1-\varepsilon)\|f - f'\|_{\mathcal{Z}}^2 - 2\lambda \leq \|f - f'\|_{\mathcal{Z}'}^2 \leq (1+\varepsilon)\|f - f'\|_{\mathcal{Z}}^2 + 8|\mathcal{Z}|\lambda/\delta.$$

Proposition 4.4.7. *For Algorithm 9, suppose $|\mathcal{Z}| \leq KH = T$, the following holds.*

1. *With probability at least $1 - \delta/(16T)$,*

$$w(\underline{\mathcal{F}}, s, a) \leq \widehat{w}(s, a) \leq w(\overline{\mathcal{F}}, s, a)$$

   where $\underline{\mathcal{F}} = \{f \in \mathcal{F} \mid \|f - \bar{f}\|_{\mathcal{Z}}^2 \leq \beta(\mathcal{F}, \delta)\}$, and $\overline{\mathcal{F}} = \{f \in \mathcal{F} \mid \|f - \bar{f}\|_{\mathcal{Z}}^2 \leq 9\beta(\mathcal{F}, \delta) + 12\}$.
2. *$\widehat{w}(\cdot, \cdot) \in \mathcal{W}$ for a function set $\mathcal{W}$ with*

$$\begin{aligned}
\log|\mathcal{W}| \leq\ & 6912\dim_E(\mathcal{F}, \delta/(16T^2))\log(16(H+1)^2T^2/\delta)\ln T\ln(4\mathcal{N}(\mathcal{F}, \delta/(566T))/\delta) \\
&\cdot \log\left(\mathcal{N}(\mathcal{S}\times\mathcal{A}, 1/(8\sqrt{4T/\delta})) \cdot 4T/\delta\right) + \log(\mathcal{N}(\mathcal{F}, 1/(8\sqrt{4T/\delta}))) \\
\leq\ & C \cdot \dim_E(\mathcal{F}, \delta/T^3) \cdot \log(H^2T^2/\delta) \cdot \ln T \cdot \ln(\mathcal{N}(\mathcal{F}, \delta/T^2)/\delta) \\
&\cdot \log\left(\mathcal{N}(\mathcal{S}\times\mathcal{A}, \delta/T)) \cdot T/\delta\right),
\end{aligned}$$

   for some absolute constant $C > 0$ if T is sufficiently large.

Proof. For the first part, conditioned on the event defined in Proposition 4.4.6, for any $f \in \mathcal{F}$, we have

$$\|f - \bar{f}\|_{\mathcal{Z}}^2/2 - 1/2 \leq \|f - \bar{f}\|_{\mathcal{Z}}^2 \leq 3\|f - \bar{f}\|_{\mathcal{Z}}^2/2 + 1/2.$$

Therefore, we have

$$\begin{aligned}&\|f-\widehat{f}\|_{\widehat{\mathcal{Z}}}^2 \le (\|f-\widehat{f}\|_{\overline{\mathcal{Z}}}+\sqrt{4T/\delta}/(8\sqrt{4T/\delta}))^2\\ \le&(\|f-\bar{f}\|_{\overline{\mathcal{Z}}}+\sqrt{4T/\delta}/(8\sqrt{4T/\delta})+\sqrt{4T/\delta}/(8\sqrt{4T/\delta}))^2\\ \le&2\|f-\bar{f}\|_{\overline{\mathcal{Z}}}^2+2(\sqrt{4T/\delta}/(8\sqrt{4T/\delta})+\sqrt{4T/\delta}/(8\sqrt{4T/\delta}))^2 \le 3\|f-\bar{f}\|_{\mathcal{Z}}^2+2\end{aligned}$$

and

$$\begin{aligned}&\|f-\widehat{f}\|_{\widehat{\mathcal{Z}}}^2 \ge (\|f-\widehat{f}\|_{\overline{\mathcal{Z}}}-\sqrt{4T/\delta}/(8\sqrt{4T/\delta}))^2\\ \ge&(\|f-\bar{f}\|_{\overline{\mathcal{Z}}}-\sqrt{4T/\delta}/(8\sqrt{4T/\delta})-\sqrt{4T/\delta}/(8\sqrt{4T/\delta}))^2\\ \ge&\|f-\bar{f}\|_{\overline{\mathcal{Z}}}^2/2-(\sqrt{4T/\delta}/(8\sqrt{4T/\delta})+\sqrt{4T/\delta}/(8\sqrt{4T/\delta}))^2 \ge \|f-\bar{f}\|_{\mathcal{Z}}^2/3-2.\end{aligned}$$

Therefore, for any $f \in \underline{\mathcal{F}}$, we have $\|f-\bar{f}\|_{\mathcal{Z}}^2 \le \beta(\mathcal{F},\delta)$, which implies $\|f-\widehat{f}\|_{\widehat{\mathcal{Z}}}^2 \le 3\beta(\mathcal{F},\delta)+2$ and thus $f \in \widehat{\mathcal{F}}$. Moreover, for any $f \in \widehat{\mathcal{F}}$, we have $\|f-\widehat{f}\|_{\widehat{\mathcal{Z}}}^2 \le 3\beta(\mathcal{F},\delta)+2$, which implies $\|f-\bar{f}\|_{\mathcal{Z}}^2 \le 9\beta(\mathcal{F},\delta)+12$.

For the second part, note that $\widehat{w}(\cdot,\cdot)$ is uniquely defined by $\widehat{\mathcal{F}}$. When $|\overline{\mathcal{Z}}| \ge 4T/\delta$ or the number of distinct elements in $\overline{\mathcal{Z}}$ exceeds

$$6912\dim_E(\mathcal{F},\delta/(16T^2))\log(16(H+1)^2T^2/\delta)\ln T\ln(4\mathcal{N}(\mathcal{F},\delta/(566T))/\delta),$$

we have $|\widehat{\mathcal{Z}}| = 0$ and thus $\widehat{\mathcal{F}} = \mathcal{F}$. Otherwise, $\widehat{\mathcal{F}}$ is defined by $\widehat{f}$ and $\widehat{\mathcal{Z}}$. Since $\widehat{f} \in \mathcal{C}(\mathcal{F},1/(8\sqrt{4T/\delta}))$, the total number of distinct $\widehat{f}$ is upper bounded by $\mathcal{N}(\mathcal{F},1/(8\sqrt{4T/\delta}))$. Since there are at most

$$6912\dim_E(\mathcal{F},\delta/(16T^2))\log(16(H+1)^2T^2/\delta)\ln T\ln(4\mathcal{N}(\mathcal{F},\delta/(566T))/\delta)$$

distinct elements in $\widehat{\mathcal{Z}}$, while each of them belongs to $\mathcal{C}(\mathcal{S}\times\mathcal{A},1/(8\sqrt{4T/\delta}))$ and $|\widehat{\mathcal{Z}}| \le 4T/\delta$, the total number of distinct $\widehat{\mathcal{Z}}$ is upper bounded by

$$\left(\mathcal{N}(\mathcal{S}\times\mathcal{A},1/(8\sqrt{4T/\delta}))\cdot 4T/\delta\right)^{6912\dim_E(\mathcal{F},\delta/(16T^2))\log(16(H+1)^2T^2/\delta)\ln T\ln(4\mathcal{N}(\mathcal{F},\delta/(566T))/\delta)}.$$

□

4.4.2 Analysis of the Algorithm

We are now ready to prove the regret bound of Algorithm 7. The next lemma establishes a bound on the estimate of a single backup.

Lemma 4.4.8 (Single Step Optimization Error)**.** *Consider a fixed $k \in [K]$. Let*

$$\mathcal{Z}^k = \{(s_{h'}^\tau, a_{h'}^\tau)\}_{(\tau,h')\in[k-1]\times[H]}$$

as defined in Line 5 in Algorithm 7. For any $V : \mathcal{S} \to [0,H]$, define

$$\mathcal{D}_V^k := \left\{\left(s_{h'}^\tau, a_{h'}^\tau, r_{h'}^\tau + V(s_{h'+1}^\tau)\right)\right\}_{(\tau,h')\in[k-1]\times[H]}$$

and

$$\widehat{f}_V := \arg\min_{f\in\mathcal{F}} \|f\|^2_{\mathcal{D}^k_V}.$$

For any $V : \mathcal{S} \to [0, H]$ and $\delta \in (0, 1)$, there is an event $\mathcal{E}_{V,\delta}$ which holds with probability at least $1 - \delta$, such that conditioned on $\mathcal{E}_{V,\delta}$, for any $V' : \mathcal{S} \to [0, H]$ with $\|V' - V\|_\infty \le 1/T$, we have

$$\left\|\widehat{f}_{V'}(\cdot,\cdot) - R(\cdot,\cdot) - \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot)V'(s')\right\|_{\mathcal{Z}^k} \le c' \cdot \left(H\sqrt{\log(2/\delta) + \log\mathcal{N}(\mathcal{F}, 1/T)}\right)$$

for some absolute constant $c' > 0$.

Proof. In our proof, we consider a fixed $V : \mathcal{S} \to [0, H]$, and define

$$f_V(\cdot,\cdot) := R(\cdot,\cdot) + \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot)V(s').$$

For any $f \in \mathcal{F}$, we consider $\sum_{(\tau,h)\in[k-1]\times[H]} \xi^\tau_h(f)$ where

$$\xi^\tau_h(f) := 2(f(s^\tau_h, a^\tau_h) - f_V(s^\tau_h, a^\tau_h)) \cdot (f_V(s^\tau_h, a^\tau_h) - r^\tau_h - V(s^\tau_{h+1})).$$

For any $(\tau, h) \in [k-1] \times [H]$, define $\mathbb{F}^\tau_h$ as the filtration induced by the sequence

$$\{(s^t_{h'}, a^t_{h'})\}_{(t,h')\in[\tau-1]\times[H]} \cup \{(s^\tau_1, a^\tau_1), (s^\tau_2, a^\tau_2), \ldots, (s^\tau_{h-1}, a^\tau_{h-1})\}.$$

Then $\mathbb{E}\left[\xi^\tau_h(f) \mid \mathbb{F}^\tau_h\right] = 0$ and

$$|\xi^\tau_h(f)| \le 2(H+1)\left|f(s^\tau_h, a^\tau_h) - f_V(s^\tau_h, a^\tau_h)\right|.$$

By Azuma-Hoeffding inequality, we have

$$\Pr\left[\left|\sum_{(\tau,h)\in[k-1]\times[H]} \xi^\tau_h(f)\right| \ge \varepsilon\right] \le 2\exp\left(-\frac{\varepsilon^2}{8(H+1)^2\|f - f_V\|^2_{\mathcal{Z}^k}}\right).$$

Let

$$\begin{aligned}\varepsilon &= \left(8(H+1)^2\log\left(\frac{2\mathcal{N}(\mathcal{F}, 1/T)}{\delta}\right)\cdot\|f - f_V\|^2_{\mathcal{Z}^k}\right)^{1/2} \\ &\le 4(H+1)\|f - f_V\|_{\mathcal{Z}^k}\cdot\sqrt{\log(2/\delta) + \log\mathcal{N}(\mathcal{F}, 1/T)}.\end{aligned}$$

We have, with probability at least $1 - \delta$, for all $f \in \mathcal{C}(\mathcal{F}, 1/T)$,

$$\left|\sum_{(\tau,h)\in[k-1]\times[H]} \xi^\tau_h(f)\right| \le 4(H+1)\|f - f_V\|_{\mathcal{Z}^k}\cdot\sqrt{\log(2/\delta) + \log\mathcal{N}(\mathcal{F}, 1/T)}.$$

We define the above event to be $\mathcal{E}_{V,\delta}$, and we condition on this event for the rest of the proof.

For all $f \in \mathcal{F}$, there exists $g \in \mathcal{C}(\mathcal{F}, 1/T)$, such that $\|f - g\|_\infty \le 1/T$, and we have

$$\begin{aligned}\left|\sum_{(\tau,h)\in[k-1]\times[H]} \xi_h^\tau(f)\right| &\le \left|\sum_{(\tau,h)\in[k-1]\times[H]} \xi_h^\tau(g)\right| + 2(H+1)\\ &\le 4(H+1)\|g - f_V\|_{\mathcal{Z}^k} \cdot \sqrt{\log(2/\delta) + \log \mathcal{N}(\mathcal{F}, 1/T)} + 2(H+1)\\ &\le 4(H+1)(\|f - f_V\|_{\mathcal{Z}^k} + 1) \cdot \sqrt{\log(2/\delta) + \log \mathcal{N}(\mathcal{F}, 1/T)} + 2(H+1).\end{aligned}$$

Consider $V' : \mathcal{S} \to [0, H]$ with $\|V' - V\|_\infty \le 1/T$. We have

$$\|f_{V'} - f_V\|_\infty \le \|V' - V\|_\infty \le 1/T.$$

For any $f \in \mathcal{F}$,

$$\begin{aligned}&\|f\|^2_{\mathcal{D}^k_{V'}} - \|f_{V'}\|^2_{\mathcal{D}^k_{V'}}\\ =&\|f - f_{V'}\|^2_{\mathcal{Z}^k} + 2\sum_{(s^\tau_{h'},a^\tau_{h'})\in\mathcal{Z}^k} (f(s^\tau_{h'},a^\tau_{h'}) - f_{V'}(s^\tau_{h'},a^\tau_{h'})) \cdot (f_{V'}(s^\tau_{h'},a^\tau_{h'}) - r^\tau_{h'} - V'(s^\tau_{h'+1})).\end{aligned}$$

For the second term, we have,

$$\begin{aligned}&2\sum_{(s^\tau_{h'},a^\tau_{h'})\in\mathcal{Z}^k} (f(s^\tau_{h'},a^\tau_{h'}) - f_{V'}(s^\tau_{h'},a^\tau_{h'})) \cdot (f_{V'}(s^\tau_{h'},a^\tau_{h'}) - r^\tau_{h'} - V'(s^\tau_{h'+1}))\\ \ge& 2\sum_{(s^\tau_{h'},a^\tau_{h'})\in\mathcal{Z}^k} (f(s^\tau_{h'},a^\tau_{h'}) - f_V(s^\tau_{h'},a^\tau_{h'})) \cdot (f_V(s^\tau_{h'},a^\tau_{h'}) - r^\tau_{h'} - V(s^\tau_{h'+1})) - 4(H+1)\cdot\|V' - V\|_\infty \cdot |\mathcal{Z}^k|\\ =&\sum_{(\tau,h)\in[k-1]\times[H]} \xi_h^\tau(f) - 4(H+1)\cdot\|V' - V\|_\infty \cdot |\mathcal{Z}^k|\\ \ge& -4(H+1)(\|f - f_V\|_{\mathcal{Z}^k} + 1) \cdot \sqrt{\log(2/\delta) + \log \mathcal{N}(\mathcal{F}, 1/T)} - 2(H+1) - 4(H+1)\cdot\|V' - V\|_\infty \cdot |\mathcal{Z}^k|\\ \ge& -4(H+1)(\|f - f_{V'}\|_{\mathcal{Z}^k} + 2) \cdot \sqrt{\log(2/\delta) + \log \mathcal{N}(\mathcal{F}, 1/T)} - 6(H+1).\end{aligned}$$

Recall that $\widehat{f}_{V'} = \arg\min_{f\in\mathcal{F}} \|f\|^2_{\mathcal{D}^k_{V'}}$. We have $\|\widehat{f}_{V'}\|^2_{\mathcal{D}^k_{V'}} - \|f_{V'}\|^2_{\mathcal{D}^k_{V'}} \le 0$, which implies,

$$\begin{aligned}0 \ge& \|\widehat{f}_{V'}\|^2_{\mathcal{D}^k_{V'}} - \|f_{V'}\|^2_{\mathcal{D}^k_{V'}}\\ =& \|\widehat{f}_{V'} - f_{V'}\|^2_{\mathcal{Z}^k} + 2\sum_{(s^\tau_{h'},a^\tau_{h'})\in\mathcal{Z}^k} (\widehat{f}(s^\tau_{h'},a^\tau_{h'}) - f_{V'}(s^\tau_{h'},a^\tau_{h'})) \cdot (f_{V'}(s^\tau_{h'},a^\tau_{h'}) - r^\tau_{h'} - V'(s^\tau_{h'+1}))\\ \ge& \|\widehat{f}_{V'} - f_{V'}\|^2_{\mathcal{Z}^k} - 4(H+1)(\|\widehat{f}_{V'} - f_{V'}\|_{\mathcal{Z}^k} + 2) \cdot \sqrt{\log(2/\delta) + \log \mathcal{N}(\mathcal{F}, 1/T)} - 6(H+1).\end{aligned}$$

Solving the above inequality, we have,

$$\|\widehat{f}_{V'} - f_{V'}\|_{\mathcal{Z}^k} \le c' \cdot \big(H \cdot \sqrt{\log \delta^{-1} + \log \mathcal{N}(\mathcal{F}, 1/T)}\big)$$

for an absolute constant $c' > 0$. □

Lemma 4.4.9 (Confidence Region)**.** *In Algorithm 7, let $\mathcal{F}_h^k$ be a confidence region defined as*

$$\mathcal{F}_h^k = \Big\{ f \in \mathcal{F} \mid \|f - f_h^k\|_{\mathcal{Z}^k}^2 \le \beta(\mathcal{F}, \delta) \Big\}.$$

Then with probability at least $1 - \delta/8$, for all $k, h \in [K] \times [H]$,

$$R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(s') \in \mathcal{F}_h^k,$$

provided

$$\beta(\mathcal{F}, \delta) \ge c' \cdot \Big(H \sqrt{\log(T/\delta) + \log(|\mathcal{W}|) + \log \mathcal{N}(\mathcal{F}, 1/T)} \Big)^2$$

for some absolute constant $c' > 0$. Here $\mathcal{W}$ is given as in Propostion 4.4.7.

Proof. For all $(k, h) \in [K] \times [H]$, the bonus function $b_h^k(\cdot,\cdot) \in \mathcal{W}$. Note that

$$\mathcal{Q} := \{ \min \{ f(\cdot,\cdot) + w(\cdot,\cdot), H \} \mid w \in \mathcal{W}, f \in \mathcal{C}(\mathcal{F}, 1/T) \} \cup \{0\}$$

is a $(1/T)$-cover of

$$Q_{h+1}^k(\cdot,\cdot) = \begin{cases} \min \left\{ f_{h+1}^k(\cdot,\cdot) + b_{h+1}^k(\cdot,\cdot), H \right\} & h < H \\ 0 & h = H \end{cases}.$$

I.e., there exists $q \in \mathcal{Q}$ such that $\|q - Q_{h+1}^k\|_\infty \le 1/T$. This implies

$$\mathcal{V} := \left\{ \max_{a \in \mathcal{A}} q(\cdot, a) \mid q \in \mathcal{Q} \right\}$$

is a $(1/T)$-cover of V_{h+1}^k with $\log(|\mathcal{V}|) \le \log|\mathcal{W}| + \log \mathcal{N}(\mathcal{F}, 1/T) + 1$. For each $V \in \mathcal{V}$, let $\mathcal{E}_{V,\delta/(8|\mathcal{V}|T)}$ be the event defined in Lemma 4.4.8. By Lemma 4.4.8, we have $\Pr\left[\bigcap_{V \in \mathcal{V}} \mathcal{E}_{V,\delta/(8|\mathcal{V}|T)}\right] \ge 1 - \delta/(8T)$. We condition on $\bigcap_{V \in \mathcal{V}} \mathcal{E}_{V,\delta/(8|\mathcal{V}|T)}$ in the rest part of the proof.

Recall that f_h^k is the solution of the optimization problem in Line 8 of Algorithm 7, i.e., $f_h^k = \operatorname{argmin}_{f \in \mathcal{F}} \|f\|_{\mathcal{D}_h^k}^2$. Let $V \in \mathcal{V}$ such that $\|V - V_{h+1}^k\|_\infty \le 1/T$. Thus, by Lemma 4.4.8, we have

$$\left\| f_h^k(\cdot,\cdot) - \left(R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(s') \right) \right\|_{\mathcal{Z}^k} \le c' \cdot \Big(H \sqrt{\log(T/\delta) + \log \mathcal{N}(\mathcal{F}, 1/T) + \log|\mathcal{W}|} \Big)$$

for some absolute constant c'. Therefore, by a union bound, for all $(k, h) \in [K] \times [H]$, we have $f_h^k(\cdot,\cdot) - \big(R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(s') \big) \in \mathcal{F}_h^k$ with probability at least $1 - \delta/8$. □

The above lemma guarantees that, with high probability, $R(\cdot,\cdot) + \sum_{s' \in \mathcal{S}} P(s' \mid \cdot,\cdot) V_{h+1}^k(\cdot,\cdot)$ lies in the confidence region. With this, it is guaranteed that $\left\{ Q_h^k \right\}_{(h,k) \in [H] \times [K]}$ are all optimistic, with high probability. This is formally presented in the next lemma.

Lemma 4.4.10. *With probability at least* $1-\delta/4$*, for all* $(k,h) \in [K]\times[H]$*, for all* $(s,a) \in \mathcal{S}\times\mathcal{A}$*,*

$$Q_h^*(s,a) \le Q_h^k(s,a) \le R(s,a) + \sum_{s'\in\mathcal{S}} P(s'|s,a)V_{h+1}^k(s') + 2b_h^k(s,a).$$

Proof. For each $(k,h) \in [K]\times[H]$, define

$$\mathcal{F}_h^k = \left\{ f \in \mathcal{F} \mid \|f - f_h^k\|_{\mathcal{Z}^k}^2 \le \beta(\mathcal{F},\delta) \right\}.$$

Let $\mathcal{E}$ be the event that for all $(k,h) \in [K]\times[H]$, $R(\cdot,\cdot) + \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot)V_{h+1}^k(s') \in \mathcal{F}_h^k$. By Lemma 4.4.9, $\Pr[\mathcal{E}] \ge 1-\delta/8$. Let $\mathcal{E}'$ be the event that for all $(k,h) \in [K]\times[H]$ and $(s,a) \in \mathcal{S}\times\mathcal{A}$, $b_h^k(s,a) \ge w(\mathcal{F}_h^k, s, a)$. By Proposition 4.4.7 and union bound, $\mathcal{E}'$ holds failure probability at most $\delta/8$. In the rest part of the proof we condition on $\mathcal{E}$ and $\mathcal{E}'$.

Note that

$$\max_{f\in\mathcal{F}_h^k} |f(s,a) - f_h^k(s,a)| \le w(\mathcal{F}_h^k, s, a) \le b_h^k(s,a).$$

Since

$$R(\cdot,\cdot) + \sum_{s'\in\mathcal{S}} P(s' \mid \cdot,\cdot)V_{h+1}^k(s') \in \mathcal{F}_h^k,$$

for any $(s,a) \in \mathcal{S}\times\mathcal{A}$ we have

$$\left| R(s,a) + \sum_{s'\in\mathcal{S}} P(s' \mid s,a)V_{h+1}^k(s') - f_h^k(s,a) \right| \le b_h^k(s,a).$$

Hence,

$$Q_h^k(s,a) \le f_h^k(s,a) + b_h^k(s,a) \le R(s,a) + \sum_{s'\in\mathcal{S}} P(s'|s,a)V_{h+1}^k(s') + 2b_h^k(s,a).$$

Now we prove $Q_h^*(s,a) \le Q_h^k(s,a)$ by induction on h. When $h = H+1$, the desired inequality clearly holds. Now we assume $Q_{h+1}^*(\cdot,\cdot) \le Q_{h+1}^k(\cdot,\cdot)$ for some $h \in [H]$. Clearly we have $V_{h+1}^*(\cdot) \le V_{h+1}^k(\cdot)$. Therefore, for all $(s,a) \in \mathcal{S}\times\mathcal{A}$,

$$\begin{aligned}
Q_h^*(s,a) &= R(s,a) + \sum_{s'\in\mathcal{S}} P(s'|s,a)V_{h+1}^*(s') \\
&\le \min\left\{ H, R(s,a) + \sum_{s'\in\mathcal{S}} P(s'|s,a)V_{h+1}^k(s') \right\} \\
&\le \min\left\{ H, f_h^k(s,a) + b_h^k(s,a) \right\} \\
&= Q_h^k(s,a).
\end{aligned}$$

□

The next lemma upper bounds the regret of the algorithm by the sum of $b_h^k(\cdot,\cdot)$.

Lemma 4.4.11. *With probability at least* $1-\delta/2$*,*

$$\mathrm{Reg}(K) \le 2\sum_{k=1}^{K}\sum_{h=1}^{H} b_h^k\left(s_h^k, a_h^k\right) + 4H\sqrt{KH\cdot\log(8/\delta)}.$$

Proof. In our proof, for any $(k,h)\in[K]\times[H-1]$ define

$$\xi_h^k = \sum_{s'\in\mathcal{S}} P(s' \mid s_h^k, a_h^k)\left(V_{h+1}^k(s') - V_{h+1}^{\pi_k}(s')\right) - \left(V_{h+1}^k(s_{h+1}^k) - V_{h+1}^{\pi_k}(s_{h+1}^k)\right)$$

and define $\mathbb{F}_h^k$ as the filtration induced by the sequence

$$\{(s_{h'}^\tau, a_{h'}^\tau)\}_{(\tau,h')\in[k-1]\times[H]} \cup \{(s_1^k,a_1^k),(s_2^k,a_2^k),\ldots,(s_h^k,a_h^k)\}.$$

Then

$$\mathbb{E}\left[\xi_h^k \mid \mathbb{F}_h^k\right] = 0 \text{ and } |\xi_h^k| \le 2H.$$

By Azuma-Hoeffding inequality, with probability at least $1-\delta/4$,

$$\sum_{k=1}^{K}\sum_{h=1}^{H-1}\xi_h^k \le 4H\sqrt{KH\cdot\log(8/\delta)}.$$

We condition on the above event in the rest of the proof. We also condition on the event defined in Lemma 4.4.10 which holds with probability $1-\delta/4$.

Recall that

$$\mathrm{Reg}(K) = \sum_{k=1}^{K}\left(V_1^*(s_1^k) - V_1^{\pi_k}(s_1^k)\right) \le \sum_{k=1}^{K} V_1^k(s_1^k) - V_1^{\pi_k}(s_1^k).$$

We have

$$\begin{aligned}
\mathrm{Reg}(K) &\le \sum_{k=1}^{K}\left(R(s_1^k,a_1^k) + \sum_{s'\in\mathcal{S}} P(s'\mid s_1^k,a_1^k)V_2^k(s') + 2b_1^k(s_1^k,a_1^k) - R(s_1^k,a_1^k) - \sum_{s'\in\mathcal{S}} P(s'\mid s_1^k,a_1^k)V_2^{\pi_k}(s')\right)\\
&= \sum_{k=1}^{K}\sum_{s'\in\mathcal{S}} P(s'\mid s_1^k,a_1^k)(V_2^k(s') - V_2^{\pi_k}(s')) + 2b_1^k(s_1^k,a_1^k)\\
&= \sum_{k=1}^{K} V_2^k(s_2^k) - V_2^{\pi_k}(s_2^k) + \xi_1^k + 2b_1^k(s_1^k,a_1^k)\\
&\le \sum_{k=1}^{K} V_3^k(s_3^k) - V_3^{\pi_k}(s_3^k) + \xi_1^k + \xi_2^k + 2b_1^k(s_1^k,a_1^k) + 2b_2^k(s_2^k,a_2^k)\\
&\le \sum_{k=1}^{K}\sum_{h=1}^{H-1}\xi_h^k + \sum_{k=1}^{K}\sum_{h=1}^{H} 2b_h^k(s_h^k,a_h^k).
\end{aligned}$$

Therefore,

$$\mathrm{Reg}(K) \le 2\sum_{k=1}^{K}\sum_{h=1}^{H} b_h^k(s_h^k, a_h^k) + 4H\sqrt{KH \cdot \log(8/\delta)}.$$

□

It remains to bound $\sum_{k=1}^{K}\sum_{h=1}^{H} b_h^k(s_h^k, a_h^k)$, for which we will exploit fact that $\mathcal{F}$ has bounded eluder dimension.

Lemma 4.4.12. *With probability at least $1-\delta/4$, for any $\varepsilon > 0$,*

$$\sum_{k=1}^{K}\sum_{h=1}^{H} \mathbb{I}\left(b_h^k(s_h^k, a_h^k) > \varepsilon\right) \le \left(\frac{c\beta(\mathcal{F}, \delta)}{\varepsilon^2} + H\right) \cdot \dim_E(\mathcal{F}, \varepsilon)$$

for some absolute constant $c > 0$. Here $\beta(\mathcal{F}, \delta)$ is as defined in (4.4).

Proof. Let $\mathcal{E}$ be the event that or all $(k, h) \in [K] \times [H]$,

$$b_h^k(\cdot, \cdot) \le w(\overline{\mathcal{F}}_h^k, \cdot, \cdot)$$

where

$$\overline{\mathcal{F}}_h^k = \{f \in \mathcal{F} : \|f - f_h^k\|_{\mathcal{Z}^k}^2 \le 9\beta + 12\}.$$

By Proposition 4.4.7, $\mathcal{E}$ holds with probability at least $1 - \delta/4$. In the rest of the proof, we condition on $\mathcal{E}$.

Let $\mathcal{L} = \{(s_h^k, a_h^k) \mid b_h^k(s_h^k, a_h^k) > \varepsilon\}$ with $|\mathcal{L}| = L$. We show that there exists $(s_h^k, a_h^k) \in \mathcal{L}$ such that (s_h^k, a_h^k) is ε-dependent on at least $L/\dim_E(\mathcal{F}, \varepsilon) - H$ disjoint subsequences in $\mathcal{Z}^k \cap \mathcal{L}$. We demonstrate this by using the following procedure. Let $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{L/\dim_E(\mathcal{F},\varepsilon)-1}$ be $L/\dim_E(\mathcal{F}, \varepsilon) - 1$ disjoint subsequences of $\mathcal{L}$ which are initially empty. We consider

$$\{(s_1^k, a_1^k), (s_2^k, a_2^k), \ldots, (s_H^k, a_H^k)\} \cap \mathcal{L}$$

for each $k \in [K]$ sequentially. For each $k \in [K]$, for each $z \in \{(s_1^k, a_1^k), (s_2^k, a_2^k), \ldots, (s_H^k, a_H^k)\} \cap \mathcal{L}$, we find $j \in [L/\dim_E(\mathcal{F}, \varepsilon) - 1]$ such that z is ε-independent of $\mathcal{L}_j$ and then add z into $\mathcal{L}_j$. By the definition of ε-independence, $|\mathcal{L}_j| \le \dim_E(\mathcal{F}, \varepsilon)$ for all j and thus we will eventually find some $(s_h^k, a_h^k) \in \mathcal{L}$ such that (s_h^k, a_h^k) is ε-dependent on each of $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{L/\dim_E(\mathcal{F},\varepsilon)-1}$. Among $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{L/\dim_E(\mathcal{F},\varepsilon)-1}$, there are at most $H - 1$ of them that contain an element in

$$\{(s_1^k, a_1^k), (s_2^k, a_2^k), \ldots, (s_H^k, a_H^k)\} \cap \mathcal{L},$$

and all other subsequences only contain elements in $\mathcal{Z}^k \cap \mathcal{L}$. Therefore, (s_h^k, a_h^k) is ε-dependent on at least $L/\dim_E(\mathcal{F}, \varepsilon) - H$ disjoint subsequences in $\mathcal{Z}^k \cap \mathcal{L}$.

On the other hand, since $(s_h^k, a_h^k) \in \mathcal{L}$, we have $b_h^k(s_h^k, a_h^k) > \varepsilon$, which implies there exists $f, f' \in \mathcal{F}$ with $\|f - f_h^k\|_{\mathcal{Z}^k}^2 \le 9\beta + 12$ and $\|f' - f_h^k\|_{\mathcal{Z}^k}^2 \le 9\beta + 12$ such that $f(z) - f'(z) > \varepsilon$. By triangle inequality, we have $\|f - f'\|_{\mathcal{Z}^k}^2 \le 36\beta + 48$. On the other hand, since (s_h^k, a_h^k) is ε-dependent on at least $L/\dim_E(\mathcal{F}, \varepsilon) - H$ disjoint subsequences in $\mathcal{Z}^k \cap \mathcal{L}$, we have

$$(L/\dim_E(\mathcal{F}, \varepsilon) - H)\varepsilon^2 \le \|f - f\|_{\mathcal{Z}^k}^2 \le 36\beta + 48,$$

which implies

$$L \le \left(\frac{36\beta + 48}{\varepsilon^2} + H\right) \dim_E(\mathcal{F}, \varepsilon).$$

□

Lastly, we apply the above lemma to bound the overall regret.

Lemma 4.4.13. *With probability at least* $1 - \delta/4$*,*

$$\sum_{k=1}^{K} \sum_{1}^{H} b_h^k(s_h^k, a_h^k) \le 1 + 4H^2 \dim_E(\mathcal{F}, 1/T) + \sqrt{c \cdot \dim_E(\mathcal{F}, 1/T) \cdot T \cdot \beta(\mathcal{F}, \delta)},$$

for some absolute constant $c > 0$*. Here* $\beta(\mathcal{F}, \delta)$ *is as defined in* (4.4)*.*

Proof. In the proof we condition on the event defined in Lemma 4.4.12. We define $w_h^k := b_h^k\left(s_h^k, a_h^k\right)$. Let $w_1 \ge w_2 \ge \ldots \ge w_T$ be a permutation of $\{w_h^k\}_{(k,h)\in[K]\times[H]}$. By the event defined in Lemma 4.4.12, for any $w_t \ge 1/T$, we have

$$t \le \left(\frac{c\beta(\mathcal{F}, \delta)}{w_t^2} + H\right) \dim_E(\mathcal{F}, w_t) \le \left(\frac{c\beta(\mathcal{F}, \delta)}{w_t^2} + H\right) \dim_E(\mathcal{F}, 1/T),$$

which implies

$$w_t \le \left(\frac{t}{\dim_E(\mathcal{F}, 1/T)} - H\right)^{-1/2} \cdot \sqrt{c\beta(\mathcal{F}, \delta)}.$$

Moreover, we have $w_t \le 4H$. Therefore,

$$\begin{aligned}\sum_{t=1}^{T} w_t \le& 1 + 4H^2 \dim_E(\mathcal{F}, 1/T) + \sum_{H \dim_E(\mathcal{F}, 1/T) < t \le T} \left(\frac{t}{\dim_E(\mathcal{F}, 1/T)} - H\right)^{-1/2} \cdot \sqrt{c\beta(\mathcal{F}, \delta)} \\ \le& 1 + 4H^2 \dim_E(\mathcal{F}, 1/T) + 2\sqrt{c \cdot \dim_E(\mathcal{F}, 1/T) \cdot T \cdot \beta(\mathcal{F}, \delta)}.\end{aligned}$$

□

We are now ready to prove our main theorem.

Proof of Theorem 8.3.1. By Lemma 4.4.11 and Lemma 4.4.13, with probability at least $1 - \delta$,

$$\begin{aligned}\mathrm{Reg}(K) &\le \min\left\{KH, \sum_{k=1}^{K}\sum_{h=1}^{H} 2b_h^k\left(s_h^k, a_h^k\right) + 4H\sqrt{KH \cdot \log(8/\delta)}\right\} \\ &\le c \cdot \min\left\{KH, \left(\dim_E(\mathcal{F}, 1/T) \cdot H^2 + \sqrt{\dim_E(\mathcal{F}, 1/T) \cdot T \cdot \beta(\mathcal{F}, \delta)} + H\sqrt{KH \cdot \log \delta^{-1}}\right)\right\}\end{aligned}$$

for some absolute constants $c > 0$. Substituting the value of $\beta(\mathcal{F}, \delta)$ completes the proof. □

Chapter 5

RL with Large State Spaces: Lower Bounds in the Online Setting

5.1 Introduction

Modern reinforcement learning (RL) problems are often challenging due to the huge state space. To tackle this challenge, function approximation schemes are often employed to provide a compact representation, so that reinforcement learning can generalize across states. Empirically, combining various RL function approximation algorithms with neural networks for feature extraction has lead to tremendous successes on various tasks [58, 74]. A major problem, however, is that these methods often require a large amount of samples to learn a good policy. For example, deep Q-network requires millions of samples to solve certain Atari games [58]. Here, one may wonder if there are fundamental statistical limitations on such methods, and, if so, under what conditions it would be possible to efficiently learn a good policy?

In the supervised learning context, it is well-known that empirical risk minimization is a statistically efficient method when using a low-complexity hypothesis space [75], e.g. a hypothesis space with bounded VC dimension. For example, polynomial number of samples suffice for learning a near-optimal d-dimensional linear classifier, even in the agnostic setting. In contrast, in the more challenging RL setting, we seek to understand if efficient learning is possible (say from a sample complexity perspective) when we have access to an accurate (and compact) parametric representation — e.g. our policy class contains a near-optimal policy or our hypothesis class accurately approximates the optimal value function. In particular, in this section, we explore if a good representation is sufficient for sample-efficient reinforcement learning. This question has largely been studied only with respect to approximation error in the more classical approximate dynamic programming literature, where it is known that algorithms are stable to certain worst-case approximation errors. With regards to sample efficiency, this question is largely unexplored, where the extant body of literature mainly focuses on conditions which are *sufficient* for efficient reinforcement learning though there is little understanding of what are *necessary* conditions for efficient reinforcement learning.

Many recent works have provided polynomial upper bounds under various sufficient conditions (including the one in Chapter 4), and in what follows we list a few examples (other than

the one that appears in Chapter 4). For value-based learning, the work of [98] showed that for deterministic systems, if the optimal Q-function can be *perfectly* predicted by linear functions of the given features, then the agent can learn the optimal policy exactly with polynomial number of samples. Recent work [38] further showed that if certain complexity measure called *Bellman rank* is bounded, then the agent can learn a near-optimal policy efficiently. For policy-based learning, [3] gave polynomial upper bounds which depend on a parameter that measures the difference between the initial distribution and the distribution induced by the optimal policy.

Our Results. In this section, we give, perhaps surprisingly, strong *negative* results to this question. The main results are *exponential lower bounds* in terms of planning horizon H for value-based, model-based, and policy-based algorithms with given good representations. Notably, the requirements on the representation that suffice for sample efficient RL are even more stringent than the more traditional approximation viewpoint. Here we briefly summarize our hardness results.

1. For value-based learning, we show even if Q-functions of all policies can be approximated by linear functions of the given representation with approximation error $\delta = \Omega\left(\sqrt{\frac{H}{d}}\right)$ where d is the dimension of the representation and H is the planning horizon, then the agent still needs to sample exponential number of trajectories to find a near-optimal policy.
2. We show even if optimal policy can be *perfectly* predicted by a linear function of the given representation with a strictly positive margin, the agent still requires exponential number of trajectories to find a near-optimal policy.

These lower bounds hold even in deterministic systems and even if the agent knows the transition model (i.e., in the Known Transition model). Note these negative results apply to the case where the Q-function, the model, or the optimal policy can be predicted well by a linear function of the given representation. Our results highlight that the requirements on the representation that suffice for sample efficient RL are significantly more stringent than the more traditional approximation viewpoint and those in supervised learning.

Furthermore, our work implies several interesting exponential separations on the sample complexity between: 1) value-based learning with perfect representation and value-based learning with a good-but-not-perfect representation, 2) value-based learning and policy-based learning, 3) policy-based learning and supervised learning and 4) reinforcement learning and imitation learning. More details will be given in Section 5.3.

5.2 Main Results

In this section we formally present our lower bounds. To streamline our analysis, in this chapter, for each $h \in [H]$, we use $\mathcal{S}_h \subseteq \mathcal{S}$ to denote the set of states at level h, and we assume $\mathcal{S}_h$ do not intersect with each other. We also assume $\sum_{h=1}^{H} r_h \in [0, 1]$ almost surely.

Before stating our results, we first list an important assumption, the optimality gap assumption, which is widely used in reinforcement learning and bandit literature. To state the assumption, we first define the function $\mathrm{gap} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as $\mathrm{gap}(s, a) = \max_{a' \in \mathcal{A}} Q_h^*(s, a') - Q_h^*(s, a)$ suppose $s \in \mathcal{S}_h$. Now we formally state the assumption.

Assumption 5.2.1 (Optimality Gap)**.** *There exists* $\rho > 0$ *such that* $\rho \leq \operatorname{gap}(s,a)$ *for all* $(s,a) \in \mathcal{S} \times \mathcal{A}$ *with* $\operatorname{gap}(s,a) > 0$.

Here, ρ is the smallest reward-to-go difference between the best set of actions and the rest. Recently, [26] gave a provably efficient Q-learning algorithm based on this assumption, and [78] showed that with this condition, the agent only incurs logarithmic regret in the tabular setting.

5.2.1 Lower Bound for Value-Based Learning

We first present our lower bound for value-based learning. A common assumption is that the Q-function can be predicted well by a linear function of the given features (representation). Formally, the agent is given a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ which can be hand-crafted or a pre-trained neural network that transforms a state-action pair to a d-dimensional embedding. The following assumption states that the given feature extractor can be used to predict the Q-function with approximation error at most δ using a linear function.

Assumption 5.2.2. *There exists* $\delta > 0$ *and* $\theta_1, \theta_2, \ldots, \theta_H \in \mathbb{R}^d$ *such that for any* $h \in [H]$ *and any* $(s,a) \in \mathcal{S}_h \times \mathcal{A}$, $|Q_h^*(s,a) - \langle \theta_h, \phi(s,a) \rangle| \leq \delta$.

Here δ is the approximation error, which indicates the quality of the representation. If $\delta = 0$, then Q-function can be perfectly predicted by a linear function of $\phi(\cdot,\cdot)$. In general, δ becomes smaller as we increase the dimension of ϕ, since larger dimension usually has more expressive power. When the feature extractor is strong enough, previous papers [16] assume that linear functions of ϕ can approximate the Q-function of *any* policy.

Assumption 5.2.3. *There exists* $\delta > 0$*, such that for any* $h \in [H]$ *and any policy* π*, there exists* $\theta_h^\pi \in \mathbb{R}^d$ *such that for any* $(s,a) \in \mathcal{S}_h \times \mathcal{A}$, $|Q_h^\pi(s,a) - \langle \theta_h, \phi(s,a) \rangle| \leq \delta$.

In the reinforcement learning literature, Assumption 5.2.3 is crucial in proving polynomial sample complexity guarantee for value iteration type of algorithms [16].

The following theorem shows when $\delta = \Omega\left(\sqrt{\frac{H}{d}}\right)$, the agent needs to sample exponential number of trajectories to find a near-optimal policy.

Theorem 5.2.1 (Exponential Lower Bound for Value-Based Learning)**.** *There exists a family of MDPs with* $|\mathcal{A}| = 2$ *and a feature extractor* ϕ *that satisfy Assumption 5.2.3, such that any algorithm that returns a* $1/2$*-optimal policy with probability* 0.9 *needs to sample*

$$\Omega\left(\min\{|\mathcal{S}|, 2^H, \exp(d\delta^2/16)\}\right)$$

trajectories.

Note this lower bound also applies to MDPs that satisfy Assumption 5.2.2, since Assumption 5.2.3 is strictly stronger. We would like to emphasize that since linear functions is a subclass of more complicated function classes, e.g., neural networks, our lower bound also holds for these function classes. Moreover, in many scenarios, the feature extractor ϕ is the last layer of a neural network. Modern neural networks are often over-parameterized, which makes d large. In this case, d is much larger than H. Thus, our lower bound holds even if the representation has small approximation error. Furthermore, the assumption that $|\mathcal{A}| = 2$ is only for simplicity. Our lower bound can be easily generalized to the case that $|\mathcal{A}| > 2$, in which case the sample complexity lower bound is $\Omega\left(\min\{|\mathcal{S}|, |\mathcal{A}|^H, \exp(d\delta^2/16)\}\right)$.

5.2.2 Lower Bound for Policy-Based Learning

Next we present our lower bound for policy-based learning. This class of methods use function approximation on the policy and use optimization techniques, e.g., policy gradient, to find the optimal policy. In this section, we focus on linear policies on top of a given representation. A linear policy π is a policy of the form $\pi(s_h) = \arg\max_{a \in \mathcal{A}} \langle \theta_h, \phi(s_h, a) \rangle$ where $s_h \in \mathcal{S}_h$, $\phi(\cdot, \cdot)$ is a given feature extractor and $\theta_h \in \mathbb{R}^d$ is the linear coefficient. Note that applying policy gradient on softmax parameterization of the policy is indeed trying to find the optimal policy among linear policies.

Similar to value-based learning, a natural assumption for policy-based learning is that the optimal policy is realizable, i.e., the optimal policy is linear.

Assumption 5.2.4. *For any $h \in [H]$, there exists $\theta_h \in \mathbb{R}^d$ that satisfies for any $s \in \mathcal{S}_h$, we have $\pi^*(s) \in \operatorname{argmax}_a \langle \theta_h, \phi(s, a) \rangle$.*

Here we discuss another assumption. For learning a linear classifier in the supervised learning setting, one can reduce the sample complexity significantly if the optimal linear classifier has a margin.

Assumption 5.2.5. *We assume $\phi(s, a) \in \mathbb{R}^d$ satisfies $\|\phi(s, a)\|_2 = 1$ for any $(s, a) \in \mathcal{S} \times \mathcal{A}$. For any $h \in [H]$, there exists $\theta_h \in \mathbb{R}^d$ with $\|\theta_h\|_2 = 1$ and $\triangle > 0$ such that for any $s \in \mathcal{S}_h$, there is a* unique *optimal action $\pi^*(s)$, and for any $a \neq \pi^*(s)$, $\langle \theta_h, \phi(s, \pi^*(s)) \rangle - \langle \theta_h, \phi(s, a) \rangle \geq \triangle$.*

Here we restrict the linear coefficients and features to have unit norm for normalization. Note that Assumption 5.2.5 is strictly stronger than Assumption 5.2.4. Now we present our result for linear policy.

Theorem 5.2.2 (Exponential Lower Bound for Policy-based Learning)**.** *There exists an absolute constant $\triangle_0$, such that for any $\triangle \leq \triangle_0$, there exists a family of MDPs with $|\mathcal{A}| = 2$ and a feature extractor ϕ that satisfy Assumption 5.2.1 with $\rho = \frac{1}{2\min\{H,d\}}$ and Assumption 5.2.5, such that any algorithm that returns a $1/4$-optimal policy with probability at least 0.9 needs to sample $\Omega\left(\min\{2^H, 2^d\}\right)$ trajectories.*

Again, our lower bound can be easily generalized to the case that $|\mathcal{A}| > 2$.

Compared with Theorem 5.2.1, Theorem 5.2.2 is even more pessimistic, in the sense that even with perfect representation with benign properties (gap and margin), the agent still needs to sample exponential number of samples. It also suggests that policy-based learning could be very different from supervised learning.

5.2.3 Proof Ideas

The Binary Tree Hard Instance. All our lower bound are proved based on reductions from the binary tree instance. In this instance, both the transition P and the reward R are deterministic. There are H levels of states, which form a full binary tree of depth H. There are 2^{h-1} states in level h, and thus $2^H - 1$ states in total. Among all the 2^{H-1} states in level H, there is only one state with reward $R = 1$, and for all other states in the MDP, the corresponding reward value $R = 0$. Intuitively, to find a $1/2$-optimal policy for such MDPs, the agent must enumerate all possible states in level H to find the state with reward $R = 1$. Doing so intrinsically induces a sample complexity of $\Omega(2^H)$. This intuition is formalized in Theorem 5.4.1 using Yao's minimax principle [104].

Lower Bound for Value-Based Learning We now show how to construct a set of features so that Assumption 5.2.3 holds. Our main idea is to the utilize the following fact regarding the identity matrix: $\varepsilon\text{-rank}(I_{2^H}) \leq O(H/\varepsilon^2)$. Here for a matrix $A \in \mathbb{R}^{n\times n}$, its ε-rank (a.k.a *approximate rank*) is defined to be $\min\{\text{rank}(B) : B \in \mathbb{R}^{n\times n}, \|A - B\|_\infty \leq \varepsilon\}$, where we use $\|\cdot\|_\infty$ to denote the entry-wise ℓ_∞ norm of a matrix. The upper bound $\varepsilon\text{-rank}(I_n) \leq O(\log n/\varepsilon^2)$ was first proved in [5] using the Johnson-Lindenstrauss Lemma [41], and we also provide a proof in Lemma 5.4.2.

This fact can be alternatively stated as follow: there exists $\Phi \in \mathbb{R}^{2^H\times O(H/\varepsilon^2)}$ such that $\|I_{2^H} - \Phi\Phi^\top\|_\infty \leq \varepsilon$. We interpret each row of Φ as the feature of a state in the binary tree. By construction of Φ, now features of states in the binary tree have a nice property that (i) each feature vector has approximately unit norm and (ii) different feature vector are nearly orthogonal. Using this set of features, we can now show that Assumption 5.2.3 hold. Here we prove Assumption 5.2.3 holds as an example and prove other assumptions also hold in the appendix. To prove Assumption 5.2.3, we note that in the binary tree hard instance, for each level h, only a single state satisfies $Q^* = 1$, and all other states satisfy $Q^* = 0$. We simply take θ_h to be the feature of the state with $Q^* = 1$. Since all feature vectors are nearly orthogonal, Assumption 5.2.3 holds.

Since the above fact regarding the ε-rank of the identity matrix can be proved by simply taking each row of Φ to be a random unit vector, our lower bound reveals another intriguing (yet pessimistic) aspect of Assumption 5.2.3: for the binary tree instance, almost all feature extractors induce a hard MDP instance. This again suggests that a good representation itself may not necessarily lead to efficient RL and additional assumptions (e.g. on the reward distribution) could be crucial.

Lower Bound for Policy-Based Learning. It is straightfoward to construct a set of feature vectors for the binary tree instance so that Assumption 5.2.4 holds, even if $d = 1$. We set $\phi(s, a)$ to be $+1$ if $a = a_1$ and -1 if $a = a_2$. For each level h, for the unique state s in level h with $Q^* = 1$, we set θ_h to be 1 if $\pi^*(s) = a_1$ and -1 if $\pi^*(s) = a_2$. With this construction, Assumption 5.2.4 holds.

To prove that the lower bound under Assumption 5.2.5, we use a new reward function for states in level H in the binary tree instance above so that there exists a unique optimal action for each state in the MDP. See Figure 5.2 for an example with $H = 3$ levels of states. Another nice property of the new reward function is that for all states s we always have $\pi^*(s) = a_1$. Now, we define 2^{H-1} different new MDPs as follow: for each state in level H, we change its original reward (defined in Figure 5.2) to 1. An exponential sample complexity lower bound for these MDPs can be proved using the same argument as the original binary tree hard instance, and now we show this set of MDPs satisfy Assumption 5.2.5. We first show in Lemma 5.4.4 that there exists a set $\mathcal{N} \subseteq \mathbb{S}^{d-1}$ with $|\mathcal{N}| = (1/\triangle)^{\Omega(d)}$, so that for each $p \in \mathcal{N}$, there exists a hyperplane L that separates p and $\mathcal{N} \setminus \{p\}$, and all vectors in $\mathcal{N}$ have distance at least $\triangle$ to L. Equivalently, for each $p \in \mathcal{N}$,we can always define a linear function f_p so that $f_p(p) \geq \triangle$ and $f_p(q) \leq -\triangle$ for all $q \in \mathcal{N} \setminus \{p\}$. This can be proved using standard lower bounds on the size of ε-nets. Now we simply use vectors in $\mathcal{N}$ as features of states. By construction of the reward function, for each level h, there could only be two possible cases for the optimal policy π^*. I.e., either $\pi^*(s) = a_1$ for all states in level h, or $\pi^*(s) = a_2$ for a unique state s and $\pi^*(s') = a_1$ for all $s \neq s'$. In both

cases, we can easily define a linear function with margin $\triangle$ to implement the optimal policy π^*, and thus Assumption 5.2.5 holds. Notice that in this proof, we critically relies on $d = \Theta(H)$, so that we can utilize the curse of dimensionality to construct a large set of vectors as features.

5.3 Separations

Perfect Representation vs. Good-But-Not-Perfect Representation. For value-based learning in deterministic systems, [98] showed polynomial sample complexity upper bound when the representation can perfectly predict the Q-function. In contrast, if the representation is only able to *approximate* the Q-function, then the agent requires exponential number of trajectories. This exponential separation demonstrates a *provable exponential benefit of better representation.*

Value-Based Learning vs. Policy-Based Learning. Note that if the optimal Q-function can be perfectly predicted by the provided representation, then the optimal policy can also be perfectly predicted using the same representation. Since [98] showed polynomial sample complexity upper bound when the representation can perfectly predict the Q-function, our lower bound on policy-based learning, which applies to perfect representations, thus demonstrates that *the ability of predicting the Q-function is much stronger than that of predicting the optimal policy.*

Supervised Learning vs. Reinforcement Learning. For policy-based learning, if the planning horizon $H = 1$, the problem becomes learning a linear classifier, for which there are polynomial sample complexity upper bounds. For policy-based learning, the agent needs to learn H linear classifiers sequentially. Our lower bound on policy-based learning shows the sample complexity dependency on H is exponential.

Imitation Learning vs. Reinforcement Learning. In imitation learning (IL), the agent can observe trajectories induced by the optimal policy (expert). If the optimal policy is linear in the given representation, it can be shown that the simple behavior cloning algorithm only requires polynomial number of samples to find a near-optimal policy [72]. Our Theorem 5.2.2 shows if the agent cannot observe expert's behavior, then it requires exponential number of samples. Therefore, our lower bound shows there is an *exponential separation between policy-based RL and IL* when function approximation is used.

5.4 Proofs of Lower Bounds

In this section we present the proof of our lower bounds. Throughout this section, for the Q-function Q_h^π and Q_h^* and the value function V_h^π and V_h^*, we may omit h from the subscript when it is clear from the context.

We first introduce the INDEX-QUERY problem, which will be useful in our lower bound arguments.

Definition 5.4.1 (INDEX-QUERY)**.** In the INDQ_n problem, there is an underlying integer $i^* \in [n]$. The algorithm sequentially (and adaptively) outputs guesses $i \in [n]$ and queries whether $i = i^*$. The goal is to output i^*, using as few queries as possible.

Definition 5.4.2 (δ-correct algorithms)**.** For a real number $\delta \in (0, 1)$, we say a randomized algorithm $\mathcal{A}$ is δ-correct for INDQ_n, if for any underlying integer $i^* \in [n]$, with probability at least $1 - \delta$, $\mathcal{A}$ outputs i^*.

The following theorem states the query complexity of INDQ_n for 0.1-correct algorithms.

Theorem 5.4.1. *Any 0.1-correct algorithm $\mathcal{A}$ for INDQ_n requires at least $0.9n$ queries in the worst case.*

Proof. The proof is a straightforward application of Yao's minimax principle [104]. We provide the full proof for completeness.

Consider an input distribution where i^* is drawn uniformly at random from $[n]$. Suppose there is a 0.1-correct algorithm for INDQ_n with worst-case query complexity T such that $T < 0.9n$. By averaging, there is a deterministic algorithm $\mathcal{A}'$ with worst-case query complexity T, such that

$$\Pr_{i \sim [n]}[\mathcal{A}' \text{ correctly outputs } i \text{ when } i^* = i] \geq 0.9.$$

We may assume that the sequence of queries made by $\mathcal{A}'$ is fixed. This is because (i) $\mathcal{A}'$ is deterministic and (ii) before $\mathcal{A}'$ correctly guesses i^*, all responses that $\mathcal{A}'$ receives are the same (i.e., all guesses are incorrect). We use $S = \{s_1, s_2, \ldots, s_m\}$ to denote the sequence of queries made by $\mathcal{A}'$. Notice that m is the worst-case query complexity of $\mathcal{A}'$. Suppose $m < 0.9n$, there exist $0.1n$ distinct $i \in [n]$ such that $\mathcal{A}'$ will never guess i, and will be incorrect if i^* equals i, which implies

$$\Pr_{i \sim [n]}[\mathcal{A}' \text{ correctly outputs } i \text{ when } i^* = i] < 0.9.$$

□

5.4.1 Proof of Lower Bound for Value-Based Learning

In this section we prove Theorem 5.2.1. We need the following existential result.

Lemma 5.4.2. *For any $n > 2$, there exists a set of vectors $\mathcal{P} = \{p_1, p_2, \ldots, p_n\} \subset \mathbb{R}^d$ with $d = \lceil 8 \ln n/\varepsilon^2 \rceil$ such that*

1. *$\|p_i\|_2 = 1$ for all $i \in [n]$;*
2. *$|\langle p_i, p_j \rangle| \leq \varepsilon$ for any $i, j \in [n]$ with $i \neq j$.*

In order to prove Lemma 5.4.2, we need the following tail inequality for random unit vectors.

Lemma 5.4.3 (Lemma 2.2 in [19])**.** *For a random unit vector u in $\mathbb{R}^d$ and $\beta > 1$, we have*

$$\Pr\left[u_1^2 \geq \beta/d\right] \leq \exp((1 + \ln \beta - \beta)/2).$$

In particular, when $\beta \geq 6$,we have

$$\Pr\left[u_1^2 > \beta/d\right] \leq \exp(-\beta/4).$$

Proof of Lemma 5.4.2. Let $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$ be a set of n independent random unit vectors in $\mathbb{R}^d$ with $d = \lceil 8 \ln n/\varepsilon^2 \rceil$. We will prove that with probability at least $1/2$, $\mathcal{Q}$ satisfies the two desired properties as stated in Lemma 5.4.2. This implies the existence of such set $\mathcal{P}$.

It is clear that $\|q_i\|_2 = 1$ for all $i \in [n]$, since each q_i is drawn from the unit sphere. We now prove that for any $i, j \in [n]$ with $i \neq j$, with probability at least $1 - \frac{1}{n^2}$, we have $|\langle q_i, q_j \rangle| \leq \varepsilon$. Notice that this is sufficient to prove the lemma, since by a union bound over all the $\binom{n}{2} = n(n-1)/2$ possible pairs of (i, j), this implies that $\mathcal{Q}$ satisfies the two desired properties with probability at least $1/2$.

Now, we prove that for two independent random unit vectors u and v in $\mathbb{R}^d$ with $d = \lceil 8 \ln n/\varepsilon^2 \rceil$, with probability at least $1 - \frac{1}{n^2}$, $|\langle u, v \rangle| \leq \varepsilon$. By rotational invariance, we assume that v is a standard basis vector. I.e., we assume $v_1 = 1$ and $v_i = 0$ for all $1 < i \leq d$. Notice that now $\langle u, v \rangle$ is the magnitude of the first coordinate of u. We finish the proof by invoking Lemma 5.4.3 and taking $\beta = 8 \ln n > 6$. □

Now we give the construction of the hard MDP instances. We first define the transitions and the reward functions. In the hard instances, both the rewards and the transitions are deterministic. There are H levels of states, and level $h \in [H]$ contains 2^{h-1} distinct states. Thus we have $|\mathcal{S}| = 2^H - 1$. If $|\mathcal{S}| > 2^H - 1$ we simply add dummy states to the state space $\mathcal{S}$. We use $s_1, s_2, \ldots, s_{2^H-1}$ to name these states. Here, s_1 is the unique state in level $h = 1$, s_2 and s_3 are the two states in level $h = 2$, s_4, s_5, s_6 and s_7 are the four states in level $h = 3$, etc. There are two different actions, a_1 and a_2, in the MDPs. For a state s_i in level h with $h \leq H - 1$, playing action a_1 transits state s_i to state s_{2i} and playing action a_2 transits state s_i to state s_{2i+1}, where s_{2i} and s_{2i+1} are both states in level $h + 1$. See Figure 5.1 for an example with $H = 3$.

In our hard instances, $R(s, a) = 0$ for all (s, a) pairs except for a unique state s in level $H - 1$ and a unique action $a \in \{a_1, a_2\}$. It is convenient to define $\overline{r}(s') = R(s, a)$, if choosing action a transits s to s'. For our hard instances, we have $\overline{r}(s) = 1$ for a unique state s in level H and $\overline{r}(s) = 0$ for all other states.

Now we define the features map $\phi(\cdot, \cdot)$. Here we assume $d \geq 2 \cdot \lceil 8 \ln 2 \cdot H/\delta^2 \rceil$, since otherwise we can simply decrease the planning horizon so that $d \geq 2 \cdot \lceil 8 \ln 2 \cdot H/\delta^2 \rceil$. We invoke Lemma 5.4.2 to get a set $\mathcal{P} = \{p_1, p_2, \ldots, p_{2^H}\} \subset \mathbb{R}^{d/2}$. For each state s_i, $\phi(s_i, a_1) \in \mathbb{R}^d$ is defined to be $[p_i; 0]$, and $\phi(s_i, a_2) \in \mathbb{R}^d$ is defined to be $[0; p_i]$. This finishes the definition of the MDPs. We now show that no matter which state s in level H satisfies $\overline{r}(s) = 1$, the resulting MDP always satisfies Assumption 5.2.3.

Verifying Assumption 5.2.3. By construction, for each level $h \in [H]$, there is a unique state s^h in level h and action $a^h \in \{a_1, a_2\}$, such that $Q^*(s^h, a^h) = 1$. For all other (s, a) such that $s \neq s^h$ or $a \neq a^h$ we have $Q^*(s, a) = 0$. For a given level h and policy π, we take θ_h^π to be $Q^\pi(s^h, a^h) \cdot \phi(s^h, a^h)$. Now we show that $|Q^\pi(s, a) - \langle \theta_h^\pi, \phi(s, a) \rangle| \leq \delta$ for all states s in level h and $a \in \{a_1, a_2\}$.

Case I: $a \neq a^h$**.** In this case, we have $Q^\pi(s, a) = 0$ and $\langle \theta_h^\pi, \phi(s, a) \rangle = 0$, since θ_h^π and $\phi(s, a)$ do not have a common non-zero coordinate.

Case II: $a = a^h$ **and** $s \neq s^h$**.** In this case, by the second property of $\mathcal{P}$ in Lemma 5.4.2 and the fact that $Q^\pi(s^h, a^h) \leq 1$, we have $|\langle \theta_h^\pi, \phi(s, a) \rangle| \leq \delta$. Meanwhile, we have $Q^\pi(s, a) = 0$.

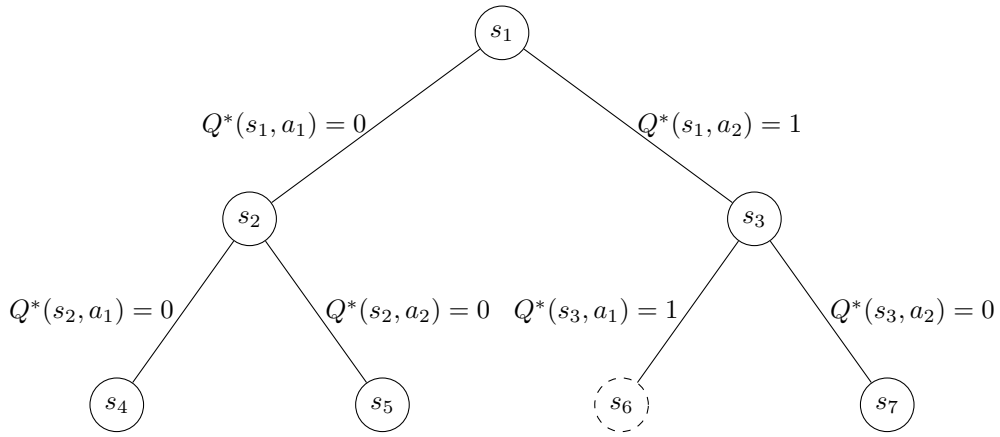


Figure 5.1: An example with $H = 3$. For this example, we have $\overline{r}(s_6) = 1$ and $\overline{r}(s) = 0$ for all other states s. The unique state s_6 which satisfies $\overline{r}(s) = 1$ is marked as dash in the figure. The induced Q^* function is marked on the edges.

Case III: $a = a^h$ and $s = s^h$. In this case, we have $\langle \theta_h^\pi, \phi(s, a) \rangle = Q^\pi(s^h, a^h)$.

Finally, we prove any algorithm that solves these MDP instances and succeeds with probability at least 0.9 needs to sample at least $\frac{9}{20} \cdot 2^H$ trajectories. We do so by providing a reduction from $\mathsf{INDQ}_{2^{H-1}}$ to solving MDPs. Suppose we have an algorithm for solving these MDPs, we show that such an algorithm can be transformed to solve $\mathsf{INDQ}_{2^{H-1}}$. For a specific choice of i^* in $\mathsf{INDQ}_{2^{H-1}}$, there is a corresponding MDP instance with

$$\overline{r}(s) = \begin{cases} 1 & \text{if } s = s_{i^* + 2^{H-1} - 1} \\ 0 & \text{otherwise} \end{cases}.$$

Notice that for all MDPs that we are considering, the transition and features are always the same. Thus, the only thing that the learner needs to learn by interacting with the environment is the reward value. Since the reward value is non-zero only for states in level H, each time the algorithm for solving MDP samples a trajectory that ends at state s_i where s_i is a state in level H, we query whether $i^* = i - 2^{H-1} + 1$ or not in $\mathsf{INDQ}_{2^{H-1}}$, and return reward value 1 if $i^* = i - 2^{H-1} + 1$ and 0 otherwise. If the algorithm is guaranteed to return a $1/2$-optimal policy, then it must be able to find i^*.

5.4.2 Proof of Lower Bound for Policy-Based Learning

In this section, we present our hardness results for linear policy learning. In order to prove Theoerem 5.2.2, we need the following geometric lemma.

Lemma 5.4.4. *Let $d \in \mathbb{N}_+$ be a positive integer and $\varepsilon \in (0, 1)$ be a real number. Then there exists a set of points $\mathcal{N} \subset SS^{d-1}$ with size $|\mathcal{N}| = \Omega(1/\varepsilon^{d/2})$ such that for every point $x \in \mathcal{N}$,*

$$\inf_{y \in \mathrm{conv}(\mathcal{N} \setminus \{x\})} \|x - y\|_2 \geq \varepsilon/2. \tag{5.1}$$

Proof. Consider a $\sqrt{\varepsilon}$-packing $\mathcal{N}$ with size $\Omega(1/\varepsilon^{d/2})$ on the d-dimensional unit sphere. For the existence of such a packing, see, e.g., [54]. Let o be the origin. For two points $x, x' \in \mathbb{R}^d$, we denote $|xx'| := \|x - x'\|_2$ the length of the line segment between x, x'. Note that every two points $x, x' \in \mathcal{N}$ satisfy $|xx'| \geq \sqrt{\varepsilon}$.

To prove the lemma, it suffices to show that $\mathcal{N}$ satisfies the property (5.1). Consider a point $x \in \mathcal{N}$, let A be a hyperplane that is perpendicular to x (notice that x is a also a vector) and separates x and every other points in $\mathcal{N}$. We let the distance between x and A be the largest possible, i.e., A contains a point in $\mathcal{N}\backslash\{x\}$. Since x is on the unit sphere and $\mathcal{N}$ is a $\sqrt{\varepsilon}$-packing, we have that x is at least $\sqrt{\varepsilon}$ away from every point on the spherical cap not containing x, defined by the cutting plane A. More formally, let b be the intersection point of the line segment ox and A. Then

$$\forall y \in \left\{y' \in SS^{d-s} : \langle b, y' \rangle \leq \|b\|_2^2\right\} : \quad \|x - y\|_2 \geq \sqrt{\varepsilon}.$$

Indeed, by symmetry, $\forall y \in \{y' \in SS^{d-1} : \langle b, y' \rangle \leq \|b\|_2^2\}$,

$$\|x - y\|_2 \geq \|x - z\|_2 \geq \sqrt{\varepsilon}.$$

where $z \in \mathcal{N} \cap A$. Notice that the distance between x and the convex hull of $\mathcal{N}\backslash\{x\}$ is lower bounded by the distance between x and A, which is given by $|bx|$. Consider the triangles defined by x, z, o, b. We have $bz \perp ox$ (note that bz lies inside A). By Pythagorean theorem, we have

$$\begin{aligned} |bz|^2 + |bx|^2 &= |xz|^2; \\ |bx| + |bo| &= |xo| = 1; \\ |bz|^2 + |bo|^2 &= |oz|^2 = 1. \end{aligned}$$

Solve the above three equations for $|bx|$, we have

$$|bx| = |xz|^2/2 \geq \varepsilon/2$$

as desired. □

Now we are ready to prove Theorem 5.2.2. In the proof we assume $H = d$, since otherwise we can take H and d to be $\min\{H, d\}$ by decreasing the planning horizon H or adding dummy dimensions to the feature extractor ϕ.

Proof of Theorem 5.2.2. We define a set of 2^{H-1} deterministic MDPs. The transitions of these hard instances are exactly the same as those in Section 5.4.1. The main difference is in the definition of the feature map $\phi(\cdot, \cdot)$ and the reward function. Again in the hard instances, $R(s, a) = 0$ for all s in the first $H - 2$ levels. Using the terminology in Section 5.4.1, we have $\overline{r}(s) = 0$ for all states in the first $H - 1$ levels. Now we define $\overline{r}(s)$ for states s in level H. We do so by recursively defining the optimal value function $V^*(\cdot)$. The initial state s_1 in level 1 satisfies $V^*(s_1) = 1/2$. For each state s_i in the first $H - 1$ levels, we have $V^*(s_{2i}) = V^*(s_i)$ and $V^*(s_{2i+1}) = V^*(s_i) - 1/2H$. For each state s_i in the level $h = H$, we have $\overline{r}(s_{2i}) = V^*(s_i)$ and $\overline{r}(s_{2i+1}) = V^*(s_i) - 1/2H$. This implies that $\rho = 1/2H$. In fact, this implies a stronger property that each state has a unique optimal action. See Figure 5.2 for an example with $H = 3$.

To define 2^{H-1} different MDPs, for each state s in level H of the MDP defined above, we define a new MDP by changing $\overline{r}(s)$ from its original value to 1. This also affects the definition of

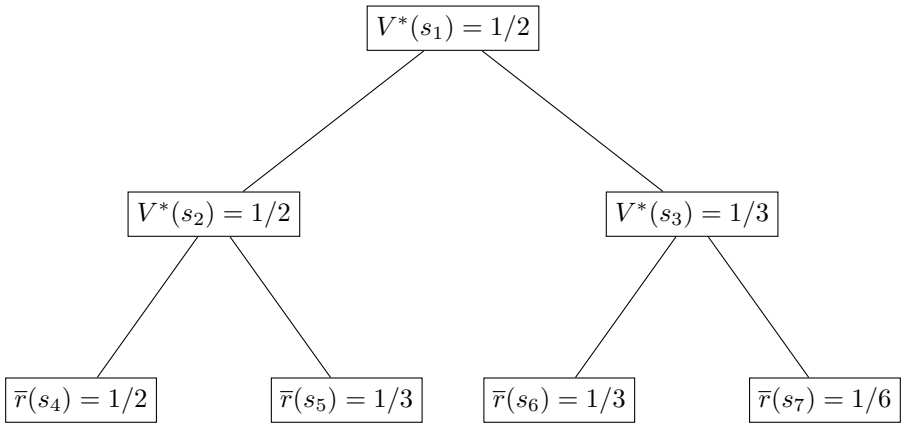


Figure 5.2: An example with $H = 3$.

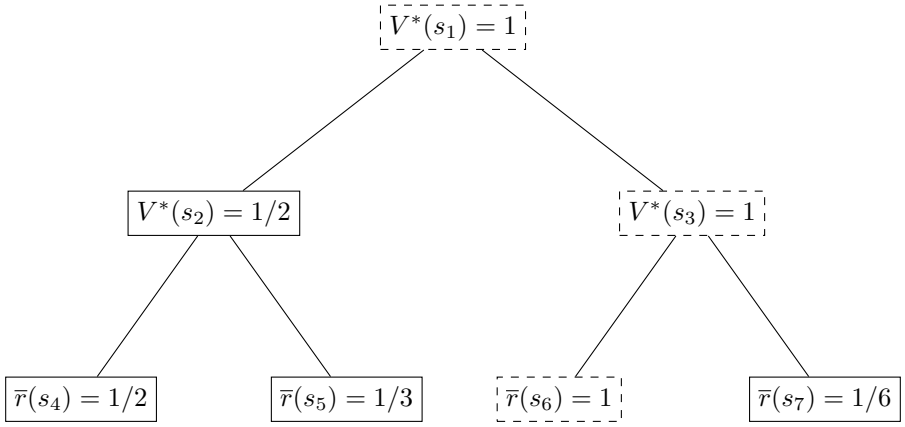


Figure 5.3: An example with $H = 3$. Here we define a new MDP by changing $\overline{r}(s_6)$ from its original value $1/3$ to 1. This also affects the value of $V(s_3)$ and $V(s_1)$.

the optimal V function for states in the first $H-1$ levels. In particular, for each level $i \in [H-1]$, we have changed the V^* value of a unique state in level i from its original value (at most $1/2$) to 1. By doing so we have defined 2^{H-1} different MDPs. See Figure 5.3 for an example with $H = 3$.

Now we define the feature map $\phi(\cdot, \cdot)$. We invoke Lemma 5.4.4 with $\varepsilon = 8\triangle$ and $d = H/2 - 1$. Since $\triangle$ is sufficiently small, we have $|\mathcal{N}| \geq 2^H$. We use $\mathcal{P} = \{p_1, p_2, \ldots, p_{2^H}\} \subset \mathbb{R}^{H/2-1}$ to denote an arbitrary subset of $\mathcal{N}$ with cardinality 2^H. By Lemma 5.4.4, for any $p \in \mathcal{P}$, the distance between p and the convex hull of $\mathcal{P} \setminus \{p\}$ is at least $4\triangle$. Thus, there exists a hyperplane L which separates p and $\mathcal{P} \setminus \{p\}$, and for all points $q \in \mathcal{P}$, the distance between q and L is at least $2\triangle$. Equivalently, for each point $p \in \mathcal{P}$, there exists $n_p \in \mathbb{R}^{H/2-1}$ and $o_p \in \mathbb{R}$ such that $\|n_p\|_2 = 1$, $|o_p| \leq 1$ and the linear function $f_p(q) = \langle q, n_p \rangle + o_p$ satisfies $f_p(p) \geq 2\triangle$ and $f_p(q) \leq -2\triangle$ for all $q \in \mathcal{P} \setminus \{p\}$. Given the set $\mathcal{P} = \{p_1, p_2, \ldots, p_{2^H}\} \subset \mathbb{R}^{H/2-1}$,

we construct a new set $\overline{\mathcal{P}} = \{\overline{p}_1, \overline{p}_2, \ldots, \overline{p}_{2^H}\} \subset \mathbb{R}^{H/2}$, where $\overline{p}_i = [p_i; 1] \in \mathbb{R}^{H/2}$. Thus $\|\overline{p}_i\|_2 = \sqrt{2}$ for all $\overline{p}_i \in \overline{\mathcal{P}}$. Clearly, for each $\overline{p} \in \overline{\mathcal{P}}$, there exists a vector $\omega_{\overline{p}} \in \mathbb{R}^{H/2}$ such that $\langle \omega_{\overline{p}}, \overline{p} \rangle \geq 2\triangle$ and $\langle \omega_{\overline{p}}, \overline{q} \rangle \leq -2\triangle$ for all $\overline{q} \in \overline{\mathcal{P}} \setminus \{\overline{p}\}$. It is also clear that $\|\omega_{\overline{p}}\|_2 \leq \sqrt{2}$. We take $\phi(s_i, a_1) = [0; \overline{p}_i] \in \mathbb{R}^H$ and $\phi(s_i, a_2) = [\overline{p}_i; 0] \in \mathbb{R}^H$.

We now show that all the 2^{H-1} MDPs constructed above satisfy Assumption 5.2.5. Namely, we show that for any state s in level H, after changing $\overline{r}(s)$ to be 1, the resulting MDP satisfies Assumption 5.2.5. As in Section 5.4.1, for each level $h \in [H]$, there is a unique state s^h in level h and action $a^h \in \{a_1, a_2\}$, such that $Q^*(s^h, a^h) = 1$. For each level h, if $a^h = a_1$, then we take $(\theta_h)_{H/2} = 1$ and $(\theta_h)_H = -1$, and all other entries in θ_h are zeros. If $a^h = a_2$, we use $\overline{p}$ to denote the vector formed by the first $H/2$ coordinates of $\phi(s^h, a_2)$. By construction, we have $\overline{p} \in \overline{\mathcal{P}}$. We take $\theta_h = [\omega_{\overline{p}}; 0]$ in this case. In any case, we have $\|\theta_h\|_2 \leq \sqrt{2}$. Now for each level h, if $a^h = a_1$, then for all states s in level h, we have $\pi^*(s) = a_1$. In this case, $\langle \phi(s, a_1), \theta_h \rangle = 1$ and $\langle \phi(s, a_2), \theta_h \rangle = -1$ for all states in level h, and thus Assumption 5.2.5 is satisfied. If $a^h = a_2$, then $\pi^*(s_h) = a_2$ and $\pi^*(s) = a_1$ for all states $s \neq s^h$ in level h. By construction, we have $\langle \theta_h, \phi(s, a_1) \rangle = 0$ for all states s in level h, since θ_h and $\phi(s, a_1)$ do not have a common non-zero entry. We also have $\langle \theta_h, \phi(s^h, a_2) \rangle \geq 2\triangle$ and $\langle \theta_h, \phi(s, a_2) \rangle \leq -2\triangle$ for all states $s \neq s^h$ in level h. We further normalize all θ_h and $\phi(s, a)$ so that they all have unit norm. Since $\|\phi(s, a)\|_2 = \sqrt{2}$ for all (s, a) pairs before normalization, Assumption 5.2.5 is still satisfied after normalization.

Finally, we prove any algorithm that solves these MDP instances and succeeds with probability at least 0.9 needs to sample at least $\Omega(2^H)$ trajectories. We do so by providing a reduction from $\mathsf{INDQ}_{2^{H-1}}$ to solving MDPs. Suppose we have an algorithm for solving these MDPs, we show that such an algorithm can be transformed to solve $\mathsf{INDQ}_{2^{H-1}}$. For a specific choice of i^* in $\mathsf{INDQ}_{2^{H-1}}$, there is a corresponding MDP instance with

$$\overline{r}(s) = \begin{cases} 1 & \text{if } s = s_{i^* + 2^{H-1} - 1} \\ \text{the original (recursively defined) value} & \text{otherwise} \end{cases}.$$

Notice that for all MDPs that we are considering, the transition and features are always the same. Thus, the only thing that the learner needs to learn by interacting with the environment is the reward value. Since the reward value is non-zero only for states in level H, each time the algorithm for solving MDP samples a trajectory that ends at state s_i where s_i is a state in level H, we query whether $i^* = i - 2^{H-1} + 1$ or not in $\mathsf{INDQ}_{2^{H-1}}$. If $i^* = i - 2^{H-1} + 1$, we return a reward value of 1, and return the original (recursively defined) reward value otherwise. If the algorithm is guaranteed to return a $1/4$-optimal policy, then it must be able to find i^*.

□

Chapter 6

RL with Large State Spaces: the Offline Setting

6.1 Introduction

Offline methods (also known as off-policy methods or batch methods) are a promising methodology to alleviate the sample complexity burden in challenging reinforcement learning (RL) settings, particularly those where sample efficiency is paramount [55, 88, 107]. Off-policy methods are often applied together with function approximation schemes; such methods take sample transition data and reward values as inputs, and approximate the value of a target policy or the value function of the optimal policy. Indeed, many practical deep RL algorithms find their prototypes in the literature of offline RL. For example, when running on off-policy data (sometimes termed as "experience replay"), deep Q-networks (DQN) [58] can be viewed as an analog of Fitted Q-Iteration [33] with neural networks being the function approximators. More recently, there are an increasing number of both model-free [32, 48, 50] and model-based [45, 71] offline RL methods, with steady improvements in performance [32, 45, 48, 99].

However, despite the importance of these methods, the extent to which data reuse is possible, especially when off-policy methods are combined with function approximation, is not well understood. For example, deep Q-network requires millions of samples to solve certain Atari games [58]. Also important is that in some safety-critical settings, we seek guarantees when offline-trained policies can be effective [84, 85]. A basic question here is that if there are fundamental statistical limits on such methods, where sample-efficient offline RL is simply not possible without further restrictions on the problem.

In supervised learning, it is well-known that empirical risk minimization is sample-efficient if the hypothesis class has bounded complexity. For example, suppose the agent is given a d-dimensional feature extractor, and the ground truth labeling function is a (realizable) linear function with respect to the feature mapping. Here, it is well-known that a polynomial number of samples in d suffice for a given target accuracy. Furthermore, in this realizable case, provided the training data has a good feature coverage, then we will have good accuracy against any test distribution.

In the more challenging offline RL setting, it is unclear if sample-efficient methods are pos-

sible, even under analogous assumptions. Here, one may hope that value estimation for a given policy is possible in the offline RL setting under the analogous set of assumptions that enable sample-efficient supervised learning, i.e., 1) (realizability) the features can perfectly represent the value functions and 2) (good coverage) the feature covariance matrix of our off-policy data has lower bounded eigenvalues.

The extant body of provable methods on offline RL either make representational assumptions that are far stronger than realizability or assume distribution shift conditions that are far stronger than having coverage with regards to the spectrum of the feature covariance matrix of the data distribution. For example, [62] analyze offline RL methods by assuming a representational condition where the features satisfy (approximate) closedness under Bellman updates, which is a far stronger representation condition than realizability. Recently, [100] propose a offline RL algorithm that only requires realizability as the representation condition. However, the algorithm in [100] requires a more stringent data distribution condition. Whether it is possible to design a sample-efficient offline RL method under the realizability assumption and a reasonable data coverage assumption — an open problem in [16] — is the focus of this section.

Theoretical Results. Perhaps surprisingly, our main result shows that, under only the above two assumptions, it is information-theoretically not possible to design a sample-efficient algorithm to non-trivially estimate the value of a given policy. The following theorem is an informal version of the result in Section 6.3.

Theorem 6.1.1 (Informal)**.** *In the offline setting, suppose the data distributions have (polynomially) lower bounded eigenvalues, and the Q-functions of* every *policy are linear with respect to a given feature mapping. Any algorithm requires an exponential number of samples in the horizon H to output a non-trivially accurate estimate of the value of* any *given policy π, with constant probability.*

This hardness result states that even if the Q-functions of *all* polices are linear with respect to the given feature mapping, we still require an exponential number of samples to evaluate *any* policy. Note that this representation condition is significantly stronger than assuming realizability with regards to a single target policy; it assumes realizability for all policies. Even under this stronger representation condition, it is hard to evaluate any policy, as specified in our hardness result.

This result also formalizes a key issue in offline reinforcement learning with function approximation: geometric error amplification. To better illustrate the issue, in Section 6.4, we analyze the classical Least-Squares Policy Evaluation (LSPE) algorithm under the realizability assumption, which demonstrates how the error propagates as the algorithm proceeds. Here, our analysis shows that, if we only rely on the realizability assumption, then a far more stringent condition is required for sample-efficient offline policy evaluation: the off-policy data distribution must be quite close to the distribution induced by the policy to be evaluated.

Our results highlight that sample-efficient offline RL is simply not possible unless either the distribution shift condition is sufficiently mild or we have stronger representation conditions that go well beyond realizability. See Section 6.4 for more details.

Experiments. From a practical point of view, it is natural to ask to what extent the above worst-case characterizations are reflective of the scenarios that arise in practical applications because, in fact, modern deep learning methods often produce representations that are extremely effective, say for transfer learning (computer vision [105] and NLP [20, 64] have both witnessed remarkable successes using pre-trained features on downstreams tasks of interest). Furthermore, there are number of offline RL methods with promising performance on certain benchmark tasks [32, 45, 48, 50, 71, 99].

In this chapter we provide a careful empirical investigation to further understand how sensitive offline RL methods are to distribution shift. Along this line of inquiry, one specific question to answer is to what extent we should be concerned about the error amplification effects as suggested by worst-case theoretical considerations.

We study these questions on a range of standard tasks (6 tasks from the OpenAI gym benchmark suite), using offline datasets with features from pre-trained neural networks trained on the task itself. Our offline datasets are a mixture of trajectories from the target policy itself, along the data from other policies (random or lower performance policies). Note that this is favorable setting in that we would not expect realistic offline datasets to have a large number of trajectories from the target policy itself.

The motivation for using pre-trained features are both conceptual and technical. First, we may hope that such features are powerful enough to permit sample-efficient offline RL because they were learned in an online manner on the task itself. Also, practically, while we are not able to verify if certain theoretical assumptions hold, we may optimistically hope that such pre-trained features will perform well under distribution shift (indeed, as discussed earlier, using pre-trained features has had remarkable successes in other domains). Second, using pre-trained features allows us to decouple practical representational learning questions from the offline RL question, where we can focus on offline RL with a given representation.

The main conclusion of this chapter, through extensive experiments on a number of tasks, is that: *we do in fact observe substantial error amplification*, even when using pre-trained representations, even we tune hyper-parameters, regardless of what the distribution was shifted to; furthermore, this amplification even occurs under relatively mild distribution shift.

These experiments also complement our theoretical results showing the issue of error amplification is a real practical concern. From a practical point of view, our experiments demonstrate that the definition of a good representation is more subtle than in supervised learning.

6.2 Preliminaries

Episodic Reinforcement Learning. In this section, for simplicity, we assume a fixed initial state $s_1 \in \mathcal{S}$. To streamline our analysis, for each $h \in [H]$, we use $\mathcal{S}_h \subseteq \mathcal{S}$ to denote the set of states at level h, and we assume $\mathcal{S}_h$ do not intersect with each other. We assume, almost surely, that $r_h \in [-1, 1]$ for all $h \in [H]$.

Linear Function Approximation. When applying linear function approximation schemes, it is commonly assumed that the agent is given a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ which can

either be hand-crafted or a pre-trained neural network that transforms a state-action pair to a d-dimensional embedding, and the Q-functions can be predicted by linear functions of the features. In this section, we are interested in the following *realizability* assumption.

Assumption 6.2.1 (Realizable Linear Function Approximation)**.** *For every policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, there exists $\theta_1^\pi, \dots \theta_H^\pi \in \mathbb{R}^d$ such that for all $h \in [H]$ and $(s,a) \in \mathcal{S}_h \times \mathcal{A}$, $Q_h^\pi(s,a) = (\theta_h^\pi)^\top \phi(s,a)$.*

Note that our assumption is much stronger than assuming realizability with regards to a single policy π (say the policy that we wish to evaluate); our assumption imposes realizability for all policies.

6.3 The Lower Bound: Realizability and Coverage are Insufficient

We now present our main hardness result for offline policy evaluation with linear function approximation. It should be evident that without feature coverage in our dataset, realizability alone is clearly not sufficient for sample-efficient estimation. Here, we will make the strongest possible assumption, with regards to the conditioning of the feature covariance matrix.

Assumption 6.3.1 (Feature Coverage)**.** *For all $(s,a) \in \mathcal{S} \times \mathcal{A}$, assume our feature map is bounded such that $\|\phi(s,a)\|_2 \leq 1$. Furthermore, suppose for each $h \in [H]$, the data distributions μ_h satisfy the following minimum eigenvalue condition:*

$$\sigma_{\min}\left(\mathbb{E}_{(s,a)\sim\mu_h}[\phi(s,a)\phi(s,a)^\top]\right) = 1/d.^1$$

Clearly, for the case where $H = 1$, the realizability assumption (Assumption 6.2.1), and feature coverage assumption (Assumption 6.3.1) imply that the ordinary least squares estimator will accurately estimate θ_1.[2] Our main result now shows that these assumptions are not sufficient for offline policy evaluation for long horizon problems.

Theorem 6.3.1. *Suppose Assumption 6.3.1 holds. Fix an algorithm that takes as input both a policy and a feature mapping. There exists a (deterministic) MDP satisfying Assumption 6.2.1, such that for* any *policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, the algorithm requires $\Omega((d/2)^H)$ samples to output the value of π up to constant additive approximation error with probability at least 0.9.*

Remark 6.3.1 (Least-Squares Policy Evaluation (LSPE) has exponential variance)**.** For offline policy evaluation with linear function approximation, the most naïve algorithm here would be LSPE, i.e., using ordinary least squares (OLS) to estimate θ^π, starting at level $h = H$ and then proceeding backwards to level $h = 1$, using the plug-in estimator from the previous level. Here, LSPE will provide an unbiased estimate (provided the feature covariance matrices are full rank, which will occur with high probability). As a direct corollary, the above theorem implies that LSPE has exponential variance in H. See Section 6.4 for a more detailed discussion on LSPE.

[1] Note that $1/d$ is the largest possible minimum eigenvalue due to that, for any data distribution $\widetilde{\mu}_h$, $\sigma_{\min}(\mathbb{E}_{(s,a)\sim\widetilde{\mu}_h}[\phi(s,a)\phi(s,a)^\top]) \leq 1/d$ since $\|\phi(s,a)\|_2 \leq 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

[2] For $H = 1$, the ordinary least squares estimator will satisfy that $\|\theta_1 - \widehat{\theta}_{\text{OLS}}\|_2^2 \leq O(d/n)$ with high probability. See e.g. [35].

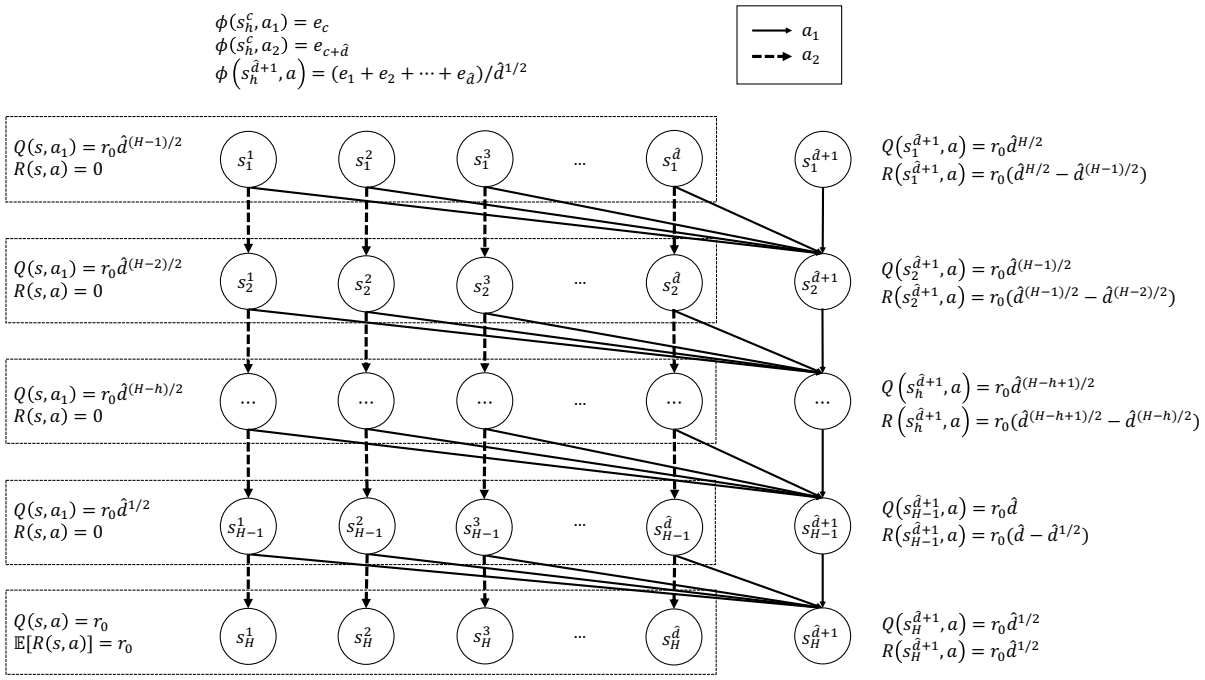


Figure 6.1: An illustration of the hard instance. Recall that $\widehat{d} = d/2$. States on the top are those in the first level ($h = 1$), while states at the bottom are those in the last level ($h = H$). Solid line (with arrow) corresponds to transitions associated with action a_1, while dotted line (with arrow) corresponds to transitions associated with action a_2. For each level $h \in [H]$, reward values and Q-values associated with $s_h^1, s_h^2, \ldots, s_h^{\widehat{d}}$ are marked on the left, while reward values and Q-values associated with $s_h^{\widehat{d}+1}$ are mark on the right. Rewards and transitions are all deterministic, except for the reward distributions associated with $s_H^1, s_H^2, \ldots, s_H^{\widehat{d}}$. We mark the expectation of the reward value when it is stochastic. For each level $h \in [H]$, for the data distribution μ_h, the state is chosen uniformly at random from those states in the dashed rectangle, i.e., $\{s_h^1, s_h^2, \ldots, s_h^{\widehat{d}}\}$, while the action is chosen uniformly at random from $\{a_1, a_2\}$. Suppose the initial state is $s_1^{\widehat{d}+1}$. When $r_0 = 0$, the value of the policy is 0. When $r_0 = \widehat{d}^{-H/2}$, the value of the policy is $r_0 \cdot \widehat{d}^{H/2} = 1$.

More generally, our theorem implies that there is no estimator that can avoid such exponential dependence in the offline setting.

In the remaining part of this section, we give the hard instance construction and the proof of Theorem 6.3.1. We use d the denote the feature dimension, and we assume d is even for simplicity. We use $\widehat{d}$ to denote $d/2$ for convenience. We also provide an illustration of the construction in Figure 6.1.

State Space, Action Space and Transition Operator. The action space $\mathcal{A} = \{a_1, a_2\}$. For each $h \in [H]$, $\mathcal{S}_h$ contains $\widehat{d} + 1$ states $s_h^1, s_h^2, \ldots, s_h^{\widehat{d}}$ and $s_h^{\widehat{d}+1}$. For each $h \in [H-1]$, for each $c \in \{1, 2, \ldots, \widehat{d} + 1\}$, we have $P(s_h^c, a_1) = s_{h+1}^{\widehat{d}+1}$ and $P(s_h^c, a_1) = s_{h+1}^c$.

Reward Distributions. Let $0 \le r_0 \le \widehat{d}^{-H/2}$ be a parameter to be determined. For each $(h,c) \in [H-1]\times[\widehat{d}]$ and $a \in \mathcal{A}$, we set $R(s_h^c, a) = 0$ and $R(s_h^{\widehat{d}+1}, a) = r_0 \cdot (\widehat{d}^{1/2}-1)\cdot \widehat{d}^{(H-h)/2}$. For the last level, for each $c \in [\widehat{d}]$ and $a \in \mathcal{A}$, we set $R(s_H^c, a) = \begin{cases} 1 & \text{with probability } (1+r_0)/2 \\ -1 & \text{with probability } (1-r_0)/2 \end{cases}$ so that $\mathbb{E}[R(s_H^c, a)] = r_0$. Moreover, for all actions $a \in \mathcal{A}$, $R(s_H^{\widehat{d}+1}, a) = r_0 \cdot \widehat{d}^{1/2}$.

Feature Mapping. Let $e_1, e_2, \ldots, e_d$ be a set of orthonormal vectors in $\mathbb{R}^d$. Here, one possible choice is to set $e_1, e_2, \ldots, e_d$ to be the standard basis vectors. For each $(h,c) \in [H] \times [\widehat{d}]$, we set $\phi(s_h^c, a_1) = e_c$, $\phi(s_h^c, a_2) = e_{c+\widehat{d}}$, and $\phi(s_h^{\widehat{d}+1}, a) = \sum_{c\in\widehat{d}} e_c/\widehat{d}^{1/2}$ for all $a \in \mathcal{A}$.

Verifying Assumption 6.2.1. The following lemma shows that Assumption 6.2.1 holds for our construction.

Lemma 6.3.2. *For every policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, for each $h \in [H]$, for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$, we have $Q_h^\pi(s,a) = (\theta_h^\pi)^\top \phi(s,a)$ for some $\theta_h^\pi \in \mathbb{R}^d$.*

Proof. We first verify Q^π is linear for the first $H-1$ levels. For each $(h,c) \in [H-1] \times [\widehat{d}]$, we have

$$Q_h^\pi(s_h^c, a_1) = R(s_h^c, a_1) + R(s_{h+1}^{\widehat{d}+1}, a_1) + R(s_{h+2}^{\widehat{d}+1}, a_1) + \ldots + R(s_H^{\widehat{d}+1}, a_1) = r_0 \cdot \widehat{d}^{(H-h)/2}.$$

Moreover, for all $a \in \mathcal{A}$,

$$Q_h^\pi(s_h^{\widehat{d}+1}, a) = R(s_h^{\widehat{d}+1}, a) + R(s_{h+1}^{\widehat{d}+1}, a_1) + R(s_{h+2}^{\widehat{d}+1}, a_1) + \ldots + R(s_H^{\widehat{d}+1}, a_1) = r_0 \cdot \widehat{d}^{(H-h+1)/2}.$$

Therefore, if we define

$$\theta_h^\pi = \sum_{c=1}^{\widehat{d}} r_0 \cdot \widehat{d}^{(H-h)/2} \cdot e_c + \sum_{c=1}^{\widehat{d}} Q_h^\pi(s_h^c, a_2) \cdot e_{c+\widehat{d}},$$

then $Q_h^\pi(s,a) = (\theta_h^\pi)^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$.

Now we verify that the Q-function is linear for the last level. Clearly, for all $c \in [\widehat{d}]$ and $a \in \mathcal{A}$, $Q_H^\pi(s_H^c, a) = r_0$ and $Q_H^\pi(s_H^{\widehat{d}+1}, a) = r_0 \cdot \sqrt{\widehat{d}}$. Thus by defining $\theta_H^\pi = \sum_{c=1}^d r_0 \cdot e_c$, we have $Q_H^\pi(s,a) = (\theta_H^\pi)^\top \phi(s,a)$ for all $(s,a) \in \mathcal{S}_H \times \mathcal{A}$.

□

The Data Distributions. For each level $h \in [H]$, the data distribution μ_h is a uniform distribution over $\{(s_h^1, a_1), (s_h^1, a_2), (s_h^2, a_1), (s_h^2, a_2), \ldots, (s_h^{\widehat{d}}, a_1), (s_h^{\widehat{d}}, a_2)\}$. Notice that $(s_h^{\widehat{d}+1}, a)$ is *not* in the support of μ_h for all $a \in \mathcal{A}$. It can be seen that, $\mathbb{E}_{(s,a)\sim\mu_h}\left[\phi(s,a)\phi(s,a)^\top\right] = \frac{1}{d}\sum_{c=1}^d e_c e_c^\top = \frac{1}{d} I$.

The Lower Bound. We show that it is information-theoretically hard for any algorithm to distinguish the case $r_0 = 0$ and $r_0 = \widehat{d}^{-H/2}$. We fix the initial state to be $s_1^{\widehat{d}+1}$, and consider any policy π. When $r_0 = 0$, all reward values will be zero, and thus the value of π is zero. On the other hand, when $r_0 = \widehat{d}^{-H/2}$, the value of π would be $r_0 \cdot \widehat{d}^{H/2} = 1$. Thus, if the algorithm approximates the value of the policy up to an error of $1/2$, then it must distinguish the case that $r_0 = 0$ and $r_0 = \widehat{d}^{-H/2}$.

We first notice that for the case $r_0 = 0$ and $r_0 = \widehat{d}^{-H/2}$, the data distributions $\{\mu_h\}_{h=1}^H$, the feature mapping $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, the policy π to be evaluated and the transition operator P are the same. Thus, in order to distinguish the case $r_0 = 0$ and $r_0 = \widehat{d}^{-H/2}$, the only way is to query the reward distribution by using sampling taken from the data distributions. For all state-action pairs (s, a) in the support of the data distributions of the first $H-1$ levels, the reward distributions will be identical. This is because for all $s \in \mathcal{S}_h \setminus \{s_h^{\widehat{d}+1}\}$ and $a \in \mathcal{A}$, we have $R(s, a) = 0$. For the case $r_0 = 0$ and $r_0 = \widehat{d}^{-H/2}$, for all state-action pairs (s, a) in the support of the data distribution of the last level, $R(s,a) = \begin{cases} 1 & \text{with probability } (1 + r_0)/2 \\ -1 & \text{with probability } (1 - r_0)/2 \end{cases}$. Therefore, to distinguish the case that $r_0 = 0$ and $r_0 = \widehat{d}^{-H/2}$, the agent needs to distinguish two reward distributions $r_1 = \begin{cases} 1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases}$ and $r_2 = \begin{cases} 1 & \text{with probability } (1 + \widehat{d}^{-H/2})/2 \\ -1 & \text{with probability } (1 - \widehat{d}^{-H/2})/2 \end{cases}$. In order to distinguish r_1 and r_2 with probability at least 0.9, any algorithm requires $\Omega(\widehat{d}^H)$ samples.

Remark 6.3.2. The key in our construction is the state $s_h^{\widehat{d}+1}$ in each level, whose feature vector is defined to be $\sum_{c \in \widehat{d}} e_c / \widehat{d}^{1/2}$. In each level, $s_h^{\widehat{d}+1}$ amplifies the Q-value by a $\widehat{d}^{1/2}$ factor, due to the linearity of the Q-function. After all the H levels, the value will be amplified by a $\widehat{d}^{H/2}$ factor. Since $s_h^{\widehat{d}+1}$ is not in the support of the data distribution, the only way to estimate the value of the policy is to estimate the expected reward value in the last level. Our construction forces the estimation error of the last level to be amplified exponentially and thus implies an exponential lower bound.

6.4 Upper Bounds: Low Distribution Shift or Policy Completeness are Sufficient

In order to illustrate the error amplification issue and discuss conditions that permit sample-efficient offline RL, in this section, we analyze Least-Squares Policy Evaluation when applied to the offline policy evaluation problem under the realizability assumption. The algorithm is presented in Algorithm 10. For simplicity here we assume the policy π to be evaluated is deterministic.

Notation. For each $h \in [H]$, define $\Lambda_h = \mathbb{E}_{(s,a)\sim\mu_h}\left[\phi(s,a)\phi(s,a)^\top\right]$ to be the feature covariance matrix of the data distribution at level h. Moreover, for each $h \in [H-1]$, define $\overline{\Lambda}_{h+1} = \mathbb{E}_{(s,a)\sim\mu_h, \overline{s}\sim P(\cdot|s,a)}\left[\phi(\overline{s}, \pi(\overline{s}))\phi(\overline{s}, \pi(\overline{s}))^\top\right]$ to be the feature covariance matrix of the one-step lookahead distribution induced by the data distribution at level h and π. Moreover,

Algorithm 10 Least-Squares Policy Evaluation

Input: policy π to be evaluated, number of samples N, regularization parameter $\lambda > 0$
Let $Q_{H+1}(\cdot,\cdot) = 0$ and $V_{H+1}(\cdot) = 0$
for $h = H, H-1, \ldots, 1$ **do**
    Take samples $(s_h^i, a_h^i) \sim \mu_h$, $r_h^i \sim R(s_h^i, a_h^i)$ and $\overline{s}_h^i \sim P(s_h^i, a_h^i)$ for each $i \in [N]$
    Let $\widehat{\Lambda}_h = \sum_{i\in[N]} \phi(s_h^i, a_h^i)\phi(s_h^i, a_h^i)^\top + \lambda I$
    Let $\widehat{\theta}^h = \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \widehat{V}_{h+1}(\overline{s}_h^i))\right)$
    Let $\widehat{Q}_h(\cdot,\cdot) = \phi(\cdot,\cdot)^\top\widehat{\theta}_h$ and $\widehat{V}_h(\cdot) = \widehat{Q}_h(\cdot, \pi(\cdot))$

define $\overline{\Lambda}_1 = \phi(s_1, \pi(s_1))\phi(s_1, \pi(s_1))^\top$. We define Φ_h to be a $N \times d$ matrix, whose i-th row is $\phi(s_h^i, a_h^i)$, and define $\overline{\Phi}_{h+1}$ to be another $N \times d$ matrix whose i-th row is $\phi(\overline{s}_h^i, \pi(\overline{s}_h^i))$. For each $h \in [H]$ and $i \in [N]$, define $\xi_h^i = r_h^i + V_h(\overline{s}_h^i) - Q_h(s_h^i, a_h^i)$. We use ξ_h to denote a vector whose i-th entry is ξ_h^i.

Now we present a general lemma that characterizes the estimation error of Algorithm 10 by an *equality*. Later, we apply this general lemma to special cases.

Lemma 6.4.1. *Suppose $\lambda > 0$ in Algorithm 10, and for the given policy π, there exists $\theta_1, \theta_2, \ldots, \theta_d \in \mathbb{R}^d$ such that for each $h \in [H]$, $Q_h^\pi(s,a) = \phi(s,a)^\top\theta_h$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$. Then we have*

$$(Q^\pi(s_1, \pi(s_1)) - \widehat{Q}(s_1, \pi(s_1)))^2 = \left\| \sum_{h=1}^H \widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\widehat{\Lambda}_2^{-1}\Phi_2^\top \cdots (\widehat{\Lambda}_h^{-1}\Phi_h^\top\xi_h - \lambda\widehat{\Lambda}_h^{-1}\theta_h) \right\|_{\overline{\Lambda}_1}^2 . \quad (6.1)$$

Proof. Clearly,

$$\begin{aligned}
\widehat{\theta}_h &= \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \widehat{V}_{h+1}(\overline{s}_h^i))\right) \\
&= \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \widehat{Q}_{h+1}(\overline{s}_h^i, \pi(\overline{s}_h^i)))\right) \\
&= \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\widehat{\theta}_{h+1})\right) \\
&= \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\theta_{h+1}) + \sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top(\widehat{\theta}_{h+1} - \theta_{h+1})\right) \\
&= \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot(r_h^i + \phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top\theta_{h+1})\right) + \widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^N \phi(s_h^i, a_h^i)\cdot\phi(\overline{s}_h^i, \pi(\overline{s}_h^i))^\top(\widehat{\theta}_{h+1} - \theta_{h+1})\right).
\end{aligned}$$

For the first term, we have

$$
\begin{aligned}
&\widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot(r_h^i+\phi(\overline{s}_h^i,\pi(\overline{s}_h^i))^\top\theta_{h+1})\right)\\
=&\widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot(r_h^i+Q^\pi(\overline{s}_h^i,\pi(\overline{s}_h^i)))\right)\\
=&\widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot(r_h^i+V^\pi(\overline{s}_h^i))\right)\\
=&\widehat{\Lambda}_h^{-1}\left(\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot(Q^\pi(s_h^i,a_h^i)+\xi_h^i)\right)\\
=&\widehat{\Lambda}_h^{-1}\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot\xi_h^i+\widehat{\Lambda}_h^{-1}\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot\phi(s_h^i,a_h^i)^\top\theta_h\\
=&\widehat{\Lambda}_h^{-1}\sum_{i=1}^{N}\phi(s_h^i,a_h^i)\cdot\xi_h^i+\widehat{\Lambda}_h^{-1}(\Phi_h^\top\Phi_h)\theta_h\\
=&\widehat{\Lambda}_h^{-1}\Phi_h\xi_h+\theta_h-\lambda\widehat{\Lambda}_h^{-1}\theta_h.
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\widehat{\theta}_1-\theta_1&=(\widehat{\Lambda}_1^{-1}\Phi_1\xi_1-\lambda\widehat{\Lambda}_1^{-1}\theta_1)+\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2(\theta_2-\widehat{\theta}_2)\\
&=(\widehat{\Lambda}_1^{-1}\Phi_1\xi_1-\lambda\widehat{\Lambda}_1^{-1}\theta_1)+\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2(\widehat{\Lambda}_2^{-1}\Phi_2^\top\xi_2-\lambda\widehat{\Lambda}_2^{-1}\theta_2)\\
&\quad+\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\widehat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3(\theta_3-\widehat{\theta}_3)\\
&=\dots\\
&=\sum_{h=1}^{H}\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\widehat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3\cdots(\widehat{\Lambda}_h^{-1}\Phi_h^\top\xi_h-\lambda\widehat{\Lambda}_h^{-1}\theta_h).
\end{aligned}
$$

Also note that

$$
(Q^\pi(s_1,\pi(s_1))-\widehat{Q}(s_1,\pi(s_1)))^2=\|\theta_1-\widehat{\theta}_1\|_{\overline{\Lambda}_1}^2.
$$

□

Now we consider two special cases where the estimation error in Equation (6.1) can be upper bounded.

Low Distribution Shift. The first special we focus on is the case where the distribution shift between the data distributions and the distribution induced by the policy to be evaluated is low. To measure the distribution shift formally, our main assumption is as follows.

Assumption 6.4.1. *We assume that for each $h\in[H]$, there exists $C_h\geq 1$ such that $\overline{\Lambda_h}\preceq C_h\Lambda_h$.*

Remark 6.4.1. For each $h \in [H]$, if $\sigma_{\min}(\Lambda_h) \succeq \frac{1}{C_h} I$ for some $C_h \geq 1$, then we have $\overline{\Lambda}_h \preceq I \preceq C_h \Lambda_h$. Therefore, Assumption 6.4.1 can be replaced with the assumption that $C_h \Lambda_h \succeq I$. However, we stick to the original version of Assumption 6.4.1, since it gives a tighter characterization of the distribution shift when applying Algorithm 10 to off-policy evaluation under the realizability assumption.

Now we state the theoretical guarantee of Algorithm 10.

Theorem 6.4.2. *Suppose for the given policy π, there exists $\theta_1, \theta_2, \ldots, \theta_d \in \mathbb{R}^d$ such that for each $h \in [H]$, $Q_h^\pi(s,a) = \phi(s,a)^\top \theta_h$ for all $(s,a) \in \mathcal{S}_h \times \mathcal{A}$ and $\|\theta_h\|_2 \leq H\sqrt{d}$.[3] Let $\lambda = CH\sqrt{d\log(dH/\delta)N}$ for some $C > 0$. With probability at least $1-\delta$, for some $c > 0$,*

$$(Q_1^\pi(s_1, \pi(s_1)) - \widehat{Q}_1(s_1, \pi(s_1)))^2 \leq c \cdot \prod_{h=1}^{H} C_h \cdot dH^5 \cdot \sqrt{d\log(dH/\delta)/N}.$$

Proof. By matrix concentration inequality [86], we have the following lemma.

Lemma 6.4.3. *For each $h \in [H]$, with probability $1 - \delta/(4H)$, for some universal constant C, we have*

$$\left\| \frac{1}{N} \Phi_h^\top \Phi_h - \Lambda_h \right\|_2 \leq C\sqrt{d\log(dH/\delta)/N}.$$

and

$$\left\| \frac{1}{N} \overline{\Phi}_{h+1}^\top \overline{\Phi}_{h+1} - \overline{\Lambda}_{h+1} \right\|_2 \leq C\sqrt{d\log(dH/\delta)/N}.$$

Therefore, since $\lambda = CH\sqrt{d\log(dH/\delta)N}$, with probability $1-\delta/(4H)$, we have

$$\widehat{\Lambda}_h = \Phi_h^\top \Phi_h + \lambda I \succeq N\Lambda_h.$$

Note that

$$\begin{aligned}
&(Q^\pi(s_1, \pi(s_1)) - \widehat{Q}(s_1, \pi(s_1)))^2 \\
\leq& H \cdot \left(\sum_{h=1}^{H} \left\| \widehat{\Lambda}_1^{-1} \Phi_1^\top \overline{\Phi}_2 \widehat{\Lambda}_2^{-1} \Phi_2^\top \overline{\Phi}_3 \cdots (\widehat{\Lambda}_h^{-1} \Phi_h^\top \xi_h - \lambda \widehat{\Lambda}_h^{-1} \theta_h) \right\|_{\overline{\Lambda}_1}^2 \right) \\
\leq& 2H \cdot \left(\sum_{h=1}^{H} \left\| \widehat{\Lambda}_1^{-1} \Phi_1^\top \overline{\Phi}_2 \widehat{\Lambda}_2^{-1} \Phi_2^\top \overline{\Phi}_3 \cdots \widehat{\Lambda}_h^{-1} \Phi_h^\top \xi_h \right\|_{\overline{\Lambda}_1}^2 + \sum_{h=1}^{H} \left\| \widehat{\Lambda}_1^{-1} \Phi_1^\top \overline{\Phi}_2 \widehat{\Lambda}_2^{-1} \Phi_2^\top \overline{\Phi}_3 \cdots \lambda \widehat{\Lambda}_h^{-1} \theta_h \right\|_{\overline{\Lambda}_1}^2 \right).
\end{aligned}$$

For each $h \in [H]$,

$$\begin{aligned}
&\|\widehat{\Lambda}_1^{-1} \Phi_1^\top \overline{\Phi}_2 \widehat{\Lambda}_2^{-1} \Phi_2^\top \overline{\Phi}_3 \cdots \widehat{\Lambda}_h^{-1} \Phi_h^\top \xi_h\|_{\overline{\Lambda}_1}^2 \\
\leq& \|\Phi_1 \widehat{\Lambda}_1^{-1} \overline{\Lambda}_1 \widehat{\Lambda}_1^{-1} \Phi_1^\top\|_2 \cdot \|\overline{\Phi}_2 \widehat{\Lambda}_2^{-1} \Phi_2^\top \overline{\Phi}_3 \cdots \widehat{\Lambda}_h^{-1} \Phi_h^\top \xi_h\|_2^2 \\
\leq& \|\widehat{\Lambda}_1^{-1/2} \overline{\Lambda}_1 \widehat{\Lambda}_1^{-1/2}\|_2 \cdot \|\Phi_1 \widehat{\Lambda}_1^{-1} \Phi_1^\top\|_2 \cdot \|\overline{\Phi}_2 \widehat{\Lambda}_2^{-1} \Phi_2^\top \overline{\Phi}_2 \cdots \widehat{\Lambda}_h^{-1} \Phi_h^\top \xi_h\|_2^2 \\
\leq& \|\widehat{\Lambda}_1^{-1/2} \overline{\Lambda}_1 \widehat{\Lambda}_1^{-1/2}\|_2 \cdot \prod_{h'=1}^{h-1} \left(\|\Phi_{h'} \widehat{\Lambda}_{h'}^{-1} \Phi_{h'}^\top\|_2 \cdot \|\widehat{\Lambda}_{h'+1}^{-1/2} (\overline{\Phi}_{h'+1}^\top \overline{\Phi}_{h'+1}) \widehat{\Lambda}_{h'+1}^{-1/2}\|_2 \right) \cdot \|\xi_h\|_{\Phi_h \widehat{\Lambda}_h^{-1} \Phi_h^\top}^2 \cdot
\end{aligned}$$

[3] Without loss of generality, we can work in a coordinate system such that $\|\theta_h\|_2 \leq H\sqrt{d}$ and $\|\phi(s,a)\|_2 \leq 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. This follows due to John's theorem (e.g. see [14]).

Similarly,

$$\begin{aligned}
&\|\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\widehat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3\cdots\lambda\widehat{\Lambda}_h^{-1}\theta_h\|_{\overline{\Lambda}_1}^2\\
\leq&\|\widehat{\Lambda}_1^{-1/2}\overline{\Lambda}_1\widehat{\Lambda}_1^{-1/2}\|_2\cdot\prod_{h'=1}^{h-1}\left(\|\Phi_{h'}\widehat{\Lambda}_{h'}^{-1}\Phi_{h'}^\top\|_2\cdot\|\widehat{\Lambda}_{h'+1}^{-1/2}(\overline{\Phi}_{h'+1}^\top\overline{\Phi}_{h'+1})\widehat{\Lambda}_{h+1}^{-1/2}\|_2\right)\cdot\lambda^2\cdot\|\theta_h\|_{\widehat{\Lambda}_h^{-1}}^2\\
\leq&\|\widehat{\Lambda}_1^{-1/2}\overline{\Lambda}_1\widehat{\Lambda}_1^{-1/2}\|_2\cdot\prod_{h'=1}^{h-1}\left(\|\Phi_{h'}\widehat{\Lambda}_{h'}^{-1}\Phi_{h'}^\top\|_2\cdot\|\widehat{\Lambda}_{h'+1}^{-1/2}(\overline{\Phi}_{h'+1}^\top\overline{\Phi}_{h'+1})\widehat{\Lambda}_{h'+1}^{-1/2}\|_2\right)\cdot\lambda\cdot H^2d.
\end{aligned}$$

For all $h\in[H]$, we have

$$\|\Phi_h\widehat{\Lambda}_h^{-1}\Phi_h^\top\|_2\leq 1$$

and

$$\|\widehat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top\overline{\Phi}_h)\widehat{\Lambda}_h^{-1/2}\|_2\leq\|N\widehat{\Lambda}_h^{-1/2}\overline{\Lambda}_h\widehat{\Lambda}_h^{-1/2}\|_2+\|\widehat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top\overline{\Phi}_h-N\overline{\Lambda}_h)\widehat{\Lambda}_h^{-1/2}\|_2.$$

Conditioned on the event in Lemma 8.3.5,

$$\widehat{\Lambda}_h\succeq N\Lambda_h\succeq\frac{N}{C_h}\overline{\Lambda}_h,$$

which implies $\|N\widehat{\Lambda}_h^{-1/2}\overline{\Lambda}_h\widehat{\Lambda}_h^{-1/2}\|\leq C_h$. Moreover, conditioned on the event in Lemma 8.3.5,

$$\|\widehat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top\overline{\Phi}_h-N\overline{\Lambda}_h)\widehat{\Lambda}_h^{-1/2}\|_2\leq C\sqrt{d\log(dH/\delta)N}/\lambda.$$

Thus,

$$\|\widehat{\Lambda}_1^{-1/2}\overline{\Lambda}_1\widehat{\Lambda}_1^{-1/2}\|_2\leq C_1/N.$$

and

$$\|\widehat{\Lambda}_h^{-1/2}(\overline{\Phi}_h^\top\overline{\Phi}_h)\widehat{\Lambda}_h^{-1/2}\|_2\leq C_h+C\sqrt{d\log(dH/\delta)N}/\lambda.$$

Finally, by Theorem 1.2 in [36], with probability $1-\delta/(4H)$, for some constant C', we have

$$\|\xi_h\|_{\Phi_h\widehat{\Lambda}_h^{-1}\Phi_h^\top}^2\leq C'H^2d\log(H/\delta).$$

Therefore,

$$\begin{aligned}
&\left\|\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\widehat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3\cdots\widehat{\Lambda}_h^{-1}\Phi_h^\top\xi_h\right\|_{\overline{\Lambda}_1}^2+\left\|\widehat{\Lambda}_1^{-1}\Phi_1^\top\overline{\Phi}_2\widehat{\Lambda}_2^{-1}\Phi_2^\top\overline{\Phi}_3\cdots\lambda\widehat{\Lambda}_h^{-1}\theta_h\right\|_{\overline{\Lambda}_1}^2\\
\leq&\frac{C_1}{N}(C_2+C\sqrt{d\log(d/\delta)N}/\lambda)\times\cdots\times(C_h+C\sqrt{d\log(d/\delta)N}/\lambda)\times(C'H^2d\log(H/\delta)+\lambda H^2d)\\
\leq&\frac{C_1}{N}(C_2+1/H)\times\cdots\times(C_h+1/H)\times(C'H^2d\log(H/\delta)+\lambda H^2d)\\
\leq&\frac{e}{N}C_1\times C_2\times\cdots\times C_h\times(C'H^2d\log(H/\delta)+CdH^3\sqrt{d\log(dH/\delta)N}).
\end{aligned}$$

Let $c>0$ be a large enough constant. We now have

$$\mathbb{E}_{s_1}[(Q_1^\pi(s_1,\pi(s_1))-\widehat{Q}_1(s_1,\pi(s_1)))^2]\leq c\cdot\left(\prod_{h=1}^H C_h\right)\cdot dH^5\cdot\sqrt{\frac{d\log(dH/\delta)}{N}}.$$

□

Remark 6.4.2. The factor $\prod_{h=1}^{H} C_h$ in Theorem 6.4.2 implies that the estimation error will be amplified *geometrically* as the algorithm proceeds. Now we briefly discuss how the error is amplified when running Algorithm 10 on the instance in Section 6.3 to better illustrate the issue. If we run Algorithm 10 on the hard instance in Section 6.3, when $h = H$, the estimation error on $V_H^\pi(s_H^c)$ would be roughly $N^{-1/2}$ for each $c \in [\widehat{d}]$. When using the linear predictor at level H to predict the value of s_H^*, the error will be amplified by $\widehat{d}^{1/2}$. When $h = H - 1$, the dataset contains only s_{H-1}^c for $c \in [\widehat{d}]$, and the estimation error on the value of s_{H-1}^c will be the same as that of s_H^*, which is roughly $(\widehat{d}/N)^{1/2}$. Again, the estimation error on the value of s_{H-1}^* will be $(\widehat{d}^2/N)^{1/2}$ when using the linear predictor at level $H - 1$. As the algorithm proceeds, the error will eventually be amplified by a factor of $\widehat{d}^{H/2}$, which corresponds to the factor $\prod_{h=1}^{H} C_h$ in Theorem 6.4.2.

Policy Completeness. In the offline RL literature, another common representation condition is closedness under Bellman update [16, 27, 62], which is stronger than realizability. In the context of offline policy evaluation, we have the following policy completeness assumption.

Assumption 6.4.2. *For the given policy π, for any $h > 1$ and $\theta_h \in \mathbb{R}^d$, there exists $\theta' \in \mathbb{R}^d$ such that for any $(s, a) \in \mathcal{S}_{h-1} \times \mathcal{A}$, $\mathbb{E}[R(s, a)] + \sum_{s' \in \mathcal{S}_h} P(s' \mid s, a)\phi(s', \pi(s'))^\top \theta_h = \phi(s, a)^\top \theta'$.*

Under Assumption 6.4.2 and the additional assumption that the feature covariance matrices of the data distributions have lower bounded eigenvalue, i.e., $\sigma_{\min}(\Lambda_h) \geq \lambda_0$ for all $h \in [H]$ for some $\lambda_0 > 0$, prior work [27] has shown that for Algorithm 10, by taking $N = \mathrm{poly}(H, d, 1/\varepsilon, 1/\lambda_0)$ samples, we have $(Q_1^\pi(s_1, \pi(s_1)) - \widehat{Q}_1(s_1, \pi(s_1)))^2 \leq \varepsilon$. We omit such an analysis and refer interested readers to [27].

Before ending this section, we would like to note that the above analysis again implies that geometric error amplification is a real issue in offline RL, and sample-efficient offline RL is impossible unless the distribution shift is sufficiently low, i.e., $\prod_{h=1}^{H} C_h$ is bounded, or stronger representation condition such as policy completeness as assumed in prior work.

6.5 Experiments

The goal of our experimental evaluation is to understand whether offline RL methods are sensitive to distribution shift in practical tasks, given a good representation (features extracted from pre-trained neural networks or random features). Our experiments are performed on a range of challenging tasks from the OpenAI gym benchmark suite, including two environments with discrete action space (MountainCar-v0, CartPole-v0) and four environments with continuous action space (Ant-v2, HalfCheetah-v2, Hopper-v2, Walker2d-v2).

6.5.1 Experimental Methodology

Our methodology proceeds according to the following steps:

1. We **decide on a (target) policy** to be evaluated, along with a good feature mapping for this policy.

2. **Collect offline data** using trajectories that are a mixture of the target policy along with another distribution.
3. **Run offline RL methods** to evaluate the target policy using the feature mapping found in Step 1 and the offline data obtained in Step 2.

We now give a detailed description for each step.

Step 1: Determine the Target Policy. To find a policy to be evaluated together with a good representation, we run classical online RL methods. For environments with discrete action space (MountainCar-v0, CartPole-v0), we run Deep Q-learning (DQN) [58], while for environments with continuous action space (Ant-v2, HalfCheetah-v2, Hopper-v2, Walker2d-v2), we run Twin Delayed Deep Deterministic policy gradient (TD3) [31]. The target policy is set to be the final policy output by DQN or TD3. We also set the feature mapping to be the output of the last hidden layer of the learned value function networks, extracted in the final stage of the online RL methods. Since the target policy is set to be the final policy output by the online RL methods, such feature mapping contains sufficient information to represent the value functions of the target policy. We also perform experiments using random Fourier features [69].

Step 2: Collect Offline Data. We consider two styles of shifted distributions: distributions induced by random policies and by lower performance policies. When the data collection policy is the same as the target policy, we will see that offline methods achieve low estimation error, as expected. In our experiments, we use the target policy to generate a dataset $D^\star$ with 1 million samples. We then consider two types of datasets induced by shifted distributions: adding random trajectories into $D^\star$, and adding samples induced by lower performance policies into $D^\star$. In both cases, the amount of data from the target policy remains unaltered (fixed to be 1 million). For the first type of dataset, we add 0.5 million, 1 million, or 2 million samples from random trajectories into $D^\star$. For the second type of dataset, we manually pick four lower performance policies $\pi^1_{\text{sub}}, \pi^2_{\text{sub}}, \pi^3_{\text{sub}}, \pi^4_{\text{sub}}$ with $V^{\pi^1_{\text{sub}}} > V^{\pi^2_{\text{sub}}} > V^{\pi^3_{\text{sub}}} > V^{\pi^4_{\text{sub}}}$, and use each of them to collect 1 million samples. We call these four datasets (each with 1 million samples) $D^1_{\text{sub}}, D^2_{\text{sub}}, D^3_{\text{sub}}, D^4_{\text{sub}}$, and we run offline RL methods on $D^\star \cup D^i_{\text{sub}}$ for each $i \in \{1, 2, 3, 4\}$.

Step 3: Run Offline RL Methods. With the collected offline data and the target policy (together with a good representation), we can now run offline RL methods to evaluate the (discounted) value of the target policy. In our experiments, we run FQI and Least-Squares Temporal Difference(LSTD, a temporal difference offline RL method) [13]. For both algorithms, the only hyperparameter is the regularization parameter λ (cf. Algorithm 10), which we choose from $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-8}\}$. In our experiments, we report the performance of the best-performing λ (measured in terms of the square root of the mean squared estimation error in the final stage of the algorithm, taking average over all repetitions of the experiment); such favorable hyperparameter tuning is clearly not possible in practice (unless we have interactive access to the environment).

In our experiments, we repeat this whole process 5 times. For each FQI round, we report the square root of the mean squared evaluation error, taking average over 100 randomly chosen states. We also report the values ($V^\pi(s)$) of those randomly chosen states in Table 6.1. We

Environment	Discounted Value
Ant-v2	411.77 ± 96.824
CartPole-v0	90.17 ± 20.61
HalfCheetah-v2	1053.71 ± 121.74
Hopper-v2	321.42 ± 30.26
MountainCar-v0	-26.16 ± 17.83
Walker2d-v2	336.64 ± 49.80

Table 6.1: Mean value of the 100 randomly chosen states (used for evaluating the estimations), $\pm$ standard deviation.

note that in our experiments, the randomness combines both from the feature generation process (representation uncertainty, Step 1) and the dataset (Step 2). Even though we draw millions of samples in Step 2, the estimation of FQI could still have high variance.

6.5.2 Results and Analysis

Distributions Induced by Random Policies. We first present the performance of FQI with features from pre-trained neural networks and distributions induced by random policies. The results are reported in Figure 6.2. Perhaps surprisingly, compared to the result on $D^\star$, adding more data (from random trajectories) into the dataset generally hurts the performance. With more data added into the dataset, the performance generally becomes worse. Thus, even with features from pre-trained neural networks, the performance of offline RL methods is still sensitive to data distribution.

Distributions Induced by Lower Performance Polices. Now we present the performance of FQI with features from pre-trained neural networks and datasets with samples from lower performance policies. The results are reported in Figure 6.3. Similar to Figure 6.2, adding more data into the dataset could hurt performance, and the performance of FQI is sensitive to the quality of the policy used to generate samples. Moreover, the estimation error increases exponentially in some cases, showing that geometric error amplification is not only a theoretical consideration, but could occur in practical tasks when given a good representation as well.

Random Fourier Features. Now we present the performance of FQI with random Fourier features and distributions induced by random policies. The results are reported in Figure 6.4. Here we tune the hyperparameters of the random Fourier features so that FQI achieves reasonable performance on $D^\star$. Again, with more data from random trajectories added into the dataset, the performance generally becomes worse. This implies our observations above hold not only for features from pre-trained neural networks, but also for random features. On the other hand, it is known random features achieve reasonable performance in policy gradient methods [70] in the online setting. This suggests that the representation condition required by offline policy evaluation could be stronger than that of policy gradient methods in online setting.

Environment	π^1_{sub}	π^2_{sub}	π^3_{sub}	π^4_{sub}
	RMSE / Gap between target policy and comparison policy			
Ant-v2	>1000 / 26.18	>1000 / 35.07	>1000 / 145.83	>1000 / 146.15
CartPole-v0	6.58 / 4.18	>1000 / 7.04	9.16 / 8.10	15.08 / 12.86
HalfCheetah-v2	35.54 / 118.29	36.45 / 166.31	36.81 / 346.05	86.31 / 482.80
Hopper-v2	36.22 / -4.84	35.54 / -4.37	165.11 / 5.97	>1000 / 17.43
MountainCar-v0	1.46 / 0.74	2.19 / 1.54	2.43 / 2.64	>1000 / 3.98
Walker2d-v2	121.35 / 5.28	>1000 / 34.48	43.47 / 35.30	>1000 / 107.93

Table 6.2: We seek to determine if FQI succeeds in comparing policies (with using features from pre-trained neural networks) with datasets induced by lower performing policies. Roughly, a red entry can be viewed as a failure (ideally, we would hope that the gap is at least a factor of 2 larger than the RMSE). For each entry, the first number is the root mean squared error of the estimation of the target policy of FQI, over 5 repetitions and 100 randomly chosen initial states. The second number is the average gap between the value of the target policy and that of the lower performing policy (π^1_{sub}, π^2_{sub}, π^3_{sub}, or π^4_{sub}), over 5 repetitions and evaluated using 100 trajectories. An entry is marked red if the root mean squared error is larger than the average gap, and is marked blue otherwise. We write > 1000 when the root mean squared error is larger than 1000.

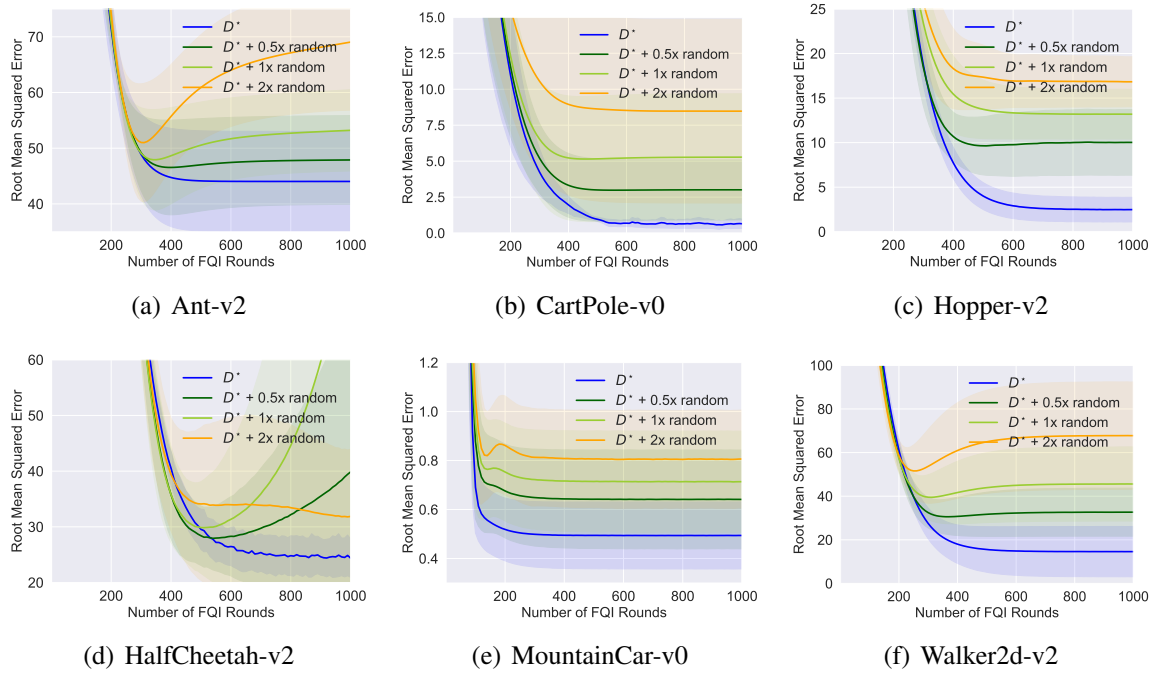
Policy Comparison. We further study whether it is possible to compare the value of the target policy and that of the lower performing policies using FQI. In Table 6.2, we present the policy comparison results of FQI with features from pre-trained neural networks and datasets induced by lower performing policies. For each lower performing policy π^i_{sub} where $i \in \{1, 2, 3, 4\}$, we report the root mean squared error of FQI when evaluating the target policy using $D^\star \cup D^i_{\text{sub}}$, and the average gap between the value of the target policy and that of π^i_{sub}. If the root mean squared error is less than the average gap, then we mark the corresponding entry to be green (meaning that FQI can distinguish between the target policy and the lower performing policy). If the root mean squared error is larger than the average gap, then we mark the corresponding entry to be red (meaning that FQI cannot distinguish between the target policy and the lower performing policy). From Table 6.2, it is clear that for most settings, FQI cannot distinguish between the target policy and the lower performing policy.

Sensitivity to Hyperparameters. In previous experiments, we tune the regularization parameter λ and report the performance of the best-performing λ. However, we remark that in practice, without access to online samples, hyperparameter tuning is hard in offline RL. Here we investigate how sensitive FQI is to different regularization parameters λ. The results are reported in Figure 6.5-6.10. Here, for each environment, we vary the number of additional samples from random trajectories and the regularization parameter λ. As observed in experiments, the regularization parameter λ significantly affects the performance of FQI, as long as there are random trajectories added into the dataset.

Performance of LSTD. Finally, we present the performance of LSTD with features from pre-trained neural networks and distributions induced by random policies. The results are reported

Dataset	$D^\star$	$D^\star$ + 0.5x random	$D^\star$ + 1x random	$D^\star$+ 2x random
Ant-v2	44.03 ± 8.98	48.05 ± 8.03	57.90 ± 13.30	72.80 ± 16.87
HalfCheetah-v2	24.86 ± 3.39	27.54 ± 6.63	30.14 ± 11.60	36.66 ± 21.32
Hopper-v2	2.18 ± 1.14	9.38 ± 3.84	13.18 ± 2.77	16.86 ± 2.84
Walker2d-v2	13.88 ± 11.22	32.73 ± 11.05	45.61 ± 17.06	67.78 ± 24.77

Table 6.3: Performance of LSTD with features from pre-trained neural networks and distributions induced by random policies. Each number of is the square root of the mean squared error of the estimation, taking average over 5 repetitions, $\pm$ standard deviation.

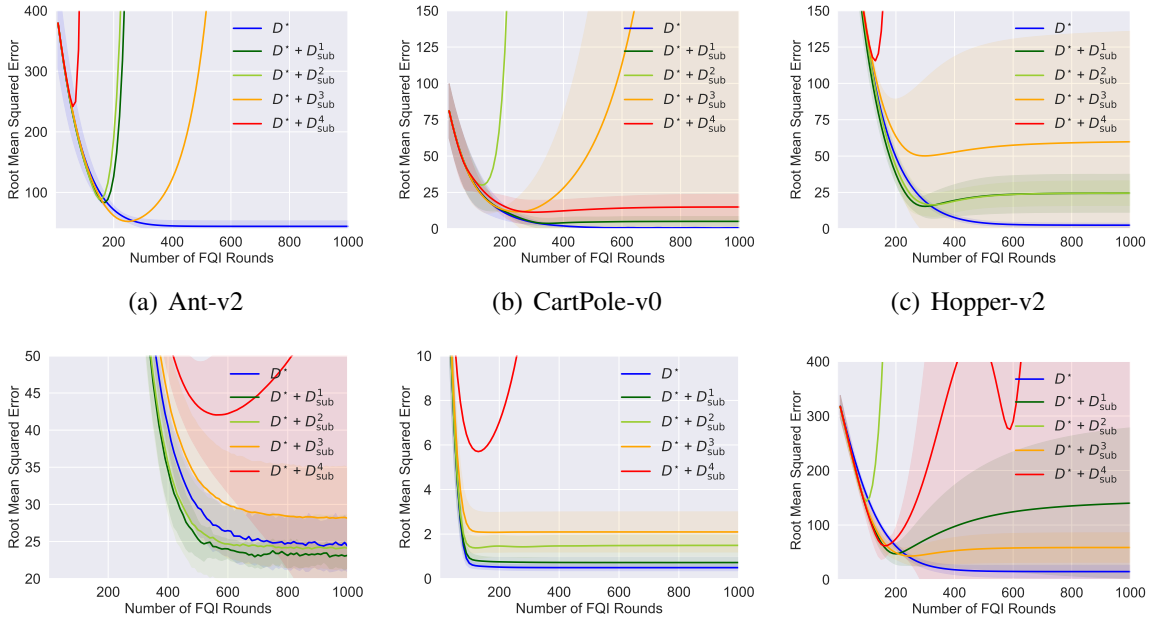


(a) Ant-v2 (b) CartPole-v0 (c) Hopper-v2

(d) HalfCheetah-v2 (e) MountainCar-v0 (f) Walker2d-v2

Figure 6.2: Performance of FQI with features from pre-trained neural networks and datasets induced by random policies.

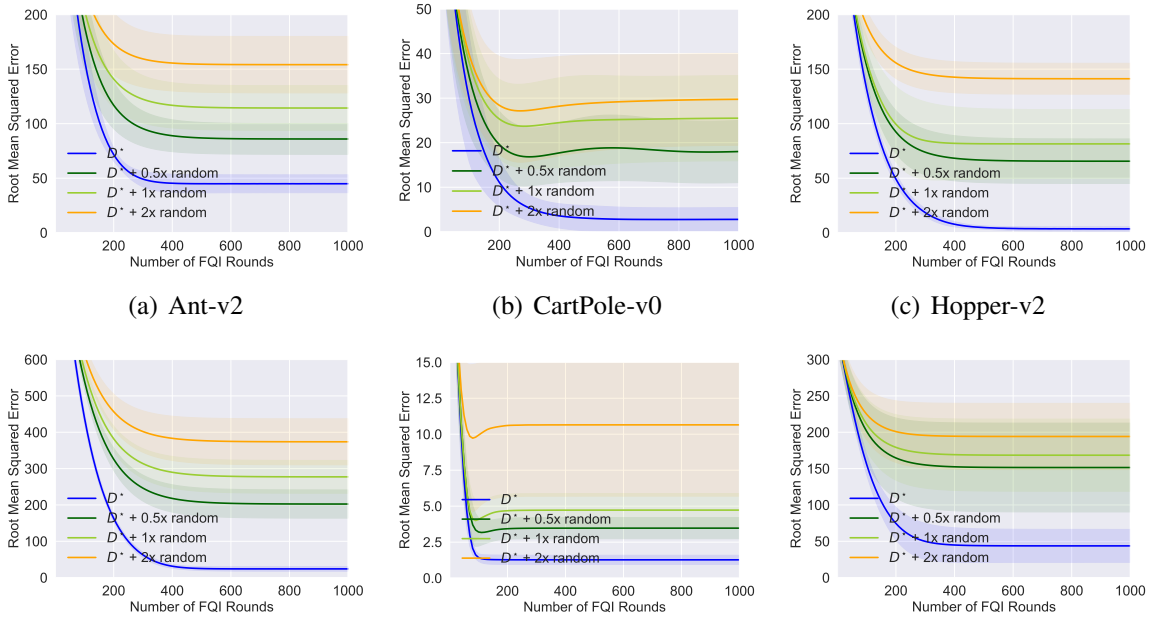
in Table 6.3. With more data from random trajectories added into the dataset, the performance of LSTD becomes worse. This means the sensitivity to distribution shift is not specific to FQI, but also holds for LSTD.



(a) Ant-v2 (b) CartPole-v0 (c) Hopper-v2

(d) HalfCheetah-v2 (e) MountainCar-v0 (f) Walker2d-v2

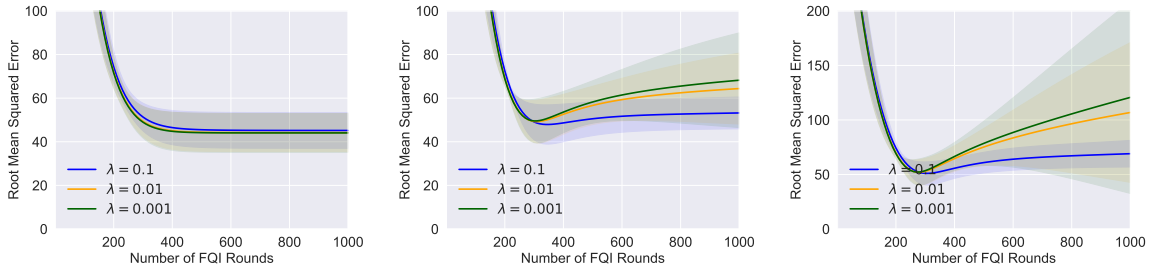
Figure 6.3: Performance of FQI with features from pre-trained neural networks and datasets induced by lower performance policies.



(a) Ant-v2 (b) CartPole-v0 (c) Hopper-v2

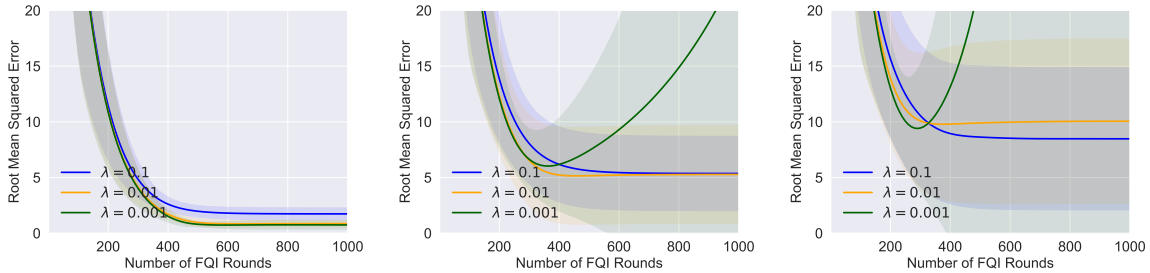
(d) HalfCheetah-v2 (e) MountainCar-v0 (f) Walker2d-v2

Figure 6.4: Performance of FQI with random Fourier features and datasets induced by random policies.



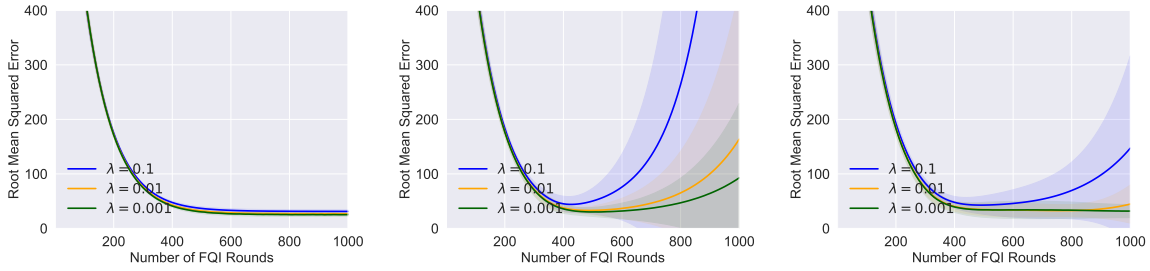
(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6.5: Performance of FQI on Ant-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ.



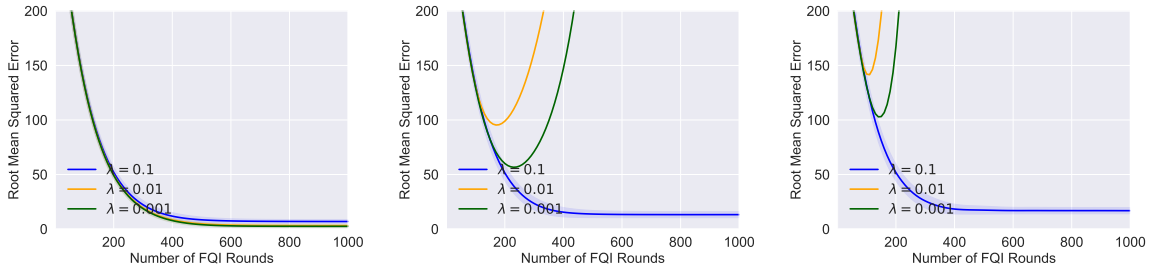
(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6.6: Performance of FQI on CartPole-v0, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ.



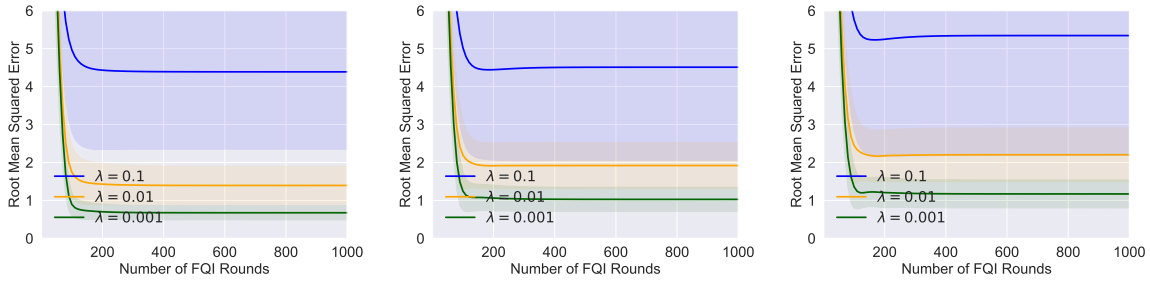
(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6.7: Performance of FQI on HalfCheetah-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ.



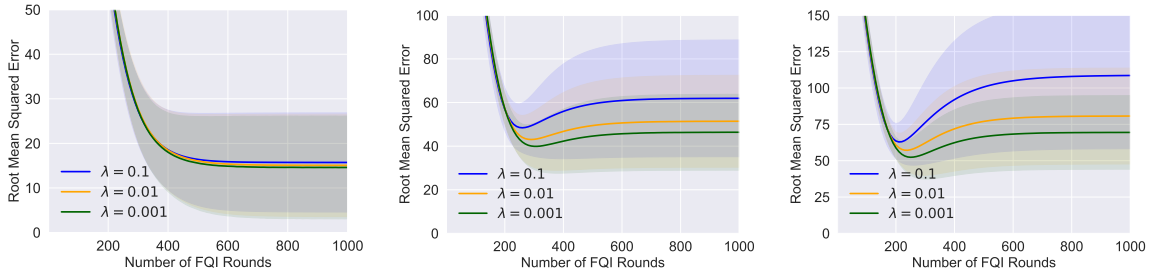
(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6.8: Performance of FQI on Hopper-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ.



(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6.9: Performance of FQI on MountainCar-v0, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ.



(a) $D^\star$ (b) $D^\star$ + 1x random (c) $D^\star$+ 2x random

Figure 6.10: Performance of FQI on Walker2d-v2, with features from pre-trained neural networks, datasets induced by random policies, and different regularization parameter λ.

Part III

RL in Other Settings

Chapter 7

Planning with General Objective Functions

7.1 Introduction

Markov decision process (MDP) is arguably the most popular paradigm for modeling sequential decision-making problems. In this paradigm, it is assumed that the reward values depend only on the current state-action pair, and the objective of the agent is to maximize the summation of all rewards $\sum_{h=1}^{H} r_h$. The drawback of the standard MDP model is that it even fails to capture some simple sequential decision-making tasks. For example, in self-driving, the goal is not to maximize the total reward but to maximize the *minimum* reward on the trajectory, say if one models a car crash as -1 reward and 0 reward otherwise. Note that in this simple example, the state transition function T and the reward function r still satisfy the Markov property. The only difference is that the objective changes from maximizing the *sum* of rewards $\sum_{r=1}^{H} r_h$ to maximizing the *minimum* of rewards $\min_{h=1}^{H} r_h$.

This difference requires the agent to change the planning strategy significantly because the agent needs to look at the full history of rewards. This gives rise the following natural problem: Can we design a provably efficient algorithm for *general* objective functions? This is a challenging question as existing approaches for the MDP model cannot be applied here.

In this chapter, we give a positive answer to the above question by designing an efficient algorithm for objective functions $f(r_1, r_2, \ldots, r_H)$ that satisfy certain technical conditions. Below we list several motivating examples of objective functions that satisfy these conditions.

1. $f(r_1, r_2, \ldots, r_H) = \min\{r_1, r_2, \ldots, r_h\}$: this objective function naturally formalizes sequential decision-making problems related to safety concerns, which we have discussed above.
2. $f(r_1, r_2, \ldots, r_H) = \max\{r_1, r_2, \ldots, r_h\}$: this objective function models the maximum reward-oriented behavior, which has been explicitly studied in the reinforcement learning literature, e.g., in [68], where the authors used this objective function to model certain financial problems.
3. $f(r_1, r_2, \ldots, r_H) = \text{median}\{r_1, r_2, \ldots, r_h\}$: maximizing cumulative rewards is equivalent to maximizing the mean of the reward values, which is not robust to adversarial perturbations and outliers. Maximizing the median or other quantiles of the reward values is a much more robust objective function, which is often used in situations where one seeks

a robust solution. For instance, if each reward is collected by a noisy sensor, the median objective gives a much more robust solution than the mean objective.

4. $f(r_1, r_2, \ldots, r_H) = \sum_{k=1}^{K} r_{(k)}$ where $r_{(k)}$ represents the k-th largest reward in $\{r_h\}_{h=1}^{H}$: this objective function naturally models problems where the agent has a capacity constraint so that the agent can only keep the largest K rewards.

Other objective functions have also appeared in previous work [12, 17, 34, 56, 60, 61, 65, 66, 81, 82]. We stress that the goal of this section is not to study specific objective functions, but to give a characterization on the class of objective functions that admits provably efficient planning algorithms.

7.1.1 Our Results

Our main result is an efficient algorithm that finds near-optimal policies in tabular deterministic systems for a wide range of objective functions. We assume there is an objective function $f : \mathbb{R}^H \to \mathbb{R}$, such that for a sequence of reward values $r_1, r_2, \ldots, r_H$, the objective function f maps the reward values to an objective value $f(r_1, r_2, \ldots, r_H)$. Here H is the planning horizon. We assume all reward values $r_h \in [0, 1]$ and the objective value $f(r_1, r_2, \ldots, r_H) \in [0, 1]$. Therefore, we may assume f is a function that maps a vector in $[0, 1]^H$ to an objective value in $[0, 1]$.

We focus on the planning problem in tabular deterministic systems with general reward functions, i.e., given a deterministic system, our goal is to output a policy which (approximately) maximizes the objective function.[1] Before stating our results, we first give three conditions on the objective function that our algorithm requires.

Definition 7.1.1 (Symmetry)**.** For a function $f \in [0, 1]^H \to [0, 1]$, we say f is *symmetric* if for any permutation $(i_1, i_2, \ldots, i_H)$ of $(1, 2, \ldots, H)$ and $x \in [0, 1]^H$, we have $f(x_1, x_2, \cdots, x_H) = f(x_{i_1}, x_{i_2}, \ldots, x_{i_H})$.

Definition 7.1.2 (Approximate Homogeneity)**.** Let $\bar{\varepsilon}, \bar{\delta} \in (0, 1)$. For a function $f \in [0, 1]^H \to [0, 1]$, we say f satisfies $(\bar{\varepsilon}, \bar{\delta})$*-approximate homogeneity* if for any $x, y \in [0, 1]^H$ such that $x_h \in [y_h, (1 + \bar{\delta})y_h]$ for all $1 \le h \le H$, we have $f(y) \in [f(x) - \bar{\varepsilon}, f(x) + \bar{\varepsilon}]$.[2]

Definition 7.1.3 (Insensitivity to Small Entries)**.** Let $\widehat{\varepsilon}, \widehat{\delta} \in (0, 1)$. For a function $f \in [0, 1]^H \to [0, 1]$, we say f is $(\widehat{\varepsilon}, \widehat{\delta})$*-insensitive to small entires* if for any $x \in [0, 1]^H$ we have $f(\overline{x}) \in [f(x) - \widehat{\varepsilon}, f(x) + \widehat{\varepsilon}]$, where $\overline{x}$ is a vector in $[0, 1]^H$ such that $\overline{x}_h = \begin{cases} x_h & \text{if } x_h \ge \widehat{\delta} \\ 0 & \text{otherwise} \end{cases}$.

Now we briefly discuss the three conditions that our algorithm requires. The first condition requires that the objective function f is symmetric under permutation of coordinates. The second condition requires that, for any input $x \in [0, 1]^H$, if one increases each coordinate in x multiplicatively by a factor of at most $(1 + \bar{\delta})$, then the error on the objective function f is bounded by $\bar{\varepsilon}$. The final condition states that, for any input $x \in [0, 1]^H$, truncating all entries smaller than

[1] We remark that in deterministic systems, the planning problem is almost equivalent to the learning problem (i.e., the agent needs to interact with the environment to learn the transition and the reward), since the agent can readily reach all state-action pairs and learn the transition and reward using linear number of samples.

[2] We remark that the condition $f(y) \in [f(x) - \bar{\varepsilon}, f(x) + \bar{\varepsilon}]$ can be changed to $f(y) \in [(1 - \bar{\varepsilon})f(x), (1 + \bar{\varepsilon})f(x)]$ so that the error on the objective function value is also multiplicative. Note that the later condition is strictly stronger since $f(x) \le 1$ for any $x \in [0, 1]^H$.

$\widehat{\delta}$ to zero leads to an approximation error of at most $\widehat{\varepsilon}$. Given these conditions, now we state our main algorithmic result.

Theorem 7.1.1 (Informal)**.** *Given an objective function f which is symmetric, $(\varepsilon/4, \widehat{\delta})$-insensitive to small entries, and satisfies $(\varepsilon/4, \bar{\delta})$-approximate homogeneity, there is an algorithm that finds an ε-optimal policy in deterministic systems with time complexity $O((|\mathcal{S}||\mathcal{A}|+\mathcal{T})\cdot H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})})$ if evaluating the objective function f on a single input costs $\mathcal{T}$ time.*

As stated in the theorem, the running time of our algorithm exponentially depends on $\log(1/\widehat{\delta})/\bar{\delta}$. However, as we will show in examples given below, $\widehat{\delta}$ and $\bar{\delta}$ are often constants if one aims at a policy with constant additive error, and therefore, our algorithm runs in polynomial time in those cases. Moreover, Our algorithm accesses the objective function f in a black-box manner and thus automatically handles a large class of loss functions.

One may ask whether it is possible to remove those conditions in Definition 7.1.1-7.1.3. We further show that removing any of the three conditions will induce an exponential lower bound and makes the problem intractable in the worst-case. Therefore, all of our three conditions are necessary.

Below we give two large families of objective functions that can be handled by our algorithm. We note that these two families of objective functions have already included all examples mentioned in the introduction.

Symmetric Norm. A symmetric norm is a norm that satisfies the additional property that for any $x \in \mathbb{R}^H$, any permutation σ and any assignment of $s_i \in \{-1, 1\}$, $f(x_1, x_2, \ldots, x_n) = f(s_1 x_{\sigma_1}, s_2 x_{\sigma_2}, \ldots, s_n x_{\sigma_n})$. Symmetric norm includes a large class of norms, for example the ℓ_p norm, the top-k norm (the sum of absolute values of the leading k coordinates of a vector), max-mix of ℓ_p norms (e.g. $\max\{\|x\|_2, c\|x\|_1\}$ for some $c > 0$), and sum-mix of ℓ_p norms (e.g. $\|x\|_2 + c\|x\|_1$ for some $c > 0$), as special cases. More complicated examples include the k-support norm [7] and the box-norm [57], which have found applications in sparse recovery.

For any symmetric norm f that satisfies $f(x) \in [0, 1]$ for any $x \in [0, 1]^H$, f is symmetric, $(\varepsilon, \varepsilon)$-insensitive to small entries and satisfies $(\varepsilon, \varepsilon)$-approximate homogeneity. Therefore, when applying our algorithm to such an objective function f, our algorithm finds an ε-optimal policy in time $O((|\mathcal{S}||\mathcal{A}|+\mathcal{T})\cdot H^{\Theta(\log(1/\varepsilon)/\varepsilon)})$. Thus, our algorithm gives a polynomial-time approximation scheme (PTAS), i.e., the algorithm runs in polynomial time for any constant $\varepsilon > 0$.

Lipschitz Functions. Recall that a function $f : [0, 1]^H \to [0, 1]$ is Lipschitz continuous with respect to the ℓ_∞ norm with Lipschitz constant L if for any $x, y \in \mathbb{R}^H$, $|f(x) - f(y)| \leq L\|x - y\|_\infty$. Clearly, such function f is $(\varepsilon, \varepsilon/L)$-insensitive to small entries and satisfies $(\varepsilon, \varepsilon/L)$-approximate homogeneity. If f is additionally symmetric, then our algorithm finds an ε-optimal policy in time $O((|\mathcal{S}||\mathcal{A}| + \mathcal{T}) \cdot H^{\Theta(\log(L/\varepsilon)L/\varepsilon)})$. Therefore, for constant L and ε, our algorithm runs in polynomial time. An important example that satisfies the above conditions is the median function (or the k-th largest reward for any k), where we have $L = 1$ and thus our algorithm gives a PTAS.

7.2 Algorithm for General Objective Functions

In this section, we present our algorithm which finds an ε-optimal policy for deterministic systems with general reward functions. Here we assume the objective function f is symmetric, $(\varepsilon/4, \widehat{\delta})$-insensitive to small entries, and satisfies $(\varepsilon/4, \bar{\delta})$-approximate homogeneity. We first give the high-level ideas of our algorithm. The formal description is given in Algorithm 11. In Section 7.4, we give the formal analysis of our algorithm.

High-level Ideas. We discretize the reward values, and then find an optimal policy for the discretized reward values using dynamic programming with augmented state space. Below we give more details for these two main components of our algorithm.

Discretization. We discretize reward values so that all rewards values are in

$$\{\widehat{\delta}, \widehat{\delta} \cdot (1+\bar{\delta}), \widehat{\delta} \cdot (1+\bar{\delta})^2, \ldots\}$$

and truncate all reward values less than $\widehat{\delta}$ to zero. Formally, for a state-action pair (s, a), the discretized reward value $\widehat{r}(s, a)$ is defined as

$$\widehat{r}(s,a) = \begin{cases} 0 & R(s,a) < \widehat{\delta} \\ \widehat{\delta} \cdot (1+\bar{\delta})^j & R(s,a) \in [\widehat{\delta} \cdot (1+\bar{\delta})^j, \widehat{\delta} \cdot (1+\bar{\delta})^{j+1}) \end{cases}. \tag{7.1}$$

There are two advantages of using such a discretization approach. First of all, there are only $\log_{1+\bar{\delta}}(1/\widehat{\delta}) = \Theta(\log(1/\widehat{\delta})/\bar{\delta})$ different reward values after discretization. Since the running time of our dynamic programming algorithm depends exponentially on the number of different reward values, such a discretization approach significantly improves the efficiency of our algorithm. Moreover, since the reward function f is assumed to be $(\varepsilon/4, \widehat{\delta})$-insensitive to small entries and satisfy $(\varepsilon/4, \bar{\delta})$-approximate homogeneity, the additive error induced by the discretization approach is upper bounded by $\varepsilon/2$. Therefore, we can find an ε-optimal policy for the original problem if we can find an optimal policy for the deterministic system with discretized reward values.

Dynamic Programming. After the discretization step, the state space for possible reward values has been significantly reduced, and we use a dynamic programming approach to find the optimal policy. For a policy π and a state $s \in \mathcal{S}$, we use $V_h^\pi(s)$ to denote the multiset of reward values on the trajectory starting from state s induced by policy π at level h. We use $V_h^*(s)$ to denote the set of all possible multisets of reward values on trajectories induced by all policies at level h, i.e.,

$$V_h^*(s) = \cup_\pi \{V_h^\pi(s)\}.$$

$Q_h^\pi(s,a)$ and $Q_h^*(s,a)$ are defined analogously. Here, we may safely ignore the order of the reward values since the objective function f is assumed to be symmetric. Moreover, for each $s \in \mathcal{S}$, the size of $V_h^*(s)$ is upper bounded by $H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})}$, since for each discretized reward value r, there are at most H rewards with discretized value r on a trajectory, and there are

only $\log_{1+\bar{\delta}}(1/\widehat{\delta}) = \Theta(\log(1/\widehat{\delta})/\bar{\delta})$ different reward values after the discretization. As shown in Algorithm 11, $V^*(\cdot)$ and $Q^*(\cdot,\cdot)$ can be efficiently calculated, using a Bellman-type dynamic programming algorithm.

Output the Policy. In order to find a policy for the discretized reward values, we enumerate all multisets of reward values $\mathcal{R} \in V_1^*(s_1)$, and find the one with the largest objective value $f(\mathcal{R})$. In order to output the policy, we start from the initial state s_1, find an action $a \in \mathcal{A}$ such that $\mathcal{R} \in Q_h^*(s,a)$, remove $\widehat{r}(s,a)$ from $\mathcal{R}$ and continue this procedure inductively.

Running Time. The time complexity of our algorithm is dominated by the dynamic programming part for calculating $V^*(\cdot)$ and $Q^*(\cdot,\cdot)$. As mentioned above, for each state-action pair (s,a), the size of $Q_h^*(s,a)$ and $V_h^*(s)$ is upper bounded by $H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})}$. Therefore, the running time of the dynamic programming part is at most $O(|\mathcal{S}||\mathcal{A}| \cdot H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})})$. Moreover, in order to output the policy, we evaluate the objective function f on $H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})}$ different inputs. Suppose evaluating the objective function f on a single input costs $\mathcal{T}$ time, the total running time of our algorithm will be

$$O((|\mathcal{S}||\mathcal{A}| + \mathcal{T}) \cdot H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})}).$$

Approximation Guarantee. Under the assumption that the objective function f is symmetric, $(\varepsilon/4, \widehat{\delta})$-insensitive to small entries, and satisfies $(\varepsilon/4, \bar{\delta})$-approximate homogeneity, for any vector $r \in [0,1]^H$, we have

$$|f(r_1, r_2, \ldots, r_H) - f(\widehat{r}_1, \widehat{r}_2, \ldots, \widehat{r}_H)| \leq \varepsilon/2,$$

where for each $h \in [H]$, $\widehat{r}_h$ is the discretized value of r_h as defined in Equation (7.1). Since our algorithm finds the optimal policy with respect to the discretized reward values, the policy π returned by our algorithm satisfies $f(\pi) \geq f(\pi^*) - \varepsilon$ where π^* is the optimal policy.

7.3 Hardness Results

In this section, we present hardness results to demonstrate the necessity of our assumptions on the objective function f. Here, we prove that without any of the three assumptions, any algorithm needs to query the values of $f(\cdot)$ for exponential many different inputs vectors to find a near-optimal policy. For an algorithm that can handle a family of objective functions, the query complexity lower bounds the running time of the algorithm. Therefore, our hardness results demonstrate that all of our three assumptions are necessary to ensure that the problem is efficiently solvable. Here we provide the high-level ideas of our hardness results, and the formal proof will be given in later sections.

Hard Instance. In our hard instances, in each level $h \in [H]$, there is a single state s_h. There are two actions a_1 and a_2 in the action space $\mathcal{A}$, and $P(s_h, a_1) = P(s_h, a_2) = s_{h+1}$ for any $1 \leq h < H$.

Algorithm 11 Deterministic Systems with General Reward Functions

for $h \in [H]$ **do**
  for $(s, a) \in \mathcal{S} \times \mathcal{A}$ **do**
    Let $Q_h^*(s,a) = \begin{cases} \{\{\widehat{r}(s,a)\}\} & \text{if } h = H \\ \{\mathcal{R} \cup \{\widehat{r}(s,a)\} \mid \mathcal{R} \in V_{h+1}^*(P(s,a))\} & \text{otherwise} \end{cases}$
    Let $V_h^*(s) = \cup_{a \in \mathcal{A}} Q_h^*(s,a)$
Initialize policy $\bar{\pi}$ arbitrarily
for $\mathcal{R} \in V_1^*(s_1)$ **do**
  Initialize policy $\pi^{\mathcal{R}}$ arbitrarily
  Let $\mathcal{R}^1 = \mathcal{R}$
  for $h \in [H]$ **do**
    Let $\pi_h^{\mathcal{R}}(s_h) = a \in \mathcal{A}$ such that $\mathcal{R}^h \in Q_h^*(s_h, a)$
    Let $\mathcal{R}^{h+1} = \mathcal{R}^h \setminus \{\widehat{r}(s_h, a)\}$
    Let $s_{h+1} = P(s_h, \pi^{\mathcal{R}}(s_h))$
  Let $\bar{\pi} = \pi^{\mathcal{R}}$ if $f(\pi^{\mathcal{R}}) > f(\bar{\pi})$
Return $\bar{\pi}$

Necessity of Symmetry. We first show that if the objective function f is insensitive to small entries, satisfies approximate homogeneity but is not symmetric, then any algorithm still needs to query exponential number of values of f to find a near-optimal policy, and thus demonstrate the necessity of the assumption that f is symmetric. Here we have $R(s, a_1) = 1/2$ and $R(s, a_2) = 1$ for any $s \in \mathcal{S}$. Now we define the objective function f, which is parameterized by a vector $\theta \in \{1/2, 1\}^H$.

For a vector $\theta \in \{1/2, 1\}^H$, we define a function $f_\theta : [0,1]^H \to [0,1]$. For a vector $x \in [0,1]^H$, if there exists $x_h = 0$ for some $h \in [H]$ then we define $f_\theta(x) = 0$. Otherwise,

$$f_\theta(x) = \min_{h \in [H]} \min\{x_h/\theta_h, \theta_h/x_h\}.$$

It is easy to verify that for any $\varepsilon > 0$, f satisfies $(\varepsilon, \varepsilon)$-approximate homogeneity and is $(2\varepsilon, \varepsilon)$-insensitive to small entries. In the hard instance, the objective function f is set to be f_θ, where θ is one of the 2^H vectors in $\{1/2, 1\}^H$.

Recall that in our hard instance, all rewards values are in $\{1/2, 1\}$, and for any $x \in \{1/2, 1\}^H$, $f_\theta(x) = 1$ if $x = \theta$, and $f_\theta(x) = 1/2$ if $x \neq \theta$. Therefore, in order to receive an objective value of 1, the agent must choose the correct actions for all the H steps, and otherwise the agent will always receive an objective value of $1/2$. Here, the optimal policy is $\pi(s_h) = a_1$ if $\theta_h = 1/2$ and $\pi(s_h) = a_2$ if $\theta_h = 1$. Therefore, the correct actions are fully encoded in the vector θ. However, there are 2^H possible vectors for θ. Therefore, intuitively, in order to find the correct actions for all the H steps, the agent must enumerate all the 2^H possible combinations of actions to figure out the underlying vector θ, which inevitably induces an exponential query complexity. This intuition is made formal is the supplementary material using Yao's minimax principle [104].

Necessity of Approximate Homogeneity. Here we show that if the objective function f is symmetric, insensitive to small entries but does not satisfy approximate homogeneity, then any

algorithm still needs to query exponential number of values of f to find a near-optimal policy, and thus demonstrate the necessity of approximate homogeneity. Here we have $R(s_h, a_1) = (2h + 2H - 1)/4H$ and $R(s, a_2) = (h+H)/2H$ for any $h \in [H]$. Now we define the objective function f, which is parameterized by a vector $\theta \in \mathbb{R}^H$ where $\theta_h \in \{(2h + 2H - 1)/4H, (h + H)/2H\}$ for all $h \in [H]$.

For any vector $x \in [0, 1]^H$, we use $i_1, i_2, \ldots, i_H$ to denote a permutation of $(1, 2, \ldots, H)$ such that $0 \leq x_{i_1} \leq x_{i_2} \leq \ldots \leq x_{i_H} \leq 1$. We define $f_\theta(x) = 1$ if $x_{i_h} = \theta_h$ for all $h \in [H]$, and $f_\theta(x) = 0$ otherwise. Clearly, f is symmetric and $(0, \varepsilon)$-insensitive to small entries for any $\varepsilon \leq 1/2$, but does not satisfy approximate homogeneity. In the hard instance, the objective function f is set to be f_θ, where θ is one of the 2^H vectors defined above.

In order to receive an objective value of 1, the agent must choose the correct actions for all the H steps, and otherwise the agent will always receive an objective value of 0, which implies an exponential query lower bound using the same argument mentioned above.

Necessity of Insensitivity to Small Entries. Here we show that if the objective function f is symmetric, satisfies approximate homogeneity but is not insensitive to small entries, then any algorithm still needs to query exponential number of values of f to find a near-optimal policy, and thus demonstrate the necessity of insensitivity to small entries.

Here we have $R(s_h, a_1) = 2^{-H(2h-1)}$ and $R(s_h, a_2) = 2^{-2Hh}$ for any $h \in [H]$. Now we define the objective function f, which is parameterized by a vector $\theta \in \mathbb{R}^H$ where

$$\theta_h \in \{2^{-H(2h-1)}, 2^{-2Hh}\}$$

for all $h \in [H]$.

For a vector θ satisfies the above condition, we define a function $f_\theta : [0, 1]^H \to [0, 1]$. For a vector $x \in [0, 1]^H$, if there exists $x_h = 0$ for some $h \in [H]$ then we define $f_\theta(x) = 0$. Otherwise, we use $i_1, i_2, \ldots, i_H$ to denote a permutation of $(1, 2, \ldots, H)$ such that

$$1 \geq x_{i_1} \geq x_{i_2} \geq \ldots \geq x_{i_H} \geq 0,$$

and we define

$$f_\theta(x) = \min_{h \in [H]} \min\{x_{i_h}/\theta_h, \theta_h/x_{i_h}\}.$$

It is easy to verify that for any $\varepsilon > 0$, f satisfies $(\varepsilon, \varepsilon)$-approximate homogeneity. It is also clear that f is symmetric. In the hard instance, the objective function f is set to be f_θ, where θ is one of the 2^H vectors defined above.

In order to receive an objective value of 1, the agent must choose the correct actions for all the H steps, and otherwise the agent will receive an objective value of $1/2$. The lower bound can be proved using the same argument as above.

7.4 Proof of Theorem 7.1.1

Recall that for a reward value r, the discretized reward value $\widehat{r}$ is defined as

$$\widehat{r} = \begin{cases} 0 & r < \widehat{\delta} \\ \widehat{\delta} \cdot (1 + \bar{\delta})^j & r \in [\widehat{\delta} \cdot (1 + \bar{\delta})^j, \widehat{\delta} \cdot (1 + \bar{\delta})^{j+1}) \end{cases}.$$

We restate Theorem 7.1.1 as follow.

Theorem 7.4.1. *Given an objective function $f : [0,1]^H \to [0,1]$ which is symmetric, $(\varepsilon/4, \widehat{\delta})$-insensitive to small entries, and satisfies $(\varepsilon/4, \bar{\delta})$-approximate homogeneity, Algorithm 11 finds an ε-optimal policy in deterministic systems with time complexity*

$$O((|\mathcal{S}||\mathcal{A}| + \mathcal{T}) \cdot H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})})$$

if evaluating the objective function f of a single policy costs $\mathcal{T}$ time.

Proof. Let us first consider the running time of Algorithm 11. For each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [h]$, each element $\mathcal{R}$ in $Q_h^*(s,a)$ is a multiset of discretized reward values. Since we only have $\log(1/\widehat{\delta})/\bar{\delta}$ different discretized reward values and the size of $\mathcal{R}$ is at most H, the size of $Q_h^*(s,a)$ is at most $H^{O(\log(1/\widehat{\delta})/\bar{\delta})}$. Thus, the running time of the first loop is at most $|\mathcal{S}| \cdot |\mathcal{A}| \cdot H^{O(\log(1/\widehat{\delta})/\bar{\delta})}$. Since the number of different multisets $\mathcal{R}$ is at most $H^{O(\log(1/\widehat{\delta})/\bar{\delta})}$ and we need $H \cdot |\mathcal{A}| + \mathcal{T}$ time for each iteration of the second loop of Algorithm 11, we need $O(H \cdot |\mathcal{A}| + \mathcal{T}) \cdot H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})}$ time to find $\bar{\pi}$. Thus, the total running time is $O((|\mathcal{S}||\mathcal{A}| + \mathcal{T}) \cdot H^{\Theta(\log(1/\widehat{\delta})/\bar{\delta})})$.

Next, we prove the correctness of the algorithm.

Claim 7.4.1. *For any state $s \in \mathcal{S}$ and $h \in [H]$, a multiset $\mathcal{R}$ belongs to $V_h^*(s)$ if and only if there is a trajectory starting from state s at level h whose multiset of discretized reward values is exactly $\mathcal{R}$.*

Proof. Suppose $\mathcal{R} \in V_h^*(s)$. We want to show that there is a trajectory starting from s at level h whose multiset of discretized reward values is exactly $\mathcal{R}$. The proof is by induction on $h \in [H]$. Consider the base case when $h = H$. For any $s_H \in \mathcal{S}$, we have that $V_H^*(s_H) = \bigcup_{a \in \mathcal{A}} Q_H^*(s_H, a) = \bigcup_{a \in \mathcal{A}} \{\{\widehat{r}(s_H, a)\}\} = \{\{\widehat{r}(s_H, a)\} \mid a \in \mathcal{A}\}$. Thus, for any $\mathcal{R} \in V_H^*(s_H)$, there is a trajectory starting from s_H whose multiset of discretized reward values is exactly $\mathcal{R}$. Suppose the claim is true for level $h+1$. Consider a level h and a state $s \in \mathcal{S}$. Let $\mathcal{R} \in V_h^*(s_h)$. According to Algorithm 11, there exists $a \in \mathcal{A}$ such that $\mathcal{R} \in Q_h^*(s_h, a)$. Let $s_{h+1} = P(s_h, a)$. We know that $\mathcal{R} \setminus \{\widehat{r}(s_h, a)\} \in V_{h+1}^*(s_{h+1})$. By the induction hypothesis, there is a trajectory starting from s_{h+1} whose multiset of discretized reward values is exactly $\mathcal{R} \setminus \{\widehat{r}(s_h, a)\}$. Thus, there exists a trajectory starting from s_h whose multiset of discretized reward values is exactly $\mathcal{R}$.

Suppose there exists a trajectory starting from state s and level h whose multiset of discretized reward values is exactly $\mathcal{R}$. We want to show that $\mathcal{R} \in V_h^*(s)$. The proof is by induction on $h \in [H]$. Consider the base case when $h = H$. For any $s_H \in \mathcal{S}$, we have that $V_H^*(s_H) = \bigcup_{a \in \mathcal{A}} Q_H^*(s_H, a) = \bigcup_{a \in \mathcal{A}} \{\{\widehat{r}(s_H, a)\}\} = \{\{\widehat{r}(s_H, a)\} \mid a \in \mathcal{A}\}$. Thus, for any trajectory starting from s_H whose multiset of discretized reward values is exactly $\mathcal{R}$, we have $\mathcal{R} \in V_H^*(s_H)$. Suppose the claim is true for level $h+1$. Consider a level h and a state $s_h \in \mathcal{S}$ and a trajectory $s_h, a_h, \widehat{r}_h, s_{h+1}, a_{h+1}, \widehat{r}_{h+1}, \cdots, s_H, a_H, \widehat{r}_H$ starting from s_h. Let $\mathcal{R} = \{\widehat{r}_h, \widehat{r}_{h+1}, \cdots, \widehat{r}_H\}$. By induction hypothesis, we have $\mathcal{R} \setminus \{\widehat{r}_h\} \in V_{h+1}^*(s_{h+1})$. According to Algorithm 11, we have $\mathcal{R} \in Q_h^*(s_h, a_h)$. Thus, we have $\mathcal{R} \in V_h^*(s_h)$. □

Claim 7.4.2. *Let $s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_H, a_H, r_H$ be the trajectory induced by policy $\pi^{\mathcal{R}}$. We have $\mathcal{R} = \{\widehat{r}_1, \widehat{r}_2, \cdots, \widehat{r}_H\}$.*

Proof. We prove that for any $h \in [H]$, we have $\mathcal{R}^h \cup \{\widehat{r}(s_i, \pi^{\mathcal{R}}(s_i)) \mid i \in [h-1]\} = \mathcal{R}$. The proof is by induction. For $h = 1$, it is true since $\mathcal{R}^1 = \mathcal{R}$. Furthermore, by Claim 7.4.1, we know that there exists $a \in \mathcal{A}$ such that $\mathcal{R}^1 \in Q_1^*(s_1, a)$. Suppose the desired claim is true for level $h-1$. In the h-th iteration, we have $\mathcal{R}^{h-1} \in Q_h^*(s_h, \pi^{\mathcal{R}}(s_h))$ and $\mathcal{R}^h = \mathcal{R}^{h-1} \setminus \{\widehat{r}(s_h, \pi^{\mathcal{R}}(s_h))\}$. Thus, we have $\mathcal{R} = \mathcal{R}^{h-1} \cup \{\widehat{r}(s_i, \pi^{\mathcal{R}}(s_i)) \mid i \in [h-1]\} = \mathcal{R}^h \cup \{\widehat{r}(s_i, \pi^{\mathcal{R}}(s_i)) \mid i \in [h-1]\} \cup \{\widehat{r}(s_h, \pi^{\mathcal{R}}(s_h))\} = \mathcal{R}^h \cup \{\widehat{r}(s_i, \pi^{\mathcal{R}}(s_i)) \mid i \in [h]\}$. Furthermore, by Claim 7.4.1, we have $\mathcal{R}^h \in V_{h+1}^*(s_{h+1})$.

Notice that $\mathcal{R}^H = \emptyset$ and thus the desired claim is proved. □

Now we formally prove Theorem 7.4.1.

Let $s_1^*, a_1^*, r_1^*, s_2^*, a_2^*, r_2^*, \cdots, s_H^*, a_H^*, r_H^*$ be the trajectory induced by the optimal policy. Let $\mathcal{R}^* = \{r_h^* \mid h \in [H]\}$, and $\widehat{\mathcal{R}} = \{\widehat{r}_h \mid h \in [H]\}$ be the discretized version of $\mathcal{R}^*$. According to Claim 7.4.1, we know that $\widehat{\mathcal{R}} \in V_1^*(s_1)$. Let $\widehat{s}_1, \widehat{a}_1, \widehat{r}_1, \widehat{s}_2, \widehat{a}_2, \widehat{r}_2, \cdots, \widehat{s}_H, \widehat{a}_H, \widehat{r}_H$ be the trajectory induced by the policy $\pi_{\widehat{\mathcal{R}}}$ with discretized reward values. According to Claim 7.4.2, we have $\widehat{\mathcal{R}} = \{\widehat{r}_1, \widehat{r}_2, \cdots, \widehat{r}_H\}$. Let $\widehat{s}_1, \widehat{a}_1, \tilde{r}_1, \widehat{s}_2, \widehat{a}_2, \tilde{r}_2, \cdots, \widehat{s}_H, \widehat{a}_H, \tilde{r}_H$ be the trajectory induced by the policy $\pi_{\widehat{\mathcal{R}}}$ with original (undiscretized) reward values. Let $\tilde{\mathcal{R}} = \{\tilde{r}_1, \tilde{r}_2, \cdots, \tilde{r}_H\}$. By the choice of $\bar{\pi}$ outputted by Algorithm 11, we have:

$$
\begin{aligned}
f(\bar{\pi}) &\geq f(\pi_{\widehat{\mathcal{R}}}) \\
&= f(\{\tilde{r}_1, \tilde{r}_2, \cdots, \tilde{r}_H\}) \\
&\geq f\left(\left\{\tilde{r}_h \cdot \mathbb{I}[(\tilde{r}_h \geq \bar{\delta}] \mid h \in [H]\right\}\right) - \varepsilon/4 \\
&\geq f(\widehat{\mathcal{R}}) - \varepsilon/2 \\
&\geq f\left(\left\{r_h^* \cdot \mathbb{I}[r_h^* \geq \bar{\delta}] \mid h \in [H]\right\}\right) - 3 \cdot \varepsilon/4 \\
&\geq f(\mathcal{R}^*) - \varepsilon,
\end{aligned}
$$

where the first step follows from $\widehat{\mathcal{R}} \in V^*(s_1)$ and the choice of $\bar{\pi}$, the third step follows from that $f(\cdot)$ is $(\varepsilon/4, \bar{\delta})$-insensitive to small entries, the fourth step follows from that $f(\cdot)$ is $(\varepsilon/4, \widehat{\delta})$-approximate homogeneous, the fifth step follows from that $f(\cdot)$ is $(\varepsilon/4, \widehat{\delta})$-approximate homogeneous and the last step follows from that $f(\cdot)$ is $(\varepsilon/4, \bar{\delta})$-insensitive to small entries.

Thus, $\bar{\pi}$ is an ε-optimal policy. □

7.5 Proof of Lower Bounds

In this section, we formally prove our lower bounds.

Theorem 7.5.1. *There is a family $\mathcal{F}$ of objective functions which are $(\varepsilon, \varepsilon)$-approximate homogeneous and are $(2\varepsilon, \varepsilon)$-insensitive to small entires but are not necessarily symmetric, such that any algorithm which can output a 0.49-optimal policy for any objective function $f \in \mathcal{F}$ with probability at least 0.9 needs to query the objective values of at least $0.9 \cdot 2^H$ policies in the worst case.*

Proof. We describe our deterministic system as the following. For each $h \in [H]$, there is a state $s_h \in \mathcal{S}$. There are two actions, a_1 and a_2, in the action space $\mathcal{A}$, and $P(s_h, a_1) = P(s_h, a_2) =$

s_{h+1} for any $1 \leq h < H$. The reward function satisfies that $R(s_h, a_1) = 1/2$ and $R(s_h, a_2) = 1$ for $h \in [H]$.

For a vector $\theta \in \mathbb{R}^H$, we define a function $f_\theta : [0,1]^H \to [0,1]^H$, if there exists $x_h = 0$ for some $h \in [H]$ then we define $f_\theta(x) = 0$. Otherwise,

$$f_\theta(x) = \min_{h \in [H]} \min\{x_h/\theta_h, \theta_h/x_h\}.$$

Let $\mathcal{F} = \{f_\theta \mid \theta \in \{1/2, 1\}^H\}$. Firstly, we show that for any $f \in \mathcal{F}$, f is $(\varepsilon, \varepsilon)$-approximate homogeneous for any $\varepsilon \geq 0$. Let $\varepsilon \geq 0$. Consider two vectors $x, y \in [0,1]^H$ such that for any $h \in [H]$, $x_h \in [y_h, (1+\varepsilon)y_h]$. For $h \in [H]$, if $x_h \leq \theta_h$, then $y_h/\theta_h \leq x_h/\theta_h$,

$$y_h/\theta_h \geq x_h/\theta_h/(1+\varepsilon) \geq (1-\varepsilon) \cdot (x_h/\theta_h) \geq x_h/\theta_h - \varepsilon,$$

and $\theta_h/y_h \geq \theta_h/x_h \geq 1$. If $x_h \geq \theta_h$, then $\theta_h/y_h \geq \theta_h/x_h$,

$$\theta_h/y_h \leq (1+\varepsilon) \cdot \theta_h/x_h \leq \theta_h/x_h + \varepsilon,$$

and $y_h/\theta_h \geq 1/(1+\varepsilon) \geq 1-\varepsilon$. Thus, $f_\theta(y) \in [f_\theta(x) - \varepsilon, f_\theta(x) + \varepsilon]$. Next, we show that for any $f \in \mathcal{F}$, f is $(2\varepsilon, \varepsilon)$-insensitive to small entries. Let $\varepsilon \geq 0$. Consider any $f_\theta \in \mathcal{F}$, and any $x \in [0,1]^H, h \in [H]$ with $x_h \leq \varepsilon$. If $\theta_h = 1$, then $\min\{x_h/\theta_h, \theta_h/x_h\} \leq \varepsilon$. If $\theta_h = 1/2$, then $\min\{x_h/\theta_h, \theta_h/x_h\} \leq 2\varepsilon$. Thus, f_θ is $(2\varepsilon, \varepsilon)$-insensitive to small entries.

Consider an arbitrary vector $x \in \{1/2, 1\}^H$ and a function $f_\theta \in \mathcal{F}$. If $x = \theta$, then by the definition of f_θ, we know that $f_\theta(x) = 1$. If $x \neq \theta$, let us consider any $h \in [H]$ such that $x_h \neq \theta_h$. If $x_h = 1/2$ and $\theta_h = 1$, then $\min\{x_h/\theta_h, \theta_h/x_h\} = 1/2$. If $x_h = 1$ and $\theta_h = 1/2$, then $\min\{x_h/\theta_h, \theta_h/x_h\} = 1/2$. Thus, $f_\theta(x) = 1/2$ when $x \neq \theta$.

Now consider a policy π and an objective function $f_\theta \in \mathcal{F}$. Let

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_H, a_H, r_H$$

be the trajectory induced by π. Let $x = (r_1, r_2, \cdots, r_H)$. We know that the optimal policy for f_θ is the policy with $x = \theta$, and in that case we have $f_\theta(\pi) = 1$. For any non-optimal policy π we know that $f_\theta(\pi) = 1/2$.

Now we prove the desired result. Our proof is by reduction from INDQ_{2^H}. Suppose we have an algorithm $\mathcal{M}$ which outputs 0.49-optimal policy for any $f \in \mathcal{F}$. We will show that there is a query algorithm for INDQ_{2^H}. In problem INDQ_{2^H}, there is an underlying $\theta^* \in \{1/2, 1\}^H$ and we want to find θ^*. We can imagine that the deterministic system has objective function $f_{\theta^*} \in \mathcal{F}$ and then we simulate $\mathcal{M}$. Suppose the i-th query policy of $\mathcal{M}$ is π, then we let $x = (r_1, r_2, \cdots, r_H)$ be the reward values induced by π. Then we query whether $x = \theta^*$. If the answer is yes, then we are done. Otherwise, since $f_{\theta^*}(x) = 1/2$ for $x \neq \theta^*$, we can return an objective value of $1/2$ for the i-th query of $\mathcal{M}$ and continue the simulation of $\mathcal{M}$. Since $\mathcal{M}$ can output a 0.49-optimal policy with probability at least 0.9, it must output the optimal policy with probability at least 0.9 which means that it can eventually find $x = \theta^*$ with probability at least 0.9. According to Theorem 5.4.1, $\mathcal{M}$ must query at least $0.9 \cdot 2^H$ policies for the worst $f \in \mathcal{F}$. □

Theorem 7.5.2. *There is a family $\mathcal{F}$ of objective functions which are symmetric and are $(0, \varepsilon)$-insensitive to small entires for any $\varepsilon \leq 1/2$ but are not necessarily approximate homogeneous,*

such that any algorithm which can output a 0.99*-optimal policy for any objective function* $f \in \mathcal{F}$ *with probability at least* 0.9 *needs to query the objective values of at least* $0.9 \cdot 2^H$ *policies in the worst case.*

Proof. We describe our deterministic system as the following. For each $h \in [H]$, there is a state $s_h \in \mathcal{S}$. There are two actions, a_1 and a_2, in the action space $\mathcal{A}$, and $P(s_h, a_1) = P(s_h, a_2) = s_{h+1}$ for any $1 \leq h < H$. The reward function satisfies that $R(s_h, a_1) = (2H + 2h - 1)/4H$ and $R(s_h, a_2) = (2H + 2h)/4H$ for $h \in [H]$.

Now we define f_θ, which is parameterized by a vector $\theta \in \mathbb{R}^H$. For any vector $x \in [0, 1]^H$, we use $i_1, i_2, \ldots, i_H$ to denote a permutation of $(1, 2, \ldots, H)$ such that

$$0 \leq x_{i_1} \leq x_{i_2} \leq \ldots \leq x_{i_H} \leq 1.$$

We define $f_\theta(x) = 1$ if $x_{i_h} = \theta_h$ for all $h \in [H]$, and $f_\theta(x) = 0$ otherwise. Let

$$\mathcal{F} = \left\{ f_\theta \mid \theta \in \mathbb{R}^H, \forall h \in [H], \theta_h \in \{(2h + 2H - 1)/4H, (h + H)/2H\} \right\}.$$

By construction, f_θ is clearly symmetric. Since for any $f_\theta \in \mathcal{F}$, each entry of θ is greater than $1/2$, $f_\theta(x)$ must be 0 if x has any entry at most $1/2$ which implies that f_θ is $(0, \varepsilon)$-insensitive to small entries for any $\varepsilon \leq 1/2$.

Consider an arbitrary vector $x \in \mathbb{R}^H$ and a function $f_\theta \in \mathcal{F}$. Without loss of generality, we can assume $x_1 \leq x_2 \leq \cdots \leq x_H$. If $x = \theta$, then $f_\theta(x) = 1$. Otherwise, $f_\theta(x) = 0$ according to the definition of f_θ.

Now consider a policy π and an objective function $f_\theta \in \mathcal{F}$. Let

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_H, a_H, r_H$$

be the trajectory induced by π. Let $x = (r_1, r_2, \cdots, r_H)$. We know that the optimal policy for f_θ is the policy with $x = \theta$. For any non-optimal policy π we know that $f_\theta(\pi) = 0$.

Now we prove the desired result. Our proof is by reduction from INDQ_{2^H}. Suppose we have an algorithm $\mathcal{M}$ which outputs 0.99-optimal policy for any $f \in \mathcal{F}$. We will show that there is a query algorithm for INDQ_{2^H}. In problem INDQ_{2^H}, there is an underlying $\theta^* \in \mathbb{R}^H$ satisfying for any $h \in [H], \theta_h \in \{(2H + 2h - 1)/4H, (H + h)/2H\}$ and we want to find θ^*. We can imagine that the deterministic system has objective function $f_{\theta^*} \in \mathcal{F}$ and then we simulate $\mathcal{M}$. Suppose the i-th query policy of $\mathcal{M}$ is π, then we let $x = (r_1, r_2, \cdots, r_H)$ be the reward values induced by π. Due to the construction of our deterministic system, we have $r_1 \leq r_2 \leq \cdots \leq r_H$. Then we query whether $x = \theta^*$. If the answer is yes, then we are done. Otherwise, since $f_{\theta^*}(x) = 0$ for $x \neq \theta^*$, we can return an objective value of 0 for the i-th query of $\mathcal{M}$ and continue the simulation of $\mathcal{M}$. Since $\mathcal{M}$ can output a 0.99-optimal policy with probability at least 0.9, it must output the optimal policy with probability at least 0.9 which means that it can eventually find $x = \theta^*$ with probability at least 0.9. According to Theorem 5.4.1, $\mathcal{M}$ must query at least $0.9 \cdot 2^H$ policies for the worst $f \in \mathcal{F}$. □

Theorem 7.5.3. *There is a family* $\mathcal{F}$ *of objective functions which are symmetric and are* $(\varepsilon, \varepsilon)$*-approximate homogeneous for any* $\varepsilon \geq 0$ *but are not necessarily insensitive to small entries, such that any algorithm which can output a* 0.49*-optimal policy for any objective function* $f \in \mathcal{F}$ *with probability at least* 0.9 *needs to query the objective values of at least* $0.9 \cdot 2^H$ *policies in the worst case.*

Proof. We describe our deterministic system as the following. For each $h \in [H]$, there is a state $s_h \in \mathcal{S}$. There are two actions, a_1 and a_2, in the action space $\mathcal{A}$, and $P(s_h, a_1) = P(s_h, a_2) = s_{h+1}$ for any $1 \le h < H$. The reward function satisfies that $R(s_h, a_1) = 2^{-H(2h-1)}$ and $R(s_h, a_2) = 2^{-2Hh}$ for $h \in [H]$.

Now we define f_θ, which is parameterized by a vector $\theta \in \mathbb{R}^H$. For any vector $x \in [0,1]^H$, we use $i_1, i_2, \ldots, i_H$ to denote a permutation of $(1, 2, \ldots, H)$ such that

$$1 \ge x_{i_1} \ge x_{i_2} \ge \ldots \ge x_{i_H} \ge 0.$$

We define

$$f_\theta(x) = \min_{h \in [H]} \min(x_{i_h}/\theta_h, \theta_h/x_{i_h}).$$

Let

$$\mathcal{F} = \left\{ f_\theta \mid \theta \in \mathbb{R}^H, \forall h \in [H], \theta_h \in \{2^{-H(2h-1)}, 2^{-2Hh}\} \right\}.$$

By construction, f_θ is symmetric. Next, we show that for any $f \in \mathcal{F}$, f is $(\varepsilon, \varepsilon)$-approximate homogeneous for any $\varepsilon \ge 0$. Consider two vectors $x, y \in [0,1]^H$ such that for any $h \in [H], x_h \in [y_h, (1+\varepsilon)y_h]$. We use $i_1, i_2, \ldots, i_H$ to denote a permutation of $(1, 2, \ldots, H)$ such that

$$1 \ge x_{i_1} \ge x_{i_2} \ge \ldots \ge x_{i_H} \ge 0.$$

We use $i'_1, i'_2, \ldots, i'_H$ to denote a permutation of $(1, 2, \ldots, H)$ such that

$$1 \ge y_{i'_1} \ge y_{i'_2} \ge \ldots \ge y_{i'_H} \ge 0.$$

We claim that for any $h \in [H]$, we have $x_{i_h} \in [y_{i'_h}, (1+\varepsilon)y_{i'_h}]$. The reason is as follows. Because

$$x_{i'_1} \ge y_{i'_1} \ge y_{i'_h}, x_{i'_2} \ge y_{i'_2} \ge y_{i'_h}, \cdots, x_{i'_h} \ge y_{i'_h},$$

the h-th largest value x_{i_h} in $x_1, x_2, \cdots, x_H$ must be at least $y_{i'_h}$. Because

$$x_{i'_H}/(1+\varepsilon) \le y_{i'_H} \le y_{i'_h}, x_{i'_{H-1}}/(1+\varepsilon) \le y_{i'_{H-1}} \le y_{i'_h}, \cdots, x_{i'_h}/(1+\varepsilon) \le y_{i'_h},$$

the $(H-h+1)$-th smallest value x_{i_h} in $x_1, x_2, \cdots, x_H$ must be at most $(1+\varepsilon)y_{i'_h}$. For $h \in [H]$, if $x_{i_h} \le \theta_h$, then $y_{i'_h}/\theta_h \le x_{i_h}/\theta_h$,

$$y_{i'_h}/\theta_h \ge x_{i_h}/\theta_h/(1+\varepsilon) \ge (1-\varepsilon) \cdot (x_{i_h}/\theta_h) \ge x_{i_h}/\theta_h - \varepsilon,$$

and $\theta_h/y_{i'_h} \ge \theta_h/x_{i_h} \ge 1$. If $x_{i_h} \ge \theta_h$, then $\theta_h/y_{i'_h} \ge \theta_h/x_{i_h}$,

$$\theta_h/y_{i'_h} \le (1+\varepsilon) \cdot \theta_h/x_{i_h} \le \theta_h/x_{i_h} + \varepsilon,$$

and $y_{i'_h}/\theta_h \ge 1/(1+\varepsilon) \ge 1-\varepsilon$. Thus, $f_\theta(y) \in [f_\theta(x) - \varepsilon, f_\theta(x) + \varepsilon]$.

Consider an arbitrary vector $x \in \mathbb{R}^H$ satisfying for any $h \in [H], x_h \in \{2^{-H(2h-1)}, 2^{-2Hh}\}$ and a function $f_\theta \in \mathcal{F}$. It is easy to see that we always have

$$1 \ge x_1 \ge x_2 \ge \cdots \ge x_H \ge 0.$$

If $x = \theta$, then $f_\theta(x) = 1$. Otherwise, consider any $h \in [H]$ such that $x_h \neq \theta_h$. If $x_h = 2^{-H(2h-1)}, \theta_h = 2^{-2Hh}$, then $\min(x_h/\theta_h, \theta_h/x_h) = 2^{-H}$. If $\theta_h = 2^{-H(2h-1)}, x_h = 2^{-2Hh}$, then $\min(x_h/\theta_h, \theta_h/x_h) = 2^{-H}$. Thus, $f_\theta(x) = 2^{-H}$ if $x \neq \theta$.

Now consider a policy π and an objective function $f_\theta \in \mathcal{F}$. Let

$$s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_H, a_H, r_H$$

be the trajectory induced by π. Let $x = (r_1, r_2, \cdots, r_H)$. We know that the optimal policy for f_θ is the policy with $x = \theta$. For any non-optimal policy π we know that $f_\theta(\pi) = 2^{-H}$.

Now we prove the desired result. Our proof is by reduction from INDQ_{2^H}. Suppose we have an algorithm $\mathcal{M}$ which outputs 0.49-optimal policy for any $f \in \mathcal{F}$. We will show that there is a query algorithm for INDQ_{2^H}. In problem INDQ_{2^H}, there is an underlying $\theta^* \in \mathbb{R}^H$ satisfying for any $h \in [H], \theta_h \in \{2^{-H(2h-1)}, 2^{-2Hh}\}$ and we want to find θ^*. We can imagine that the deterministic system has objective function $f_{\theta^*} \in \mathcal{F}$ and then we simulate $\mathcal{M}$. Suppose the i-th query policy of $\mathcal{M}$ is π, then we let $x = (r_1, r_2, \cdots, r_H)$ be the reward values induced by π. By our construction of the deterministic system we have $1 \geq r_1 \geq r_2 \geq \cdots \geq r_H \geq 0$. Then we query whether $x = \theta^*$. If the answer is yes, then we are done. Otherwise, since $f_{\theta^*}(x) = 2^{-H}$ for $x \neq \theta^*$, we return an objective value of 2^{-H} for the i-th query of $\mathcal{M}$ and continue the simulation of $\mathcal{M}$. Since $\mathcal{M}$ can output a 0.49-optimal policy with probability at least 0.9, it must output the optimal policy with probability at least 0.9 which means that it can eventually find $x = \theta^*$ with probability at least 0.9. According to Theorem 5.4.1, $\mathcal{M}$ must query at least $0.9 \cdot 2^H$ policies for the worst $f \in \mathcal{F}$. $\square$

Chapter 8

Reward-Free Exploration with Linear Function Approximation

8.1 Introduction

In RL, an agent repeatedly interacts with an unknown environment to maximize the cumulative reward. To achieve this goal, RL algorithms must be equipped with exploration mechanisms to effectively solve tasks with long horizons and sparse reward signals. Empirically, there is a host of success by combining deep RL methods with different exploration strategies. However, the theoretical understanding of exploration in RL by far is rather limited.

In this section we study the reward-free exploration setting which was formalized in the recent work by [39]. There are two phases in the reward-free setting: the exploration phase and the planning phase. During the exploration phase, the agent collects trajectories from an unknown environment without any pre-specified reward function. Then, in the planning phase, a specific reward function is given to the agent, and the goal is to use samples collected during the exploration phase to output a near-optimal policy for the given reward function. From a practical point of view, this paradigm is particularly suitable for 1) the offline RL setting where data collection and planning are explicitly separated and 2) the setting where there are multiple reward function of interest, e.g., constrained RL [2, 83]. From a theoretical point view, this setting separates the exploration problem and the planning problem which allows one to handle them in a theoretically principled way, in contrast to the standard RL setting where one needs to deal with both problems simultaneously.

Key in this framework is to collect a dataset with sufficiently good coverage over the state space during the exploration phase, so that one can apply a offline RL algorithm on the dataset [3, 6, 16, 62] during the planning phase. For the reward-free exploration setting, existing theoretical works only apply to the tabular RL setting. [39] showed that in the tabular setting where the state space has bounded size, $\widetilde{O}(\mathrm{poly}(|\mathcal{S}||\mathcal{A}|H)/\varepsilon^2)$ samples during the exploration phase is *necessary and sufficient* in order to output ε-optimal policies in the planning phase. Here, $|\mathcal{S}|$ is the number of states, $|\mathcal{A}|$ is the number of actions and H is the planning horizon.

The sample complexity bound in [39], although being near-optimal in the tabular setting, can be unacceptably large in practice due to the polynomial dependency on the size of the state

space. For environments with a large state space, function approximation schemes are needed for generalization. RL with linear function approximation is arguably the simplest yet most fundamental setting. Clearly, in order to understand more general function classes, e.g., deep neural networks, one must understand the class of linear functions first. In this section, we study RL with linear function approximation in the reward-free setting, and our goal is to answer the following question: is it possible to design provably efficient RL algorithms with linear function approximation in the reward-free setting? We obtain both a polynomial upper bound and a hardness result to the above question.

Our Results. Our first contribution is a provably efficient algorithm for reward-free exploration under the linear MDP assumption [40, 102], which, roughly speaking, requires both the transition operators and the reward functions to be linear functions of a d-dimensional feature extractor given to the agent. See Assumption 8.2.1 for the formal statement of the linear MDP assumption. Our algorithm, formally presented in Section 8.3, samples $\widetilde{O}(\mathrm{poly}(H, d, 1/\varepsilon))$ trajectories during the exploration phase, and outputs ε-optimal policies for an arbitrary number of reward functions satisfying Assumption 8.2.1 during the planning phase with high probability. Here d is the feature dimension, H is the planning horizon and ε is the desired accuracy.

One may wonder whether is possible to further weaken the linear MDP assumption, since it requires the feature extractor to encode model information, and such feature extractor might be hard to construct in practice. Our second contribution is a hardness result for reward-free exploration under the linear Q^* assumption, which only requires the optimal value function to be a linear function of the given feature extractor and thus weaker than the linear MDP assumption. Our hardness result shows that under the linear Q^* assumption, any algorithm requires exponential number of samples during the exploration phase, so that the agent could output a near-optimal policy during the planning phase with high probability. The hardness result holds even when the MDP is deterministic.

Our results highlight the following conceptual insights.

- **Reward-free exploration might require the feature to encode model information.** Under model-based assumption (linear MDP assumption), there exists a polynomial sample complexity upper bound for reward-free exploration, while under value-based assumption (linear Q^* assumption), there is an exponential sample complexity lower bound. Therefore, the linear Q^* assumption is *strictly weaker* than the linear MDP assumption in the reward-free setting.
- **Reward-free exploration could be exponentially harder than standard RL.** For deterministic systems, under the assumption that the optimal Q-function is linear, there exists a polynomial sample complexity upper bound [98] in the standard RL setting. However, our hardness result demonstrates that under the same assumption, any algorithm requires exponential number of samples in the reward-free setting.
- **Simulators could be exponentially more powerful.** In the setting where the agent has sampling access to a generative model (a.k.a. simulator) of the MDP, the agent can query the next state s' sampled from the transition operator given any state-action pair as input. In the supplementary material, we show that for deterministic systems, under the linear Q^* assumption, there exists a polynomial sample complexity upper bound in the reward-free

setting when the agent has sampling access to a generative model. Compared with the hardness result above, this upper bound demonstrates an exponential separation between the sample complexity of reward-free exploration in the generative model and that in the standard RL model. To the best our knowledge, this is the first exponential separation between the standard RL model and the generative model for a natural question.

8.2 Notations and Background

8.2.1 Notations

In this chapter, for a specific set of reward functions $r = \{r_h\}_{h=1}^H$ where $r_h : \mathcal{S} \times \mathcal{A} \to [0,1]$ for each $h \in [H]$, given a policy π, a level $h \in [H]$ and a state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, the Q-function is defined as

$$Q_h^\pi(s,a,r) = \mathbb{E}\left[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'}) \mid s_h = s, a_h = a, \pi\right].$$

Similarly, the value function of a given state $s \in \mathcal{S}$ is defined as

$$V_h^\pi(s,r) = \mathbb{E}\left[\sum_{h'=h}^H r_{h'}(s_{h'}, a_{h'}) \mid s_h = s, \pi\right].$$

For a specific set of reward functions $r = \{r_h\}_{h=1}^H$ where $r_h : \mathcal{S} \times \mathcal{A} \to [0,1]$ for each $h \in [H]$, We use π_r^* to denote an optimal policy with respect to r, i.e., π_r^* is a policy that maximizes

$$\mathbb{E}\left[\sum_{h=1}^H r_h(s_h, a_h) \mid \pi\right].$$

We also denote $Q_h^*(s,a,r) = Q_h^{\pi_r^*}(s,a,r)$ and $V_h^*(s,r) = V_h^{\pi_r^*}(s,r)$. We say a policy π is ε-optimal with respect to r if

$$\mathbb{E}\left[\sum_{h=1}^H r_h(s_h, a_h) \mid \pi\right] \geq \mathbb{E}\left[\sum_{h=1}^H r_h(s_h, a_h) \mid \pi_r^*\right] - \varepsilon.$$

Throughout this chapter, when r is clear from the context, we may omit r from $Q_h^\pi(s,a,r)$, $V_h^\pi(s,r)$, $Q_h^*(s,a,r)$, $V_h^*(s,r)$ and π_r^*.

8.2.2 Linear Function Approximation

When applying linear function approximation schemes, it is commonly assumed that the agent is given a feature extractor $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ which can either be hand-crafted or a pre-trained neural network that transforms a state-action pair to a d-dimensional embedding, and the model or the Q-function can be predicted by linear functions of the features. In this section, we consider two different kinds of assumptions: a model-based assumption (linear MDP) and a value-based assumption (linear Q^*).

Linear MDP. The following linear MDP assumption, which was first introduced in [40, 102], states that the model of the MDP can be predicted by linear functions of the given features.

Assumption 8.2.1 (Linear MDP)**.** *An MDP is said to be a linear MDP if the followings hold:*

1. *there are d unknown signed measures $\mu = (\mu^{(1)}, \mu^{(2)}, \ldots, \mu^{(d)})$ such that for any $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, $P(s' \mid s, a) = \langle \mu(s'), \phi(s, a) \rangle$;*
2. *there exists an unknown vector $\eta \in \mathbb{R}^d$ such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $R(s, a) = \langle \phi(s, a), \eta \rangle$.*

As in [40], we assume $\|\phi(s, a)\|_2 \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\mu(\mathcal{S})\|_2 \leq \sqrt{d}$, and $\|\eta\|_2 \leq \sqrt{d}$.

Linear Q^*. The following linear Q^* assumption, which is a common assumption in the theoretical RL literature (see e.g. [24, 26]), states that the optimal Q-function can be predicted by linear functions of the given features.

Assumption 8.2.2 (Linear Q^*)**.** *An MDP M satisfies the linear Q^* assumption if there exist H unknown vectors $\theta_1, \theta_2, \ldots, \theta_H \in \mathbb{R}^d$ such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $Q_h^*(s, a) = \langle \phi(s, a), \theta_h \rangle$. We assume $\|\phi(s, a)\|_2 \leq 1$ and $\|\theta_h\|_2 \leq \sqrt{d}$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $h \in [H]$.*

We note that Assumption 8.2.2 is weaker than Assumption 8.2.1. Under Assumption 8.2.1, it can be shown that for any policy π, $Q_h^\pi(\cdot, \cdot)$ is a linear function of the given feature extractor $\phi(\cdot, \cdot)$. In this section, we show that Assumption 8.2.2 is *strictly weaker* than Assumption 8.2.1 in the reward-free setting, meaning that reward-free exploration under Assumption 8.2.2 is *exponentially* harder than that under Assumption 8.2.1.

8.2.3 Reward-Free Exploration

In the reward-free setting, the goal is to design an algorithm that efficiently explore the state space without the guidance of reward information. Formally, there are two phases in the reward-free setting: *exploration phase* and *planning phase*.

Exploration Phase. During the exploration phase, the agent interacts with the environment for K episodes. In the k-th episode, the agent chooses a policy π^k which induces a trajectory. The agent observes the states and actions $s_1^k, a_1^k, s_2^k, a_2^k, \ldots, s_h^k, a_h^k$ as usual, but does not observe any reward values. After K episodes, the agent collects a dataset of visited state-actions pairs $\mathcal{D} = \{(s_h^k, a_h^k)\}_{(k,h) \in [K] \times [H]}$ which will be used in the planning phase.

Planning Phase. During the planning phase, the agent is no longer allowed to interact with the MDP. Instead, the agent is given a reward function $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the deterministic reward function, and the goal here is to output an ε-optimal policy with respect to R using the collected dataset $\mathcal{D}$.

To measure the performance of an algorithm, we define the *sample complexity* to be the number of episodes K required in the exploration phase to output an ε-optimal policy in the planning phase.

Algorithm 12 Reward-Free Exploration for Linear MDPs: Exploration Phase

Input: Failure probability $\delta > 0$ and target accuracy $\varepsilon > 0$
$\beta \leftarrow c_\beta \cdot dH\sqrt{\log(dH\delta^{-1}\varepsilon^{-1})}$ for some $c_\beta > 0$
$K \leftarrow c_K \cdot d^3H^6\log(dH\delta^{-1}\varepsilon^{-1})/\varepsilon^2$ for some $c_K > 0$
for $k = 1, 2, \ldots K$ **do**
  $Q^k_{H+1}(\cdot,\cdot) \leftarrow 0$ and $V^k_{H+1}(\cdot) = 0$
  for $h = H, H-1, \ldots, 1$ **do**
    $\Lambda^k_h \leftarrow \sum_{\tau=1}^{k-1} \phi(s^\tau_h, a^\tau_h)\phi(s^\tau_h, a^\tau_h)^\top + I$
    $u^k_h(\cdot,\cdot) \leftarrow \min\left\{\beta \cdot \sqrt{\phi(\cdot,\cdot)^\top(\Lambda^k_h)^{-1}\phi(\cdot,\cdot)}, H\right\}$
    Define the exploration-driven reward function $r^k_h(\cdot,\cdot) \leftarrow u^k_h(\cdot,\cdot)/H$
    $w^k_h \leftarrow (\Lambda^k_h)^{-1}\sum_{\tau=1}^{k-1}\phi(s^\tau_h, a^\tau_h)\cdot V^k_{h+1}(s^\tau_{h+1})$
    $Q^k_h(\cdot,\cdot) \leftarrow \min\{(w^k_h)^\top\phi(\cdot,\cdot) + r^k_h(\cdot,\cdot) + u^k_h(\cdot,\cdot), H\}$ and $V^k_h(\cdot) = \max_{a\in\mathcal{A}} Q^k_h(\cdot, a)$
    $\pi^k_h(\cdot) \leftarrow \mathrm{argmax}_{a\in\mathcal{A}} Q^k_h(\cdot, a)$
  Receive initial state $s^k_1 \sim \mu$
  for $h = 1, 2, \ldots H$ **do**
    Take action $a^k_h \leftarrow \pi^k(s^k_h)$ and observe $s^k_{h+1} \sim P(s^k_h, a^k_h)$
return $\mathcal{D} \leftarrow \{(s^k_h, a^k_h)\}_{(k,h)\in[K]\times[H]}$

Algorithm 13 Reward-Free Exploration for Linear MDPs: Planning Phase

Input: Dataset $\mathcal{D} = \{(s^k_h, a^k_h)\}_{(k,h)\in[K]\times[H]}$, reward function R
$Q_{H+1}(\cdot,\cdot) \leftarrow 0$ and $V_{H+1}(\cdot) = 0$
for step $h = H, H-1, \ldots, 1$ **do**
  $\Lambda_h \leftarrow \sum_{\tau=1}^{K}\phi(s^\tau_h, a^\tau_h)\phi(s^\tau_h, a^\tau_h)^\top + I$
  Let $u_h(\cdot,\cdot) \leftarrow \min\left\{\beta \cdot \sqrt{\phi(\cdot,\cdot)^\top(\Lambda_h)^{-1}\phi(\cdot,\cdot)}, H\right\}$
  $w_h \leftarrow (\Lambda_h)^{-1}\sum_{\tau=1}^{K}\phi(s^\tau_h, a^\tau_h)\cdot V_{h+1}(s^\tau_{h+1}, a)$
  $Q_h(\cdot,\cdot) \leftarrow \min\{(w_h)^\top\phi(\cdot,\cdot) + R(\cdot,\cdot) + u_h(\cdot,\cdot), H\}$ and $V_h(\cdot) = \max_{a\in\mathcal{A}} Q_h(\cdot, a)$
  $\pi_h(\cdot) \leftarrow \mathrm{argmax}_{a\in\mathcal{A}} Q_h(\cdot, a)$
Return $\pi = \{\pi_h\}_{h\in[H]}$

8.3 Reward-Free Exploration for Linear MDPs

In this section, we present our reward-free exploration algorithm under the linear MDP assumption. The exploration phase of the algorithm is presented in Algorithm 12, and the planning phase is presented in Algorithm 13.

Exploration Phase. During the exploration phase of the algorithm, we employ the least-square value iteration (LSVI) framework introduced in [40]. In each episode, we first update the parameters (Λ_h, w_h) that are used to calculate the Q-functions, and then execute the greedy policy with respect to the updated Q-function to collect samples. As in [40], to encourage exploration, Algorithm 12 adds an upper-confidence bound (UCB) bonus function u_h.

The main difference between Algorithm 12 and the one in [40] is the definition of the *exploration-driven* reward function. Since the algorithm in [40] is designed for the standard RL setting, the agent can obtain reward values by simply interacting with the environment. On the other hand, in the exploration phase of the reward-free setting, the agent does not have any knowledge about the reward function. In our algorithm, in each episode, we design an exploration-driven reward function which is defined to be $r_h(\cdot,\cdot) = u_h(\cdot,\cdot)/H$, where $u_h(\cdot,\cdot)$ is the UCB bonus function defined in Line 8. Note that we divide $u_h(\cdot,\cdot)$ by H so that $r_h(\cdot,\cdot)$ always lies in $[0,1]$. Intuitively, such a reward function encourages the agent to explore state-action pairs where the amount of uncertainty (quantified by $u_h(\cdot,\cdot)$) is large. After sufficient number of episodes, the uncertainty of all state-action pairs should be low on average, since otherwise the agent would have visited those state-action pairs with large uncertainty as guided by the reward function.

Planning Phase. After the exploration phase, the returned dataset contains sufficient amount of information for the planning phase. In the planning phase (Algorithm 13), for each step $h = H, H-1, \ldots, 1$, we optimize a least squares predictor to predict the Q-function, and return the greedy policy with respect to the predicted Q-function. During the planning phase, we still add an UCB bonus function $u_h(\cdot,\cdot)$ to guarantee optimism and thus correctness of the algorithm. However, as mentioned above and will be made clear in the analysis, since the agent has acquired sufficient information during the exploration phase, $u_h(\cdot,\cdot)$ should be small on average, which implies the returned policy is near-optimal.

8.3.1 Analysis

In this section we present the analysis of our algorithm. We first give the formal theoretical guarantee of our algorithm.

Theorem 8.3.1. *After collecting $O\left(d^3H^6\log(dH\delta^{-1}\varepsilon^{-1})/\varepsilon^2\right)$ trajectories during the exploration phase, with probability $1-\delta$, our algorithm outputs an ε-optimal policy for an arbitrary number of reward functions satisfying Assumption 8.2.1 during the planning phase.*

Now we show how to prove Theorem 8.3.1. Our first lemma shows that the estimated value functions V^k are optimistic with high probability, and the summation of $V_1^k(s_1^k)$ should be small.

Lemma 8.3.2. *With probability $1-\delta/2$, for all $k \in [K]$,*

$$V_1^*(s_1^k, r^k) \le V_1^k(s_1^k)$$

and

$$\sum_{k=1}^{K} V_1^k(s_1^k) \le c\sqrt{d^3H^4K\cdot\log(dKH/\delta)}$$

for some constant $c > 0$ where $V_1^k(\cdot)$ is as defined in Algorithm 12.

Note that the definition of the exploration driven reward function r^k used in the k-th episode depends only on samples collected during the first $k-1$ episodes. Therefore, the first part of the proof is nearly identical to that of Theorem 3.1 in [40]. To prove the second part of the lemma, we first recursively decompose $V_1^k(s_1^k)$ (similar to the standard regret decomposition for optimistic

algorithms), and then use the fact that $r_h(\cdot) = u_h(\cdot)/H$ and the elliptical potential lemma in [1] to given an upper bound on $\sum_{k=1}^K V_1^k(s_1^k)$. The formal proof is provided in the supplementary material.

Our second lemma shows that with high probability, if one divides the bonus function $u_h(\cdot,\cdot)$ (defined in Line 5 in Algorithm 13) by H and uses it as a reward function, then the optimal policy has small cumulative reward on average.

Lemma 8.3.3. *With probability $1-\delta/4$, for the function $u_h(\cdot,\cdot)$ defined in Line 5 in Algorithm 13, we have*

$$\mathbb{E}_{s\sim\mu}\left[V_1^*(s, u_h/H)\right] \le c'\sqrt{d^3H^4\cdot\log(dKH/\delta)/K}$$

for some absolute constant $c' > 0$.

To prove Lemma 8.3.3, we first note that $\mathbb{E}_{s\sim\mu}\left[\sum_{k=1}^K V_1^*(s, r^k)\right]$ is close to $\sum_{k=1}^K V_1^*(s_1^k, r^k)$ by Azuma–Hoeffding inequality and $\sum_{k=1}^K V_1^*(s_1^k, r^k)$ can be bounded by using Lemma 8.3.2. Moreover, for Λ_h defined in Line 4 in Algorithm 13, we have $\Lambda_h \succeq \Lambda_h^k$ for all $k \in [K]$ where Λ_h^k is defined in Line 7 in Algorithm 12, which implies $u_h(\cdot,\cdot)/H \le r_h^k(\cdot,\cdot)$ for all $k \in [K]$. Therefore, we have

$$\mathbb{E}_{s\sim\mu}\left[V_1^*(s, u_h/H)\right] \le \mathbb{E}_{s\sim\mu}\left[V_1^*(s, r^k)\right]$$

for all $k \in [K]$, which implies the desired result.

Our third lemma states the estimated Q-function is always optimistic, and is upper bounded by $R(\cdot,\cdot) + \sum_{s'} P(s' \mid \cdot,\cdot)V_{h+1}(s')$ plus the UCB bonus function $u_h(\cdot,\cdot)$. The lemma can be proved using the same concentration argument as in [40].

Lemma 8.3.4. *With probability $1-\delta/2$, for any reward function R satisfying Assumption 8.2.1 and all $h \in [H]$, we have*

$$Q_h^*(\cdot,\cdot,R) \le Q_h(\cdot,\cdot) \le R(\cdot,\cdot) + \sum_{s'} P(s' \mid \cdot,\cdot)V_{h+1}(s') + 2u_h(\cdot,\cdot).$$

Now we sketch how to prove Theorem 8.3.1 by combining Lemma 8.3.3 and Lemma 8.3.4. Note that With probability $1-\delta$, the events defined in Lemma 8.3.3 and Lemma 8.3.4 both hold. Conditioned on both events, we have

$$\begin{aligned}&\mathbb{E}_{s\sim\mu}[V_1^*(s,R) - V_1^\pi(s,R)] \le \mathbb{E}_{s\sim\mu}[V_1(s) - V_1^\pi(s,R)]\\ \le&\mathbb{E}_{s\sim\mu}[V_1^\pi(s,u)] \le \mathbb{E}_{s\sim\mu}[V_1^*(s,u)] \le c'H\sqrt{d^3H^4\cdot\log(dKH/\delta)/K},\end{aligned}$$

where the first inequality follows by Lemma 8.3.4, the second inequality follows by Lemma 8.3.4 and decomposing the V-function recursively, the third inequality follows by the definition of V^*, and the last inequality follows by Lemma 8.3.3.

8.3.2 Missing Proofs in Section 8.3.1

In this section, for all $(k,h) \in [K]\times[H]$, we denote

$$\phi_h^k := \phi(s_h^k, a_h^k).$$

In Algorithm 12 and 13, we recall that

$$\beta = c_\beta dH\sqrt{\log(dH/\delta/\varepsilon)}.$$

Since $K = c_K \cdot d^3H^6\log(dH\delta^{-1}\varepsilon^{-1})/\varepsilon^2$, we have

$$\beta \geq c_\beta dH\sqrt{\log(dHK/\delta)}$$

for appropriate choices of c_β and c_K.

8.3.2.1 Proof of Lemma 8.3.2

To prove Lemma 8.3.2, we need a concentration lemma similar to Lemma B.3 in [40].

Lemma 8.3.5. *Suppose Assumption 8.2.1 holds. Let $\mathcal{E}$ be the event that for all $(k,h) \in [K]\times[H]$,*

$$\left\|\sum_{\tau=1}^{k-1}\phi_h^\tau\left(V_{h+1}^k(s_{h+1}^\tau) - \sum_{s'\in\mathcal{S}}P(s'|s_h^\tau,a_h^\tau)V_{h+1}^k(s')\right)\right\|_{(\Lambda_h^k)^{-1}} \leq c\cdot dH\sqrt{\log(dKH/\delta)}$$

for some absolute constant $c>0$. Then $\Pr[\mathcal{E}] \geq 1-\delta/4$.

Proof. The proof is nearly identical to that of Lemma B.3 in [40]. The only deference in our case is that we have a different reward functions at different episodes. However, note that in our case

$$r_h^k(\cdot,\cdot) = u_h^k(\cdot,\cdot)/H$$

and hence

$$r_h^k(\cdot,\cdot) + u_h^k(\cdot,\cdot) = (1+1/H)\cdot\min\left\{\beta\cdot\sqrt{\phi(\cdot,\cdot)^\top(\Lambda_h^k)^{-1}\phi(\cdot,\cdot)}, H\right\}.$$

Thus our value function V_{h+1}^k is of the form

$$V(\cdot) := \min\left\{\max_a w^\top\phi(\cdot,a) + \beta\cdot(1+1/H)\cdot\sqrt{\phi(\cdot,a)^\top\Lambda^{-1}\phi(\cdot,a)}, H\right\}$$

for some $\Lambda\in\mathbb{R}^{d\times d}$, and $w\in\mathbb{R}^d$. Therefore, the value function shares exactly the same function class as that in Lemma D.6 in [40]. The rest of the proof follow similarly. □

We are now ready to prove Lemma 8.3.2.

Proof of Lemma 8.3.2. In our proof, we condition on the event $\mathcal{E}$ defined in Lemma 8.3.5, which holds with probability at least $1-\delta/4$. Since $P(s'|s,a) = \phi(s,a)^\top\mu_h(s')$, we have

$$\sum_{s'\in\mathcal{S}}P(s'|s,a)V_{h+1}^k(s') = \phi(s,a)^\top\widetilde{w}_h^k$$

where

$$\widetilde{w}_h^k := \sum_{s'\in\mathcal{S}}\mu_h(s')V_{h+1}^k(s')$$

is an unknown vector. By Assumption 8.2.1, $\sum_{s'\in\mathcal{S}}\mu_h(s')\le\sqrt{d}$. Therefore,

$$\|\widetilde{w}_h^k\|_2\le H\sqrt{d}.$$

We thus have, for all $(h,k)\in[H]\times[K]$ and $(s,a)\in\mathcal{S}\times\mathcal{A}$,

$$\begin{aligned}
&\phi(s,a)^\top w_h^k-\sum_{s'\in\mathcal{S}}P(s'\mid s,a)^\top V_{h+1}^k(s')\\
=&\phi(s,a)^\top(\Lambda_h^k)^{-1}\sum_{\tau=1}^{k-1}\phi_h^\tau\cdot V_{h+1}^k(s_{h+1}^\tau)-\sum_{s'\in\mathcal{S}}P(s'|s,a)V_{h+1}^k(s')\\
=&\phi(s,a)^\top(\Lambda_h^k)^{-1}\left(\sum_{\tau=1}^{k-1}\phi_h^\tau V_{h+1}^k(s_{h+1}^\tau)-\Lambda_h^k\widetilde{w}_h^k\right)\\
=&\phi(s,a)^\top(\Lambda_h^k)^{-1}\left(\sum_{\tau=1}^{k-1}\phi_h^\tau V_{h+1}^k(s_{h+1}^\tau)-\widetilde{w}_h^k-\sum_{\tau=1}^{k-1}\phi_h^\tau(\phi_h^\tau)^\top\widetilde{w}_h^k\right)\\
=&\phi(s,a)^\top(\Lambda_h^k)^{-1}\left(\sum_{\tau=1}^{k-1}\phi_h^\tau\left(V_{h+1}^k(s_{h+1}^\tau)-\sum_{s'}P(s'|s_h^\tau,a_h^\tau)V_{h+1}^k(s')\right)-\widetilde{w}_h^k\right).
\end{aligned}$$

We have,

$$\begin{aligned}
&\left|\phi(s,a)^\top(\Lambda_h^k)^{-1}\left(\sum_{\tau=1}^{k-1}\phi_h^\tau\left(V_{h+1}^k(s_{h+1}^\tau)-\sum_{s'}P(s'|s_h^\tau,a_h^\tau)V_{h+1}^k(s')\right)\right)\right|\\
=&\left|\phi(s,a)^\top(\Lambda_h^k)^{-1/2}(\Lambda_h^k)^{-1/2}\left(\sum_{\tau=1}^{k-1}\phi_h^\tau\left(V_{h+1}^k(s_{h+1}^\tau)-\sum_{s'}P(s'|s_h^\tau,a_h^\tau)V_{h+1}^k(s')\right)\right)\right|\\
\le&\|\phi(s,a)\|_{(\Lambda_h^k)^{-1}}\cdot\left\|\sum_{\tau=1}^{k-1}\phi_h^\tau\left(V_{h+1}^k(s_{h+1}^\tau)-\sum_{s'}P(s'|s_h^\tau,a_h^\tau)V_{h+1}^k(s')\right)\right\|_{(\Lambda_h^k)^{-1}}.
\end{aligned}$$

By Lemma 8.3.5, we have

$$\begin{aligned}
&\left|\phi(s,a)^\top(\Lambda_h^k)^{-1}\left(\sum_{\tau=1}^{k-1}\phi_h^\tau\left(V_{h+1}^k(s_{h+1}^\tau)-\sum_{s'}P(s'|s_h^\tau,a_h^\tau)V_{h+1}^k(s')\right)\right)\right|\\
\le&cdH\sqrt{\log(dKH/\delta)}\cdot\|\phi(s,a)\|_{(\Lambda_h^k)^{-1}}.
\end{aligned}$$

Moreover, we have

$$\left|\phi(s,a)^\top(\Lambda_h^k)^{-1}\widetilde{w}_h^k\right|\le\|\phi(s,a)\|_{(\Lambda_h^k)^{-1}}\cdot\|\widetilde{w}_h^k\|_{(\Lambda_h^k)^{-1}}\le\|\phi(s,a)\|_{(\Lambda_h^k)^{-1}}\cdot H\sqrt{d}.$$

Therefore, we have

$$\begin{aligned}
&\left|\phi(s,a)^\top w_h^k - \sum_{s'\in\mathcal{S}} P(s' \mid s,a) V_{h+1}^k(s')\right| \\
\leq& cdH\sqrt{\log(dKH/\delta)} \cdot \|\phi(s,a)\|_{(\Lambda_h^k)^{-1}} + \|\phi(s,a)\|_{(\Lambda_h^k)^{-1}} \cdot H\sqrt{d} \\
\leq& c_\beta dH\sqrt{\log(dKH/\delta)} \cdot \|\phi(s,a)\|_{(\Lambda_h^k)^{-1}} \\
=& \beta \cdot \|\phi(s,a)\|_{(\Lambda_h^k)^{-1}}.
\end{aligned}$$

Now we prove the first part of the lemma.

First Part. Our proof is by induction on h. Indeed, for $h = H+1$, it holds that for all $s \in \mathcal{S}$,

$$V_{H+1}^*(s, r^k) \leq V_{H+1}^k(s)$$

since $V_{H+1}^* = V_{H+1}^k = 0$. Suppose for some $h \in [H]$, it holds that for all $s \in \mathcal{S}$,

$$V_{h+1}^*(s, r^k) \leq V_{h+1}^k(s).$$

Then we have

$$\begin{aligned}
V_h^*(s, r^k) &= \max_{a\in\mathcal{A}} \left(r_h^k(s,a) + \sum_{s'\in\mathcal{S}} P(s' \mid s,a) V_{h+1}^*(\cdot, r^k) \right) \\
&\leq \max_{a\in\mathcal{A}} \left(r_h^k(s,a) + \sum_{s'\in\mathcal{S}} P(s' \mid s,a) V_{h+1}^k(s', r^k) \right).
\end{aligned}$$

Notice that for all $(s,a) \in \mathcal{S}\times\mathcal{A}$,

$$\sum_{s'\in\mathcal{S}} P(s' \mid s,a)^\top V_{h+1}^k(s', r^k) \leq \phi(s,a)^\top w_h^k + \beta \cdot \|\phi(s,a)\|_{(\Lambda_h^k)^{-1}}.$$

We have

$$V_h^*(s, r^k) \leq \min\left\{ \max_{a\in\mathcal{A}} \left(r_h^k(s,a) + \phi(s,a)^\top w_h^k + \beta \cdot \|\phi(s,a)\|_{(\Lambda_h^k)^{-1}} \right), H \right\} = V_h^k(s)$$

as desired.

Second Part. To prove the second part, for all $(k,h) \in [K] \times [H-1]$, we denote

$$\xi_h^k = \sum_{s'\in\mathcal{S}} P(s' | s_h^k, a_h^k) V_{h+1}^k(s') - V_{h+1}^k(s_{h+1}^k).$$

Conditioned on $\mathcal{E}$,

$$\begin{aligned}
\sum_{k=1}^{K} V_1^k(s_1^k) &\le \sum_{k=1}^{K} \left(r_1^k(s_1^k, a_1^k) + \phi(s_1^k, a_1^k)^\top w_h^k + \beta \cdot \|\phi(s_1^k, a_1^k)\|_{(\Lambda_1^k)^{-1}} \right) \\
&= \sum_{k=1}^{K} \left(\phi(s_1^k, a_1^k)^\top w_h^k + (1 + 1/H) \cdot \beta \cdot \|\phi(s_1^k, a_1^k)\|_{(\Lambda_1^k)^{-1}} \right) \\
&\le \sum_{k=1}^{K} \left(\sum_{s' \in \mathcal{S}} P(s'|s_1^k, a_1^k) V_2^k(s') + (2 + 1/H) \cdot \beta \cdot \|\phi(s_1^k, a_1^k)\|_{(\Lambda_1^k)^{-1}} \right) \\
&\le \sum_{k=1}^{K} \left(\xi_1^k + V_2^k(s_2^k) + (2 + 1/H) \cdot \beta \cdot \|\phi(s_1^k, a_1^k)\|_{(\Lambda_1^k)^{-1}} \right) \\
&\le \dots \\
&\le \sum_{k=1}^{K} \sum_{h=1}^{H-1} \xi_h^k + \sum_{k=1}^{K} \sum_{h=1}^{H} (2 + 1/H) \cdot \beta \cdot \|\phi(s_h^k, a_h^k)\|_{(\Lambda_h^k)^{-1}}.
\end{aligned}$$

Note that for each $h \in [H-1]$, $\{\xi_h^k\}_{k=1}^K$ is a martingale difference sequence with $|\xi_h^k| \le H$. Define $\mathcal{E}'$ to be the even that

$$\left| \sum_{k=1}^{K} \sum_{h=1}^{H-1} \xi_h^k \right| \le c' H^2 \sqrt{K \log(KH/\delta)}.$$

By Azuma–Hoeffding inequality, we have $\Pr[\mathcal{E}'] \ge 1 - \delta/4$.

Next, we have,

$$\sum_{k=1}^{K} \sum_{h=1}^{H} \|\phi(s_h^k, a_h^k)\|_{(\Lambda_h^k)^{-1}} \le \sqrt{KH \sum_{k=1}^{K} \sum_{h=1}^{H} \phi(s_h^k, a_h^k)^\top (\Lambda_h^k)^{-1} \phi(s_h^k, a_h^k)}.$$

By Lemma D.2 in [40], we have

$$\sum_{h=1}^{H} \sum_{k=1}^{K} \phi(s_h^k, a_h^k)^\top (\Lambda_h^k)^{-1} \phi(s_h^k, a_h^k) \le 2dH \log(K).$$

Conditioned on $\mathcal{E} \cap \mathcal{E}'$ which holds with probability at least $1 - \delta/2$, we have

$$\begin{aligned}
\sum_{k=1}^{K} V_1^k(s_1^k) &\le c' H^2 \sqrt{K \log(KH/\delta)} + (2 + 1/H) \cdot \beta \cdot \sqrt{KH \cdot 2dH \log(K)} \\
&\le c \sqrt{d^3 H^4 K \cdot \log(dKH/\delta)}
\end{aligned}$$

for some absolute constant $c > 0$. □

8.3.2.2 Proof of Lemma 8.3.3

Proof of Lemma 8.3.3. We denote $\Delta^k = V_1^*(s_1^k, r^k) - \mathbb{E}_{s\sim\mu}[V_1^*(s, r^k)]$. Since r^k depends only on data collected during the first $k-1$ episodes, $\{\Delta^k\}_{k=1}^K$ is a martingale difference sequence. Moreover, $|\Delta^k| \leq H$ almost surely. Thus, by Azuma-Hoeffding inequality, we have, with probability at least $1 - \delta/8$, there exists an absolute constant $c_1 > 0$, such that

$$\left|\sum_{k=1}^K \Delta^k\right| \leq c_1 H\sqrt{K\log(1/\delta)},$$

which we condition on in the rest of the proof. Therefore, we have,

$$\mathbb{E}_{s\sim\mu}\left[\sum_{k=1}^K V_1^*(s, r^k)\right] \leq \sum_{k=1}^K V_1^*(s, r^k) + c_1 H\sqrt{K\log(1/\delta)}.$$

Next, we notice that for all $k \in [K]$,

$$\Lambda_h \succeq \Lambda_h^k.$$

Hence we have for all $(k, h) \in [K] \times [H]$,

$$r_h^k(\cdot,\cdot) \geq u_h(\cdot,\cdot)/H.$$

Hence

$$V_1^*(\cdot, u_h/H) \leq V_1^*(\cdot, r_h^k).$$

Together with Lemma 8.3.2, we have

$$\begin{aligned}\mathbb{E}_{s\sim\mu}\big[V_1^*(s, u_h/H)\big] &\leq \mathbb{E}_{s\sim\mu}\left[\sum_{k=1}^K V_1^*(s, r^k)/K\right] \leq K^{-1}\sum_{k=1}^K V_1^*(s_1^k, r^k) + c_1 H\sqrt{\log(1/\delta)/K} \\ &\leq c'\sqrt{d^3H^4\cdot\log(dKH/\delta)/K}\end{aligned}$$

for some absolute constant $c' > 0$. □

8.3.2.3 Proof of Lemma 8.3.4

Proof of Lemma 8.3.4. Using the same argument in the proof of Lemma 8.3.2, with probability at least $1 - \delta/4$, for all $h \in [H]$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\left|\phi(s,a)^\top w_h - \sum_{s'\in\mathcal{S}} P(s' \mid s, a)V_{h+1}(s')\right| \leq \beta\cdot\|\phi(s,a)\|_{(\Lambda_h)^{-1}}.$$

Therefore, for all $h \in [H]$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\begin{aligned}Q_h(s,a) &\leq (w_h)^\top\phi(s,a) + R(s,a) + u_h(s,a) \\ &\leq R(s,a) + \sum_{s'\in\mathcal{S}} P(s' \mid s, a)V_{h+1}(s') + 2\beta\cdot\|\phi(s,a)\|_{(\Lambda_h)^{-1}}.\end{aligned}$$

Moreover, $Q_h(s,a) \le H$. Since $u_h(\cdot,\cdot) = \min\left\{\beta \cdot \sqrt{\phi(\cdot,\cdot)^\top (\Lambda_h)^{-1}\phi(\cdot,\cdot)}, H\right\}$, we have

$$Q_h(s,a) \le R(s,a) + \sum_{s'} P(s' \mid s,a) V_{h+1}(s') + 2u_h(s,a).$$

Now we prove for all $h \in [H]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$, $Q_h^*(s,a,R) \le Q_h(s,a)$. We prove by induction on h. When $h = H+1$ this is clearly true. Suppose for some $h \in [H]$, $Q_{h+1}^*(s,a,R) \le Q_{h+1}(s,a)$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. We have

$$Q_h(s,a) = \min\{(w_h)^\top \phi(s,a) + R(s,a) + u_h(s,a), H\}.$$

Since $Q_{h+1}^*(s,a,R) \le H$ and $u_h(\cdot,\cdot) = \min\left\{\beta \cdot \sqrt{\phi(\cdot,\cdot)^\top (\Lambda_h)^{-1}\phi(\cdot,\cdot)}, H\right\}$, it suffices to prove that

$$Q_{h+1}^*(s,a,R) \le (w_h)^\top \phi(s,a) + R(s,a) + \beta \cdot \|\phi(s,a)\|_{(\Lambda_h)^{-1}}.$$

By the induction hypothesis,

$$\begin{aligned}
\phi(s,a)^\top w_h &\ge \sum_{s' \in \mathcal{S}} P(s' \mid s,a) V_{h+1}(s') - \beta \cdot \|\phi(s,a)\|_{(\Lambda_h)^{-1}} \\
&\ge \sum_{s' \in \mathcal{S}} P(s' \mid s,a) V_{h+1}^*(s',r) - \beta \cdot \|\phi(s,a)\|_{(\Lambda_h)^{-1}}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
Q_h^*(s,a,R) &= R(s,a) + \sum_{s' \in \mathcal{S}} P(s' \mid s,a) V_{h+1}^*(s',R) \\
&\ge (w_h)^\top \phi(s,a) + R(s,a) + \beta \cdot \|\phi(s,a)\|_{(\Lambda_h)^{-1}}.
\end{aligned}$$

□

8.3.2.4 Proof of Theorem 8.3.1

Proof of Theorem 8.3.1. In our proof we condition on the events defined in Lemma 8.3.3 and Lemma 8.3.4 which hold with probability at least $1-\delta$. By Lemma 8.3.4, for any $s \in \mathcal{S}$,

$$V_1(s) = \max_{a \in \mathcal{A}} Q_1(s,a) \ge \max_{a \in \mathcal{A}} Q_1^*(s,a,R) = V_1^*(s,R),$$

which implies

$$\mathbb{E}_{s_1 \sim \mu}[V_1^*(s_1,R) - V_1^\pi(s_1,R)] \le \mathbb{E}_{s_1 \sim \mu}[V_1(s_1) - V_1^\pi(s_1,R)].$$

Note that

$$\begin{aligned}
&\mathbb{E}_{s_1 \sim \mu}[V_1(s_1) - V_1^\pi(s_1,R)] \\
=&\mathbb{E}_{s_1 \sim \mu}[Q(s_1,\pi_1(s_1)) - Q_1^\pi(s_1,\pi_1(s_1),R)] \\
=&\mathbb{E}_{s_1 \sim \mu, s_2 \sim P(\cdot \mid s_1, \pi_1(s_1))}[R(s_1,\pi_1(s_1)) + V_2(s_2) + u_1(s_1,\pi(s_1)) - R(s_1,\pi_1(s_1)) - V_2^\pi(s_2)] \\
=&\mathbb{E}_{s_1 \sim \mu, s_2 \sim P(\cdot \mid s_1, \pi_1(s_1))}[V_2(s_2) + u_1(s_1,\pi(s_1)) - V_2^\pi(s_2)] \\
=&\mathbb{E}_{s_1 \sim \mu, s_2 \sim P(\cdot \mid s_1, \pi_1(s_1)), s_3 \sim P(\cdot, \mid s_2, \pi_2(s_2))}[u_1(s_1,\pi(s_1)) + u_2(s_2,\pi(s_2)) + V_3(s_3) - V_3^\pi(s_3)] \\
=&\dots \\
=&\mathbb{E}_{s \sim \mu}[V_1^\pi(s,u)].
\end{aligned}$$

By definition of $V_1^*(s,u)$, we have

$$\mathbb{E}_{s\sim\mu}[V_1^{\pi}(s,u)] \le \mathbb{E}_{s\sim\mu}[V_1^*(s,u)].$$

By Lemma 8.3.3,

$$\mathbb{E}_{s\sim\mu}[V_1^*(s,u)] = H\cdot\mathbb{E}_{s\sim\mu}[V_1^*(s,u/H)] \le c'H\sqrt{d^3H^4\cdot\log(dKH/\delta)/K}.$$

By taking $K = c_K\cdot d^3H^6\log(dH\delta^{-1}\varepsilon^{-1})/\varepsilon^2$ for a sufficiently large constant $c_K > 0$, we have

$$\mathbb{E}_{s_1\sim\mu}[V_1^*(s_1,R) - V_1^{\pi}(s_1,R)] \le H\cdot\mathbb{E}_{s\sim\mu}[V_1^*(s,u/H)] \le c'H\sqrt{d^3H^4\cdot\log(dKH/\delta)/K} \le \varepsilon,$$

which implies π is ε-optimal with respect to R. □

8.4 Lower Bound for Reward-Free Exploration under Linear Q^* Assumption

Now we focus on hardness results for reward-free exploration under the linear Q^* assumption. We show that there exists a class of MDPs which satisfies Assumption 8.2.2, such that any reward-free exploration algorithm requires exponential number of samples during the exploration phase in order to find a near-optimal policy during the planning phase. In particular, we prove the following theorem.

Theorem 8.4.1. *There exists a class of deterministic systems that satisfy Assumption 8.2.2 with $d = \mathrm{poly}(H)$, such that any reward-free algorithm requires at least $\Omega(2^H)$ samples during the exploration phase in order to find a 0.1-optimal policy with probability at least 0.9 during the planning phase for a given reward function R.*

Since deterministic systems are special cases of general MDPs, the hardness result in Theorem 8.4.1 applies to general MDPs as well. In the remaining part of this section, we describe the construction of the hard instance and outline the proof of Theorem 8.4.1.

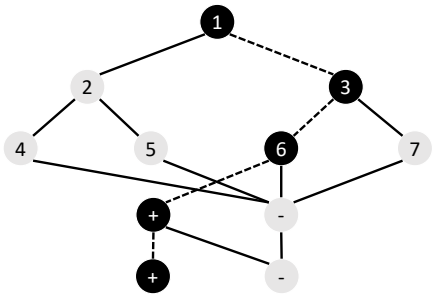


Figure 8.1: An illustration of the hard instance with $H = 5$. Black states and dashed transitions are those on the optimal trajectory $s_1^*, a_1^*, s_2^*, a_2^*, \ldots, s_{H-1}^*, a_{H-1}^*, s_H^*, a_H^*$.

State Space and Action Space. In the hard instance, there are H levels of states

$$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \cup \mathcal{S}_H$$

where $\mathcal{S}_h$ contains all states that can be reached in level h. The action space $\mathcal{A} = \{0, 1\}$. For each $h \in [H-2]$, we represent each state in $\mathcal{S}_h$ by an integer in $[2^{h-1}, 2^h)$, i.e., $\mathcal{S}_1 = \{1\}$, $\mathcal{S}_2 = \{2, 3\}$, $\mathcal{S}_3 = \{4, 5, 6, 7\}$, etc. We also have $S_{H-1} = \{s_{H-1}^+, s_{H-1}^-\}$ and $S_H = \{s_H^+, s_H^-\}$. The initial states is $1 \in \mathcal{S}_1$.

Transition. For each $h \in [H-3]$, for each $s \in \mathcal{S}_h$, $P(s, a)$ is fixed and thus known to the algorithm. In particular, for each $h \in [H-3]$, for each $s \in \mathcal{S}_h$, we define $P(s, a) = 2s+a \in \mathcal{S}_{h+1}$ where $a \in \{0, 1\}$. We will define the transition operator for those states $s \in \mathcal{S}_{H-2} \cup \mathcal{S}_{H-1}$ shortly.

Feature Extractor. For each $h \in [H-2]$, for each $(s, a) \in \mathcal{S}_h \times \mathcal{A}$, we define $\phi(s, a) \in \mathbb{R}^d$ so that $\|\phi(s, a)\|_2 = 1$ and for any $(s', a') \in \mathcal{S}_h \times \mathcal{A} \setminus \{(s, a)\}$, we have $|(\phi(s, a))^\top \phi(s', a')| \leq 0.01$. In the formal proof, we use the Johnson–Lindenstrauss Lemma [41] to show that such feature extractor exists if $d = \mathrm{poly}(H)$.

For all states $s \in \mathcal{S}_{H-1}$, we define

$$\phi(s, a) = \begin{cases} [1, 0, 0, \ldots, 0]^\top & s = s_{H-1}^+, a = 0 \\ [0, 1, 0, \ldots, 0]^\top & s = s_{H-1}^+, a = 1 \\ [0, 0, 0, \ldots, 0]^\top & s = s_{H-1}^- \end{cases}.$$

Finally, for all states $s \in \mathcal{S}_H$, we define

$$\phi(s, a) = \begin{cases} [1, 0, 0, \ldots, 0]^\top & s = s_H^+, a = 0 \\ [0, 0, 0, \ldots, 0]^\top & \text{otherwise} \end{cases}.$$

The Hard MDPs. By Yao's minimax principle [104], to prove a lower bound for randomized algorithms, it suffices to define a hard distribution and show that any deterministic algorithm fails for the hard distribution. We now define the hard distribution. We first define the transition operator $P(s, a)$ for those states $s \in \mathcal{S}_{H-2}$. To do this, we first pick a state-action pair (s_{H-2}^*, a_{H-2}^*) from $\mathcal{S}_{H-2} \times \mathcal{A}$ uniformly at random, and define

$$P(s, a) = \begin{cases} s_{H-1}^+ & s = s_{H-2}^*, a = a_{H-2}^* \\ s_{H-1}^- & \text{otherwise} \end{cases}.$$

To define the transition function $P(s, a)$ for those states $s \in \mathcal{S}_{H-1}$, we pick a random action a_{H-1}^* from $\{0, 1\}$ uniformly at random, and define

$$P(s, a) = \begin{cases} s_H^+ & s = s_{H-1}^+, a = a_{H-1}^* \\ s_H^- & \text{otherwise} \end{cases}.$$

The Reward Function. We now define the optimal Q-function which automatically implies a reward function R. During the planning phase, the agent will receive R as the reward function. By construction, there exists a unique trajectory

$$s_1^*, a_1^*, s_2^*, a_2^*, \ldots, s_{H-1}^*, a_{H-1}^*, s_H^*, a_H^*$$

with $(s_H^*, a_H^*) = (s_H^+, 0)$. For each $h \in [H]$, we define θ_h in Assumption 8.2.2 as $\phi(s_h^*, a_h^*)/2$. This implies that for each $(s, a) \in \mathcal{S}_H \times \mathcal{A}$,

$$R(s,a) = Q_H^*(s,a) = \begin{cases} 0.5 & s = s_H^*, a = a_H^* \\ 0 & \text{otherwise} \end{cases}.$$

For each $(s, a) \in \mathcal{S}_{H-1} \times \mathcal{A}$, we have

$$Q_{H-1}^*(s,a) = \begin{cases} 0.5 & s = s_{H-1}^*, a = a_{H-1}^* \\ 0 & \text{otherwise} \end{cases},$$

which implies that $R(s, a) = 0$ for all $(s, a) \in \mathcal{S}_{H-1} \times \mathcal{A}$. Now for each $h \in [H-2]$, for each $(s, a) \in \mathcal{S}_h \times \mathcal{A}$, we define

$$R(s_h, a_h) = Q_h^*(s_h, a_h) - \max_{a \in \mathcal{A}} Q_{h+1}^*(P(s_h, a_h), a)$$

so that the Bellman equations hold. Moreover, by construction, for each $h \in [H]$, we have $Q_h^*(s, a) = 0.5$ when $(s, a) = (s_h^*, a_h^*)$, and $|Q_h^*(s, a)| \leq 0.01$ when $(s, a) \neq (s_h^*, a_h^*)$ and thus $R(\cdot, \cdot) \in [-0.02, 0.5]$.[1]

Proof of Hardness. Now we sketch the final proof of the hardness result. We define $\mathcal{E}$ to be the event that for all $(s, a) \in \mathcal{D}$ where $\mathcal{D}$ are the state-action pairs collected by the algorithm, we have $s \neq s_{H-1}^* = s_{H-1}^+$. For any deterministic algorithm, we claim that if the algorithm samples at most $2^H/100$ trajectories during the exploration phase, with probability at least 0.9 over the randomness of the distribution of MDPs, $\mathcal{E}$ holds. This is because the feature extractor is fixed and thus the algorithm receives the same feedback before reaching s_{H-1}^+. Since there are 2^{H-2} state-action pairs $(s, a) \in \mathcal{S}_{H-2} \times \mathcal{A}$ and only one of them satisfies $P(s, a) = s_{H-1}^+$, and the algorithm samples at most $2^H/100$ trajectories during the exploration phase, $\mathcal{E}$ holds with probability at least 0.9.

Now during the planning phase, by construction of the optimal Q-function, the only 0.1-optimal policy is $\pi_h(s_h^*) = a_h^*$. However, conditioned on $\mathcal{E}$, any deterministic algorithm correctly output $\pi_{H-1}(s_{H-1}^*) = a_{H-1}^*$ with probability at most 0.5, since conditioned on $\mathcal{E}$, $\mathcal{D}$ does not contain s_{H-1}^*, and the reward function R also does not depend on a_{H-1}^*. Therefore, during the planning phase of the algorithm, a 0.1-optimal policy is found with probability at most $0.6 < 0.9$.

[1]Note that this is slightly different from the assumption that $R(\cdot, \cdot) \in [0, 1]$. However, this can be readily fixed by shifting all reward values by 0.02.

8.4.1 Missing Proofs

In the hard instance construction, for each $h \in [H-2]$, for each $(s,a) \in \mathcal{S}_h \times \mathcal{A}$, we define $\phi(s,a) \in \mathbb{R}^d$ so that $\|\phi(s,a)\|_2 = 1$ and for any $(s',a') \in \mathcal{S}_h \times \mathcal{A} \setminus \{(s,a)\}$, we have $|(\phi(s,a))^\top \phi(s',a')| \leq 0.01$. The following lemma demonstrates the existence of such feature extractor.

Lemma 8.4.2. *There exists a set of vectors $\{\phi_1, \phi_2, \ldots, \phi_{2^H}\} \subset \mathbb{R}^d$ with $d = \mathrm{poly}(H)$ such that*

1. *$\|\phi_i\| = 1$ for all $i \in [2^H]$;*
2. *$|\phi_i^\top \phi_j| \leq 0.01$ for all $i, j \in [2^H]$ with $i \neq j$.*

Proof. This is a direct implication of Lemma 5.4.2 by setting $n = 2^H$ and $\varepsilon = 0.01$. □

Note that the above lemma implies the existence of the required feature exactor, since for each $h \in [H-2]$, there are less than 2^H state-action pairs in $\mathcal{S}_h \times \mathcal{A}$. We simply define the feature of the i-th state-action pair in $\mathcal{S}_h \times \mathcal{A}$ to be ϕ_i in the above lemma.

Proof of Theorem 8.4.1. In order to prove Theorem 8.4.1, by Yao's minimax principle [104], it suffices to prove that for the hard distribution constructed in Section 8.4, for any deterministic algorithm $\mathcal{A}$ that samples at most $2^H/100$ trajectories during the exploration phase, the probability (over the randomness of the hard distribution) that $\mathcal{A}$ outputs a 0.1-optimal policy in the planning phase is at most 0.9.

We first show that for the deterministic algorithm $\mathcal{A}$, among all the 2^{H-2} choices for (s^*_{H-2}, a^*_{H-2}), s^+_{H-1} is in the collected dataset $\mathcal{D}$ for at most $2^H/100$ choices for (s^*_{H-2}, a^*_{H-2}) during the exploration phase. Note that whenever $(s_{H-2}, a_{H-2}) \neq (s^*_{H-2}, a^*_{H-2})$, we must have $s_{H-1} = s^-_{H-1}$ and $s_H = s^-_H$. Therefore, the feedback received by $\mathcal{A}$ is always the same unless $(s_{H-2}, a_{H-2}) = (s^*_{H-2}, a^*_{H-2})$. However, since $\mathcal{A}$ samples at most $2^H/100$ trajectories during the exploration phase, there are most $2^H/100$ choices for (s^*_{H-2}, a^*_{H-2}) during the exploration phase for which s^+_{H-1} is in the collected dataset $\mathcal{D}$.

Recall that $\mathcal{A}$ is deterministic. For any choice of (s^*_{H-2}, a^*_{H-2}), if s^+_{H-1} is not in the collected dataset $\mathcal{D}$, the collected dataset $\mathcal{D}$ is always the same, no matter $a^*_{H-1} = 0$ or $a^*_{H-1} = 1$. Moreover, for any fixed choice of (s^*_{H-2}, a^*_{H-2}), it can be verified that the reward function R does not depend on the choice of a^*_{H-1}. Note that during the planning phase, algorithm $\mathcal{A}$ deterministically maps the collected dataset $\mathcal{D}$ and the reward function R to a policy. Furthermore, the only 0.1-optimal policy must satisfy $\pi(s^*_h) = a^*_h$. However, for any choice of (s^*_{H-2}, a^*_{H-2}), if s^+_{H-1} is not in the collected dataset $\mathcal{D}$, $\pi(s^*_{H-1})$ does not depend on a^*_{H-1} since both the collected dataset $\mathcal{D}$ and the reward function R does not depend on a^*_{H-1}. Therefore, for those choices of (s^*_{H-2}, a^*_{H-2}), $\mathcal{A}$ outputs a 0.1-optimal policy with probability at most 0.5. Therefore, the probability that $\mathcal{A}$ outputs a 0.1-optimal policy is at most

$$\frac{2^H/100}{2^{H-2}} + \left(1 - \frac{2^H/100}{2^{H-2}}\right)/2 \leq 0.6.$$

□

Part IV

Conclusion

Chapter 9

Conclusion and Future Directions

In this thesis, to build a better understanding of modern RL methods, we studied three challenges in RL problems. Below we describe a list of interesting questions that remain open given our results.

Theory of RL. While we have made progress towards understanding the theory of RL, there are still many unsolved fundamental questions. For example, although we have shown that tabular RL is possible with a sample complexity that is independent of the planning horizon, to achieve such a result, the sample complexity will be exponential in the number of states. Is that possible to design a tabular RL algorithm whose sample complexity is completely independent of the planning horizon and also depends only polynomially on the number of states, or there are statistical limits that prevent us from doing that? For RL with large state spaces, all existing positive results rely on assumptions that are hard to verify in practice. Is that possible to design algorithms that only rely on assumptions that can be easily verified? Answering such a question might require first understanding the properties of features learned by practical RL algorithms.

Theoretically-Principled RL Systems and Benchmark Suites. We plan to build more efficient and more robust RL systems and benchmark suites based on the theoretical insights. Currently, we are investigating how to design a better representation learning process for RL. Existing RL algorithms largely treat deep neural networks as black-boxes and use the same set of training algorithms as in supervised learning. However, as demonstrated in this thesis, the definition of a good representation in RL could be significantly different from that in supervised learning. Thus, better representation learning methods could be beneficial for practical RL systems. Moreover, many existing offline RL datasets are collected under a distribution that contains a large fraction from the target policy itself. As demonstrated by our theoretical analysis and experimental results, such a dataset may substantially limit the methodology to only testing algorithms in a low distribution shift regime. We are currently building a new dataset that has example transitions from a diverse set of states, while trajectories in the dataset do not resemble the target policy. With such a dataset, we can now test existing algorithms in a more realistic setting. We believe such a benchmark suite could be beneficial for future offline RL research.

Bibliography

[1] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvári. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Proceedings of the AISTATS*, pages 1–9, 2012. 8.3.1

[2] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *Proceedings of the ICML*, pages 22–31, 2017. 8.1

[3] A. Agarwal, S. M. Kakade, and L. F. Yang. Model-based reinforcement learning with a generative model is minimax optimal. In *Proceedings of the COLT*, pages 67–83, 2020. 5.1, 8.1

[4] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 4.1

[5] N. Alon. Perturbed identity matrices have high rank: Proof and applications. *Combinatorics, Probability and Computing*, 18(1-2):3–15, 2009. 5.2.3

[6] A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129, 2008. 8.1

[7] A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k-support norm. In *Proceedings of the NIPS*, volume 25, 2012. 7.1.1

[8] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Proceedings of the NeurIPS*, volume 32, 2019. 1.2

[9] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *Proceedings of the ICML*, pages 322–332, 2019. 1.2

[10] S. Arora, S. S. Du, Z. Li, R. Salakhutdinov, R. Wang, and D. Yu. Harnessing the power of infinitely wide deep nets on small-data tasks. In *Proceedings of the ICLR*, 2020. 1.2

[11] M. G. Azar, I. Osband, and R. Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the ICML*, pages 263–272, 2017. 4.1, 4.4.1

[12] V. Borkar and R. Jain. Risk-constrained markov decision processes. *IEEE Transactions on Automatic Control*, 59(9):2574–2579, 2014. 7.1

[13] S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996. 6.5.1

[14] S. Bubeck, N. Cesa-Bianchi, and S. M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Proceedings of the COLT*, pages 41.1–41.14, 2012. 3

[15] Q. Cai, Z. Yang, C. Jin, and Z. Wang. Provably efficient exploration in policy optimization. In *Proceedings of the ICML*, pages 1283–1294, 2020. 4.1

[16] J. Chen and N. Jiang. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the ICML*, pages 1042–1051, 2019. 5.2.1, 5.2.1, 6.1, 6.4, 8.1

[17] Y. Chow, A. Tamar, S. Mannor, and M. Pavone. Risk-sensitive and robust decision-making: a CVaR optimization approach. In *Proceedings of the NIPS*, volume 28, 2015. 7.1

[18] K. L. Clarkson, R. Wang, and D. P. Woodruff. Dimensionality reduction for tukey regression. In *Proceedings of the ICML*, pages 1262–1271, 2019. 1.2

[19] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003. 5.4.3

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL*, pages 4171–4186, 2019. 6.1

[21] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the SODA*, pages 1127–1136, 2006. 4.4.3

[22] S. S. Du, K. Hou, R. Salakhutdinov, B. Póczos, R. Wang, and K. Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Proceedings of the NeurIPS*, volume 32, 2019. 1.2

[23] S. S. Du, S. M. Kakade, J. D. Lee, S. Lovett, G. Mahajan, W. Sun, and R. Wang. Bilinear classes: A structural framework for provable generalization in RL. In *Proceedings of the ICML*, pages 2826–2836, 2021. 1.2

[24] S. S. Du, S. M. Kakade, R. Wang, and L. F. Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *Proceedings of the ICLR*, 2020. 2, 8.2.2

[25] S. S. Du, J. D. Lee, G. Mahajan, and R. Wang. Agnostic Q-learning with function approximation in deterministic systems: Near-optimal bounds on approximation error and sample complexity. In *Proceedings of the NeurIPS*, volume 33, pages 22327–22337, 2020. 1.2, 4.1

[26] S. S. Du, Y. Luo, R. Wang, and H. Zhang. Provably efficient Q-learning with function approximation via distribution shift error checking oracle. In *Proceedings of the NeurIPS*, volume 32, 2019. 1.2, 4.1, 5.2, 8.2.2

[27] Y. Duan, Z. Jia, and M. Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *Proceedings of the ICML*, pages 2701–2709, 2020. 6.4, 6.4

[28] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proceedings of the STOC*, pages 569–578, 2011. 4.3.1

[29] D. Feldman, M. Schmidt, and C. Sohler. Turning big data into tiny data: Constant-

size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020. 4.3.1

[30] F. Feng, R. Wang, W. Yin, S. S. Du, and L. F. Yang. Provably efficient exploration for reinforcement learning using unsupervised learning. In *Proceedings of the NeurIPS*, volume 33, pages 22492–22504, 2020. 1.2

[31] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the ICML*, pages 1587–1596, 2018. 6.5.1

[32] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration. In *Proceedings of the ICML*, pages 2052–2062, 2019. 6.1, 6.1

[33] G. J. Gordon. *Approximate Solutions to Markov Decision Processes*. PhD thesis, Carnegie Mellon University, 1999. 6.1

[34] X. Guo, L. Ye, and G. Yin. A mean–variance optimization problem for discounted markov decision processes. *European Journal of Operational Research*, 220(2):423–429, 2012. 7.1

[35] D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. In *Proceedings of the COLT*, pages 9.1–9.24, 2012. 2

[36] D. Hsu, S. M. Kakade, and T. Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17:1–6, 2012. 6.4

[37] N. Jiang and A. Agarwal. Open problem: The dependence of sample complexity lower bounds on planning horizon. In *Proceedings of the COLT*, pages 3395–3398, 2018. 1.1.1, 1.1.1, 3.1, 3.1, 3.1, 3.2

[38] N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire. Contextual decision processes with low bellman rank are PAC-learnable. In *Proceedings of the ICML*, pages 1704–1713, 2017. 5.1

[39] C. Jin, A. Krishnamurthy, M. Simchowitz, and T. Yu. Reward-free exploration for reinforcement learning. In *Proceedings of the ICML*, pages 4870–4879, 2020. 1.1.3, 8.1

[40] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of the COLT*, pages 2137–2143, 2020. 4.1, 4.2, 4.4.2, 4.4, 4.4.3, 8.1, 8.2.2, 8.2.1, 8.3, 8.3.1, 8.3.1, 8.3.2.1, 8.3.2.1, 8.3.2.1

[41] W. B. Johnson, , and J. Lindenstrauss. Extensions of lipschitz mappings into hilbert space. *Contemporary mathematics*, 26:189–206, 1984. 5.2.3, 8.4

[42] S. M. Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University of London, 2003. 2, 3.3.1

[43] M. Kearns, Y. Mansour, and A. Y. Ng. Approximate planning in large POMDPs via reusable trajectories. In *Proceedings of the NIPS*, volume 12, 2000. 3.3.1

[44] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002. 2

[45] R. Kidambi, A. Rajeswaran, P. Netrapalli, and T. Joachims. MOReL: Model-based offline reinforcement learning. In *Proceedings of the NeurIPS*, volume 33, pages 21810–21823,

2020. 6.1, 6.1

[46] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013. 1

[47] K. R. Koedinger, E. Brunskill, R. S. Baker, E. A. McLaughlin, and J. Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013. 1

[48] A. Kumar, J. Fu, G. Tucker, and S. Levine. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *Proceedings of the NeurIPS*, volume 32, 2019. 6.1, 6.1

[49] M. Langberg and L. J. Schulman. Universal ε-approximators for integrals. In *Proceedings of the SODA*, pages 598–607, 2010. 4.3.1

[50] R. Laroche, P. Trichelair, and R. Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *Proceedings of the ICML*, pages 3652–3661, 2019. 6.1, 6.1

[51] Y. Li, R. Wang, and D. P. Woodruff. Tight bounds for the subspace sketch problem with applications. *SIAM Journal on Computing*, 50(4):1287–1335, 2021. 1.2

[52] Y. Li, R. Wang, and L. F. Yang. Settling the horizon-dependence of sample complexity in reinforcement learning. *arXiv preprint arXiv:2111.00633*, 2021. 1

[53] Y. Li, R. Wang, L. F. Yang, and H. Zhang. Nearly linear row sampling algorithm for quantile regression. In *Proceedings of the ICML*, pages 5979–5989, 2020. 1.2

[54] G. Lorentz. Metric entropy and approximation. *Bulletin of the American Mathematical Society*, 72(6):903–937, 1966. 5.4.2

[55] T. Mandel, Y.-E. Liu, S. Levine, E. Brunskill, and Z. Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the AAMAS*, pages 1077–1084, 2014. 6.1

[56] S. Mannor and J. N. Tsitsiklis. Algorithmic aspects of mean–variance optimization in markov decision processes. *European Journal of Operational Research*, 231(3):645–653, 2013. 7.1

[57] A. M. McDonald, M. Pontil, and D. Stamos. Spectral k-support norm regularization. In *Proceedings of the NIPS*, volume 27, 2014. 7.1.1

[58] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 1, 1.1.2, 4.1, 5.1, 6.1, 6.5.1

[59] A. Modi, N. Jiang, A. Tewari, and S. Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *Proceedings of the AISTATS*, pages 2010–2020, 2020. 4.1

[60] T. M. Moldovan and P. Abbeel. Risk aversion in markov decision processes via near optimal chernoff bounds. In *Proceedings of the NIPS*, volume 25, 2012. 7.1

[61] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the ICML*, pages 799–806, 2010. 7.1

[62] R. Munos and C. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(27):815–857, 2008. 6.1, 6.4, 8.1

[63] I. Osband and B. Van Roy. Model-based reinforcement learning and the eluder dimension. In *Proceedings of the NIPS*, volume 27, 2014. 4.2

[64] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the NAACL*, pages 2227–2237, 2018. 6.1

[65] L. Prashanth. Policy gradients for CVaR-constrained MDPs. In *Proceedings of the ALT*, pages 155–169, 2014. 7.1

[66] L. Prashanth and M. Ghavamzadeh. Actor-critic algorithms for risk-sensitive MDPs. In *Proceedings of the NIPS*, volume 26, 2013. 7.1

[67] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994. 2, 3.5.1.2

[68] K. H. Quah and C. Quek. Maximum reward reinforcement learning: A non-cumulative reward criterion. *Expert Systems with Applications*, 31(2):351–359, 2006. 2

[69] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Proceedings of the NIPS*, volume 20, 2007. 6.5.1

[70] A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade. Towards generalization and simplicity in continuous control. In *Proceedings of the NIPS*, volume 30, 2017. 6.5.2

[71] S. Ross and D. Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the ICML*, pages 1905–1912, 2012. 6.1, 6.1

[72] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the AISTATS*, pages 627–635, 2011. 5.3

[73] D. Russo and B. Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Proceedings of the NIPS*, volume 26, 2013. 4.1, 4.2, 4.2, 4.4

[74] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of the ICML*, pages 1889–1897, 2015. 1.1.2, 5.1

[75] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 5.1

[76] A. Sidford, M. Wang, X. Wu, L. F. Yang, and Y. Ye. Near-optimal time and sample complexities for for solving discounted markov decision process with a generative model. In *Proceedings of the NeurIPS*, volume 31, 2018. 2

[77] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017. 4.1

[78] M. Simchowitz and K. G. Jamieson. Non-asymptotic gap-dependent regret bounds for tabular MDPs. In *Proceedings of the NeurIPS*, volume 32, 2019. 5.2

[79] Z. Song, R. Wang, L. F. Yang, H. Zhang, and P. Zhong. Efficient symmetric norm regres-

sion via linear sketching. In *Proceedings of the NeurIPS*, volume 32, 2019. 1.2

[80] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. 4.4.3

[81] A. Tamar, D. Di Castro, and S. Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the ICML*, pages 1651–1658, 2012. 7.1

[82] A. Tamar, Y. Glassner, and S. Mannor. Optimizing the CVaR via sampling. In *Proceedings of the AAAI*, pages 2993–2999, 2015. 7.1

[83] C. Tessler, D. J. Mankowitz, and S. Mannor. Reward constrained policy optimization. In *Proceedings of the ICLR*, 2018. 8.1

[84] P. S. Thomas. *Safe reinforcement learning.* PhD thesis, University of Massachusetts, Amherst, 2014. 6.1

[85] P. S. Thomas, B. Castro da Silva, A. G. Barto, S. Giguere, Y. Brun, and E. Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019. 6.1

[86] J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015. 6.4

[87] S. S. Vempala, R. Wang, and D. P. Woodruff. The communication complexity of optimization. In *Proceedings of the SODA*, pages 1733–1752, 2020. 1.2

[88] L. Wang, W. Zhang, X. He, and H. Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the SIGKDD*, pages 2447–2456, 2018. 6.1

[89] R. Wang, S. S. Du, L. F. Yang, and S. M. Kakade. Is long horizon RL more difficult than short horizon RL? In *Proceedings of the NeurIPS*, volume 33, pages 9075–9085, 2020. 1

[90] R. Wang, S. S. Du, L. F. Yang, and R. Salakhutdinov. On reward-free reinforcement learning with linear function approximation. In *Proceedings of the NeurIPS*, volume 33, pages 17816–17826, 2020. 3

[91] R. Wang, D. P. Foster, and S. M. Kakade. What are the statistical limits of offline RL with linear function approximation? In *Proceedings of the ICLR*, 2021. 2

[92] R. Wang, R. Salakhutdinov, and L. F. Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. In *Proceedings of the NeurIPS*, pages 6123–6135, 2020. 2

[93] R. Wang and D. P. Woodruff. Tight bounds for ℓ_1 oblivious subspace embeddings. *ACM Transactions on Algorithms*, 18(1):1–32, 2022. 1.2

[94] R. Wang, Y. Wu, R. Salakhutdinov, and S. M. Kakade. Instabilities of offline RL with pre-trained neural representation. In *Proceedings of the ICML*, pages 10948–10960, 2021. 2

[95] R. Wang, P. Zhong, S. S. Du, R. Salakhutdinov, and L. F. Yang. Planning with general objective functions: Going beyond total rewards. In *Proceedings of the NeurIPS*, volume 33, pages 14486–14497, 2020. 3

[96] Y. Wang, R. Wang, S. S. Du, and A. Krishnamurthy. Optimism in reinforcement learning with generalized linear function approximation. In *Proceedings of the ICLR*, 2021. 1.2, 4.1, 4.2, 4.4.2, 4.4, 4.4.3

[97] Y. Wang, R. Wang, and S. M. Kakade. An exponential lower bound for linearly-realizable MDPs with constant suboptimality gap. In *Proceedings of the NeurIPS*, volume 34, 2021. 1.2

[98] Z. Wen and B. Van Roy. Efficient exploration and value function generalization in deterministic systems. In *Proceedings of the NIPS*, volume 26, 2013. 5.1, 5.3, 5.3, 8.1

[99] Y. Wu, G. Tucker, and O. Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019. 6.1, 6.1

[100] T. Xie and N. Jiang. Batch value-function approximation with only realizability. In *Proceedings of the ICML*, pages 11404–11413, 2020. 6.1

[101] Y. Xu, R. Wang, L. F. Yang, A. Singh, and A. W. Dubrawski. Preference-based reinforcement learning with finite-time guarantees. In *Proceedings of the NeurIPS*, volume 32, pages 18784–18794, 2020. 1.2

[102] L. F. Yang and M. Wang. Sample-optimal parametric Q-learning using linearly additive features. In *Proceedings of the ICML*, pages 6995–7004, 2019. 4.1, 4.2, 8.1, 8.2.2

[103] L. F. Yang and M. Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *Proceedings of the ICML*, pages 10746–10756, 2020. 4.1, 4.2

[104] A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the FOCS*, pages 222–227, 1977. 5.2.3, 5.4, 7.3, 8.4, 8.4.1

[105] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the NIPS*, volume 27, 2014. 6.1

[106] C. Yu, J. Liu, S. Nemati, and G. Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys*, 55(1):1–36, 2021. 1

[107] C. Yu, G. Ren, and J. Liu. Deep inverse reinforcement learning for sepsis treatment. In *Proceedings of the ICHI*, pages 1–3, 2019. 6.1

[108] A. Zanette, A. Lazaric, M. Kochenderfer, and E. Brunskill. Learning near optimal policies with low inherent bellman error. In *Proceedings of the ICML*, pages 10978–10989, 2020. 4.1, 4.4.2

[109] A. Zanette, A. Lazaric, M. J. Kochenderfer, and E. Brunskill. Limiting extrapolation in linear approximate value iteration. In *Proceedings of the NeurIPS*, volume 32, 2019. 4.1