Provably Efficient Reinforcement Learning Under Linear Model Structures: From Tabular to Feature Based Exploration

by

Aditya Modi

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor Satinder S. Baveja, Co-chair
Professor Ambuj Tewari, Co-chair
Assistant Professor Mahdi Cheraghchi
Assistant Professor Nan Jiang
Professor Clayton Scott

Aditya Modi
admodi@umich.edu
ORCID iD: 0000-0002-6959-0593

Dedicated to my parents for their complete and selfless devotion towards the success of their two children.

ACKNOWLEDGMENTS

I would like to begin by thanking my PhD advisors, Ambuj Tewari and Satinder Singh, without whom this thesis wouldn't have been possible, as they have played a big role in shaping my progress as a researcher and developing a broad scholarly outlook towards research, which is a key takeaway for any PhD student. Ambuj, in particular, has spent countless number of hours on research meetings, ranging from advising on minute technical details of research to lending a friendly ear to my woes of failures in research every now and then. Over the five years that I've worked with him, I've come to strongly admire the immense breadth of his knowledge in various topics, not only in machine learning/statistics/optimization, but their connections to other disciplines as well. I'll count myself as a successful student of his, on the day, when I can reflect even a fraction of the breadth and depth of his knowledge in my research works. I'm equally indebted to my co-advisor, Satinder Singh, who has went through the exacting process of teaching me the fundamentals of reinforcement learning and training me on different aspects of research ranging from writing/presentation skills to the importance of being clear and creative in one's research agenda. Owing to the numerous open ended discussions that we have had during my PhD, I've been able to expand my critical thinking and understanding from minute technicalities of theoretical research to broad, creative and potentially impactful ideas. Overall, I've been extremely fortunate to have the two as my mentors as they have provided me the freedom to pursue my own ideas, while carefully steering my progress as a researcher and simultaneously tolerating the many long unproductive months over the years.

In addition to my advisors, I owe a special thanks to Nan Jiang, who has played the role of a friendly mentor and collaborator over the past five years. He has been kind enough to teach me the about different topics in RL theory, has answered countless (and often absurd) queries and most importantly, has helped me in being part of the larger RL theory community by introducing me to different people in his broad circle during the various conferences, something that can be an intimidating experience for a newcomer. I'd also like to thank my two other committee members, Clayton Scott and Mahdi Cheraghchi, who graciously agreed to serve on my committee and spend time on reviewing my PhD dissertation and research.

No research endeavour can be fruitful without a strong collaboration. As such, I'd like to extend my gratitude to various collaborators: to Debadeepta Dey and folks in the MSR ASI group who

gave me a chance to work with them on an internship despite my inexperience, thereby, expanding the scope of my research as well as research network; to Alekh Agarwal and Akshay Krishnamurthy who I've had the pleasure to collaborate with to learn more about learning theory topics which I hope will continue in the future; to Jinglin Chen with whom I've shared and discussed so many ideas during our work on the MOFFLE paper; to Mohamad Kazem S Faradonbeh who went through the effort of teaching me the fundamentals of LQR and linear systems as part of the work in the final chapter of this thesis; to George Michailidis who anchored our discussions in so many meetings on the multi-task control problem; to Jenna Wiens and Shengpu Tang who introduced me to the potential applications of RL to healthcare domains and looped me in on their work. Lastly, I owe a lot to Barzan Mozafari who fortunately accepted me as a PhD student and advised me during the first year at Michigan.

I also want to thank my friends who have been the support staff for my PhD over the years; the IITK gang: Mayank, Vishal, Srajal for the umpteen random discussions, hangouts and trips; Aniket Deshmukh and Julian Katz-Samuels for helpful discussions on bandit and RL theory; the Willowtree junta: Ankush, Aman, Subarno, Arun, Vivek, Jana, Ankit and Mani; RL lab members and CASI lab members; and many others whose names I've surely missed. I'm also indebted to the CSE staff, especially, Ashley Andreae and Karen Liska, for simplifying the different departmental and university logistics.

I'm eternally indebted to my parents who have devoted their adult life to the success and happiness of their two sons and I dedicate this dissertation to them as a token of my gratitude towards them and to their belief in us which has always been stronger than our own. Similarly, I owe a lot of my achievements in life to my elder brother, Abhishek, who has faced so many firsts in his life and has paved the path for me to follow. I'm looking forward to the fun things we can do now that we are both in the same place! Lastly, I'd like to thank Ankita, who has shared the ups and downs which come with a PhD and has uplifted my spirits whenever they were down over the last five years. I always felt great even during a difficult day realizing that I can simply find solace in your company, despite the long distance between us.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

ABSTRACT

Reinforcement learning (RL) is a machine learning paradigm where an agent learns to interact with an environment in the presence of a reward signal as feedback. Recent breakthroughs have led to a renewed interest in building intelligent RL agents by combining the core RL algorithms with the expressivity of deep function approximators and advances in computation via simulation. Despite the recent advances, in most complex domains RL algorithms need a large amount of interaction data in order to learn a good policy. As a result, recent theoretical research has focussed on problems pertaining to the quantification and/or improvement of sample efficiency of RL under various interaction protocols. These efforts are directed towards understanding the statistical aspects of reinforcement learning, which can be a key factor in making progress towards real world applications, ranging from healthcare and robotics to control of large scale web systems.

The main theme of this thesis is the analysis of such information-theoretic aspects of RL in terms of the structural complexity of the environment by using tools from the learning theory literature. In this thesis, we consider a spectrum of scenarios: ranging from tabular to rich observation domains, single to multi-task settings and reward-specific to reward-free learning. Specifically, this thesis presents theoretical results in the following settings: the 1st part studies the sample efficiency of online learning in contextual Markov decision processes (MDPs) where the agent interacts with a sequence of tabular environments and has access to a high-dimensional representation that determines the dynamics of each MDP. The 2nd part studies the sample complexity of learning in a structured model class where the true model of an arbitrarily large MDP can be approximated using a feature-based linear combination of a known ensemble of models. The 3rd part investigates the problem of learning in a low-rank MDP where I design and analyze the first provably efficient model-free representation learning and exploration algorithm for reward-free learning. Lastly, in the 4th part, I provide results for online multi-task learning in linear quadratic regulators, under the assumption that the transition matrices for each system share a common basis.

A common thread running through all results in this thesis is the effect of a linear/low-rank structure in the model on the sample efficiency of learning. Overall, this thesis proposes novel provably efficient exploration and model selection algorithms under various linear model structures, and highlights the role of environment complexity in the statistical efficiency of reinforcement learning.

CHAPTER 1

Introduction

Reinforcement learning (RL) is a machine learning framework which can be used to model and build artificially intelligent agents (Sutton and Barto, 2018). RL studies the problem of building agents which can learn to make decisions in an environment in order to optimize a long-term reward. This learning paradigm provides a natural and general framework which characterises a range of applications scenarios, ranging from control of web systems (Li et al., 2010), learning adaptive health interventions (Nahum-Shani et al., 2018), online tutoring platforms (Koedinger et al., 2013), weather monitoring systems (Bellemare et al., 2020), to simulated or physical domains like gameplay (Mnih et al., 2015), robotics (Mordatch et al., 2015) and many more.

The field of reinforcement learning has witnessed a resurgence in research and development, after the recent empirical successes of RL algorithms on a range of domains like video/board games, and simulated robotics environments. The resurgence can also be attributed to the advances in modern deep learning architectures that provide highly expressive function approximation techniques, which can then be combined effectively with core RL algorithms. However, owing to the generality of the RL framework, such general learning approaches are often poor in terms of their statistical efficiency. As such, majority of the recent breakthroughs have been observed in simulated environments, and therefore, significant efforts are being made to improve RL algorithms and translate the empirical breakthroughs from simulated domains to domains where high-fidelity simulation is not a good design choice (eg. healthcare, web systems, education etc.). One aspect of this research endeavour is to understand the fundamental statistical properties of different components of the RL framework and employ such an understanding to design provably efficient algorithms. This thesis explores questions motivated by the same principles and presents a set of theoretical results which aid to our understanding of the statistical aspects of RL.

As a motivating example, consider any application domain where the goal is to design an intelligent agent which learns optimal decision making policies while interacting with a heterogeneous set of entities. Such scenarios arise in various domains like healthcare (heterogeneity among patients), online advertising (users on the platform) and so on. Typically, for many such interactive domains, neither a high-fidelity simulation model, nor an exhaustive interaction dataset, is available for the

empirical advances to take effect. As a consequence, the agent has to interact with such entities in the real world, take actions to collect interaction data and learn a good decision making policy for the given domain. However, this data collection step is expensive in these high stake scenarios and highlights the importance of designing sample efficient reinforcement learning algorithms.

The problem of studying the statistical aspects of reinforcement learning algorithms is a not new and has been explored in previous literature as well. The nature of results ranges from early asymptotic convergence results for core RL algorithms like TD-learning, Q-learning (Jaakkola et al., 1994; Singh et al., 2000) to finite time statistical guarantees for online learning in RL environments (Kearns and Singh, 2002; Jaksch et al., 2010; Dann and Brunskill, 2015). The latter problem deals with the problem of efficient exploration-exploitation during learning in a given environment, and is the main focus of this thesis. In an online RL problem, the agent has to adaptively collect interaction data to quickly learn a policy by balancing exploration of new regions in the environment and exploitation of already collected experience. This trade-off between exploration and exploitation is crucial for RL and is a well-studied phenomenon. Until recently, the majority of results for efficient exploration were given for simple environments which consist of finite state and action spaces.

On the other hand, much of the empirical success in RL has been achieved via expressive function approximators which learn good representations of the environment. Contrary to this, supervised learning has also witnessed huge breakthroughs in empirical results but, at the same time, has a much advanced understanding of generalization and robustness issues via statistical and online learning theories. A major component in supervised learning theory is the use of structural complexity measures for quantifying the statistical rates of learning under both supervised and online learning settings (Bousquet et al., 2003; Mohri et al., 2018). In this thesis, we will use these tools from the learning theory literature and study how an underlying structure in the model of the environment can be used to quantify the sample complexity of reinforcement learning under various learning scenarios.

Specifically, we will study the effect of various linear/low-rank structures in the model of the environment on the sample complexity of reinforcement learning under a range of problem settings: tabular to rich-observation domains, single-task to multi-task problems, and reward-specific to reward-free. In our results pertaining to all of these aspects in the reinforcement learning setting, we utilize various tools from the learning theory literature, and, also highlight the challenges which are unique to the the long-horizon decision making problem in RL. For instance, in Chapter 4 and Chapter 5, we will show that feature selection, which is a well understood problem in supervised learning, is a fundamentally hard problem in RL, and, therefore we need to exploit the problem structure appropriately to design provably efficient exploration algorithms. Overall, in this thesis, we investigate the problem of sample efficient learning in environments under various linear model structures and interaction protocols. Our results across all formulations underscore the importance of

environment structure in the statistical efficiency of RL and contribute towards the goal of designing provably efficient RL algorithms.

1.1 Thesis Statement

By utilizing linear/low-rank structures in the underlying model of the environment, we can design provably efficient reinforcement learning algorithms under a range of evaluation criteria for various problem settings: tabular to rich-observation domains, single or multi-task problems, reward-specific to reward-free learning and representation learning.

1.2 Contributions

In this section, we give a brief overview of the contributions of the thesis. In Chapter 2, we discuss the preliminary background of the topics considered in this thesis: MDP basics, planning and learning in MDPs, different efficiency criteria in online learning in MDPs and a brief background on linear quadratic regulators. In Chapters 3, 4, 5 and 6, we discuss the new research contributions in this thesis. Finally, in Chapter 7, we conclude with a summary and discussion on future work.

Online Learning in Contextual Markov Decision Processes (Chapter 3) In this chapter, we consider an RL setting in which the agent interacts with a sequence of episodic MDPs. Our setting is motivated by applications in healthcare, e-commerce and recommender systems where the agent has to adaptively interact with a *population* of users instead of a single environment. We capture this heterogeneity under the framework of contextual Markov decision processes (CMDPs) (Hallak et al., 2015), where, at the start of each episode, the agent has access to some side-information or context that determines the dynamics of the MDP for that episode. In the first part of this chapter, we derive the first PAC mistake bound guarantees for smoothly varying CMDPs and linearly parameterized CMDPs. These results demonstrate the hardness of learning in CMDPs under the weaker assumption of smooth variation, thereby, motivating the linear structural assumption. In the second part of this chapter, we extend the linear CMDP model using generalized linear models and propose optimistic and randomized regret minimization algorithms. For all settings, we provide a minimax lower bounds which further reflect the tradeoffs between the general smooth CMDP setting and the stronger linear structure. This chapter is based on joint work with Nan Jiang, Ambuj Tewari and Satinder Singh (Modi et al. (2018); Modi and Tewari (2020) published in ALT 2018 and UAI 2020 respectively).

Best Policy Identification in Linear Mixture MDPs (Chapter 4) In this chapter, we propose a new structural assumption, wherein, the true model of the environment can be approximated using a linear combination of a known ensemble of models, where the coefficients are determined by the given state-action features and some unknown parameters. The structural assumption is motivated by a setting where the agent has access to an ensemble of pre-trained and possibly inaccurate simulators (models). We propose an oracle-efficient algorithm which provably learns a near-optimal policy with a sample complexity polynomial in the number of unknown parameters, and incurs no dependence on the size of the state (or action) space. Given that the algorithm requires access to an expressive state-action feature, we also consider the more challenging problem of model selection, where the state features are unknown and can be chosen from a large candidate set. We provide exponential lower bounds that illustrate the fundamental hardness of this problem, and develop a provably efficient algorithm under additional natural assumptions. This chapter is based on joint work with Nan Jiang, Ambuj Tewari and Satinder Singh (Modi et al. (2020) published in AIStats 2020).

Model-Free Feature Learning and Exploration in Low-Rank MDPs (Chapter 5) In this chapter, we study the problem of learning in a low-rank MDP, a model which has emerged as an important setting for studying representation learning and exploration in reinforcement learning. With a known representation, several model-free exploration strategies exist, starting from the initial results in Yang and Wang (2019) and Jin et al. (2020). In contrast, all algorithms for the unknown representation setting are model-based, thereby requiring the ability to model the full dynamics. In this chapter, we present the first model-free representation learning algorithms for low-rank MDPs. The key algorithmic contribution is a new minimax representation learning objective, for which we provide variants with differing tradeoffs in their statistical and computational properties. We interleave this representation learning step with an exploration strategy to cover the state space in a reward-free manner. The resulting algorithms are provably sample efficient and can accommodate general function approximation to scale to complex environments. This chapter is based on joint work with Jinglin Chen[1], Akshay Krishnamurthy, Nan Jiang and Alekh Agarwal (Modi et al., 2021).

Provably Efficient Multi-Task Learning for Linear Quadratic Regulators (Chapter 6) In previous chapters, we have studied the problem of learning in Markov decision processes under varying structural assumptions. In reinforcement learning, MDPs are generally considered as the core framework for modeling the interaction between an agent and the environment. In this chapter, we consider a continuous control setting which has been studied extensively in the field of optimal control theory. In recent literature, a renewed interest has been observed in developing

[1]The co-author contributed equally to the work.

a non-asymptotic theory for linear-quadratic control, one of the fundamental problems in optimal control. Motivated by various application domains, we study a multi-task setting where the goal is to jointly learn (also known as multi-task learning) and control closely related linear time-invariant dynamical systems (LTIDS) under quadratic costs (LQR). To our knowledge, such a multi-task formulation hasn't been explored in the literature. To that end, we develop a joint estimator for learning the LTIDS' transition matrices, under the assumption that they share common structure in the form of common basis vectors. Further, we establish finite-time error bounds for this multi-task problem that depend on the underlying LTIDS' sample size, dimension, number of tasks and spectral properties of the transition matrices. The results are obtained under mild regularity assumptions and showcase the gains from pooling information across LTIDS, in comparison to using the data from each system separately. We also study the impact of misspecifying the joint structure of the transition matrices and show that the established results are robust in the presence of moderate degrees of misspecification. For the control setting, we show that the joint estimator can then be used in a certainty equivalence based no-regret learner and further analyze the multi-task finite time regret for the algorithm. This chapter is based on joint work with Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari and George Michailidis.

CHAPTER 2

Background

2.1 Markov Decision Processes

In reinforcement learning (RL), an *agent* (also referred to as *learner*) interacts repeatedly with an *environment* in the presence of a feedback loop enabling it make decisions optimally to achieve any given set of goals. This is largely formalized under the framework of a Markov Decision Process (MDP) (Puterman, 2014), which can be specified using the following components:

- State space: State of the world as observed by the agent denoted by $\mathcal{S}$.
- Action space: Set of actions available to the agent throughout the interaction, denoted by $\mathcal{A}$.
- Transition dynamics: A Markov transition function dictating the dynamics of the environment: $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$. Here, $\Delta(\mathcal{S})$ denotes the space of probability measures over the state space $\mathcal{S}$. Transition function can also depend on the episode timestep, in which case, we denote it by $P_h(\cdot|s_h, a_h)$ and $P \equiv \{P_0, P_1, \ldots, P_{H-1}\}$.
- Reward function: The preference function given to the agent, $R : \mathcal{S} \times \mathcal{A} \to \Delta([0, 1])$ where $r(s, a)$ denotes the scalar reward received by the agent on taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$, sampled from distribution $R(s, a)$. We will overload this notation to use $R(\cdot)$ for the mean reward as well. This thesis only covers bounded rewards in MDPs. We use a similar notation $R_h(\cdot)$ for time-dependent rewards.
- Horizon: The agent interacts with the environment for a fixed number of steps before the world resets, denoted by H.

Below, we collect the common notation and setup used in this thesis and describe the basic tools and concepts for learning and planning in MDPs.

MDP notation We will denote an MDP M by the tuple $(\mathcal{S}, \mathcal{A}, P, R, H, \mu)$ and follow the following basic interaction protocol for learning in MDPs: (1) the agent observes an initial state $s_0 \sim \mu$ where μ is a fixed initial state distribution, (2) for any episode timestep $h \in [H]$, the agent takes an action $a_h \in \mathcal{A}$ upon observing the state $s_h \in \mathcal{S}$ and (3) observes the next state $s_{h+1} \sim P(\cdot|s_h, a_h)$ and reward $r_h(s_h, a_h) \sim R(s_h, a_h)$. The episode terminates at the H^{th} timestep. The number of states and actions, if finite, are denoted by S and A respectively. Wherever applicable, we will use notation t to denote the episode index and total episodes by T. The t^{th} trajectory will be denoted by $\tau_t \equiv (s_{t,0}, a_{t,0}, r_{t,0}, s_{t,1} \ldots, s_{k,H})$[1]. If the policy is varies across episodes, we use the notation: π_t to denote the overall policy for episode t and $\pi_{t,h}(\cdot)$ to denote the h-th step policy for episode k. Further, for any quantity x, we use $x_{h:h'}$ to denote the set $\{x_h, x_{h+1}, \ldots, x_{h'}\}$.

General notation We use $[N]$ to denote the set of intergers $\{0, 1, \ldots N-1\}$ and $[N]_+$ to denote set $\{1, 2, \ldots, N\}$. For a matrix $A \in \mathbb{R}^{d_1 \times d_2}$, A' denotes the transpose of A. For a square matrix A, we use $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ to denote the smallest and largest eigenvalues (in magnitude), and for symmetric matrices, $\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \lambda_d(A) = \lambda_{\min}(A)$. For singular values, we will use $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$. For any vector $v \in \mathbb{R}^d$, we will use $\|v\|_p$ to denote its ℓ_p norm. Similarly, for a vector $x \in \mathbb{R}^d$ and a matrix $A \in \mathbb{R}^{d \times d}$, we define $\|x\|_A^2 \coloneqq x^\top A x$. For matrices $W \in \mathbb{R}^{m \times n}$ and $X \in \mathbb{R}^{n \times n}$, we define $\|W\|_X^2 \coloneqq \sum_{i=1}^m \|W^{(i)}\|_X^2$ where $W^{(i)}$ is the i^{th} row of the matrix. We use $\|\!|A|\!\|_{\gamma \to \beta}$ to denote the operator norm defined as follows for $\beta, \gamma \in (0, \infty]$ and $A \in \mathbb{C}^{d_1 \times d_2}$: $\|\!|A|\!\|_{\gamma \to \beta} = \sup_{v \in \mathbb{C}^{d_2} \setminus \{\mathbf{0}\}} \frac{\|Av\|_\beta}{\|v\|_\gamma}$. When $\gamma = \beta$, we simply write the operator norm as $\|\!|A|\!\|_\beta$. For any two matrices $A, B \in \mathbb{R}^{d_1 \times d_2}$, we define the inner product $\langle A, B \rangle = \operatorname{tr} A'B$. Then, define the Frobenius norm of matrices as $\|A\|_F = \sqrt{\langle A, A \rangle}$. The sigma-field generated by random variables $X_1, X_2, \ldots, X_n$ is denoted by $\sigma(X_1, X_2, \ldots X_n)$. We denote the i-th component of the vector $x \in \mathbb{R}^d$ by $x[i]$. The notation $\Delta_{d-1} \coloneqq \left\{x \in \mathbb{R}^d : \|x\|_1 = 1, x \geq 0\right\}$ will be used to refer to the simplex in dimension d. Finally, we use $\vee$ ($\wedge$) to denote the maximum (minimum).

2.1.1 Value function and policy

The agent's decision making strategy is denoted by a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$. A policy can be deterministic or randomized and can be stationary or time-dependent. In the latter case, we denote it by π_h and the action a_h is chosen according to the policy π_h ($a_h \sim \pi_h$). When the environment is too complex, we often consider a setting where the agent can act according to a class of policies which we denote by Π (or Π_h). For instance, the class of all deterministic policies in a tabular MDP[2] is of size A^S. In reinforcement learning, the agent's goal is to maximize the long term utility for a

[1] However, as a mnemonic, if we sample trajectories for empirical estimates of quantities, we will use notation $\left(s_h^{(i)}, a_h^{(i)}\right)$ for pairs in i-th trajectory.

[2] When the state-action space $\mathcal{S} \times \mathcal{A}$ is finite, we call it a tabular/finite MDP.

policy π in an episode. We denote the *value* obtained by an agent using a policy π in MDP M as follows:

$$v_M^\pi = \mathbb{E}\left[\sum_{h=0}^{H-1} r_h(s_h, a_h) \middle| s_0 \sim \mu, s_{h+1} \sim P_h(\cdot|s_h, a_h), a_h \sim \pi_h\right]. \tag{2.1}$$

In addition to this scalar metric for evaluation, the notion of a policy's value $V_{M,h}^\pi : \mathcal{S} \to [0, H]$ when using policy π from step h at a state s is given as follows:

$$V_{M,h}^\pi(s) = \mathbb{E}\left[\sum_{h'=h}^{H-1} r_{h'}(s_{h'}, a_{h'}) \middle| s_h = s, s_{h'+1} \sim P_h(\cdot|s_{h'}, a_{h'}), a_{h'} \sim \pi_{h'}\right]. \tag{2.2}$$

Since, we only consider bounded rewards $r(s, a) \in [0, 1]$, the value function range is also bounded in $[0, H]$. Similarly, we also define an action value function $Q_{M,h}^\pi : \mathcal{S} \times \mathcal{A} \to [0, H]$ as follows:

$$Q_{M,h}^\pi(s, a) = \mathbb{E}\left[\sum_{h'=h}^{H-1} r_{h'}(s_{h'}, a_{h'}) \middle| s_h = s, a_h = a, s_{h'+1} \sim P_h(\cdot|s_{h'}, a_{h'}), a_{h'} \sim \pi_{h'}\right] \tag{2.3}$$

The policy which maximizes the value in an MDP M is denoted by π_M^* (or π^*) and the optimal value by v_M^* (shorthand for $v_M^{\pi^*}$). The value functions associated with the optimal policy π^* are written as $V_{M,h}^*(\cdot)$ and $Q_{M,h}^*(\cdot)$ (shorthand for $V_{M,h}^{\pi^*}$ and $Q_{M,h}^{\pi^*}$ respectively). In subsequent notation, if the MDP M is clear by context, we will eliminate the subscript for conciseness.

2.1.2 Bellman equations for MDPs

The value function of a given policy in an MDP can be expressed using a set of fixed point equations derived from the principles of dynamic programming (Bellman, 1952; Bertsekas and Tsitsiklis, 1996). For $V_h^\pi : \mathcal{S} \to [0, H]$ and $Q_h^\pi \to [0, H]$, we have:

$$\begin{aligned} Q_h^\pi(s, a) &= \mathbb{E}_{s' \sim P_h(\cdot|s,a)}\left[R_h(s, a) + V_{h+1}^\pi(s')\right] \\ V_h^\pi(s) &= \mathbb{E}_{a \sim \pi_h(s)}\left[Q_h^\pi(s, a)\right]. \end{aligned} \tag{2.4}$$

We will also use the shorthand $Q_h^\pi(s, \pi'(s))$ for the term $\mathbb{E}_{a \sim \pi'(s)}\left[Q_h^\pi(s, a)\right]$. The value function for the termination timestep H is always set to 0: $Q_H^\pi(s) = 0, \forall s \in \mathcal{S}, \pi$.

From (2.4), we can see that the value function of a policy is governed by a system of linear equations over the variables $Q^\pi \in [0, H]^{SAH}$.

Similar to the case of policy evaluation, dynamic programming also allows us to express the

optimal value function and policy as the solution of a set of fixed point equations:

$$\begin{aligned} Q_h^*(s,a) &= \mathbb{E}_{s' \sim P_h(\cdot|s,a)} \left[R_h(s,a) + V_{h+1}^*(s') \right] \\ V_h^*(s) &= \max_{a \in \mathcal{A}} Q_h^*(s,a). \end{aligned} \tag{2.5}$$

The optimal policy can be shown as the greedy policy with respect to the optimal Q^* value function:

$$\pi_h^*(s) = \operatorname*{argmax}_{a \in \mathcal{A}} Q_h^*(s,a). \tag{2.6}$$

Since, we will use the greedy policy with respect to a function frequently, we use the notation $\pi_f(s) = \operatorname{argmax}_{a \in \mathcal{A}} f(s,a)$ to denote the greedy policy with respect to an action-value function f.

The one-step update to the value function Q_{h+1} in the equations plays an important role in a lot of methods, and for ease of exposition, we will denote the one-step backup of any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ by the *Bellman optimality operator* as follows:

$$(\mathcal{T}_h f)(s,a) = R_h(s,a) + \mathbb{E}_{s' \sim P_h(\cdot|s,a)} \left[\max_{a' \in \mathcal{A}} f(s',a') \right]. \tag{2.7}$$

The Bellman optimality equation in (2.5) can then be succinctly written as $Q_h^* = \mathcal{T}_h Q_{h+1}^*$ for all $h \in [H]$. Similarly, using the policy-specific Bellman equation in (2.4), we can also define a *Bellman operator* $\mathcal{T}^\pi$:

$$(\mathcal{T}_h^\pi f)(s,a) = R_h(s,a) + \mathbb{E}_{\substack{s' \sim P_h(\cdot|s,a) \\ a' \sim \pi(s')}} \left[f(s',a') \right]. \tag{2.8}$$

In both cases, whenever clear by context, we remove the subscript h for succinctness.

2.1.3 Planning in an MDP

In this section, we review classical methods used to evaluate policies or computing optimal policies and value functions for an MDP. For a more comprehensive and quick overview, the reader can refer to the first part of the RL book by Sutton and Barto (2018). The planning problem in RL requires the agent to compute the optimal policy π_M^* for a given MDP M. Beginning with the problem of policy evaluation, we will see that all of the following algorithms follows naturally from the Bellman operators dictating the structure of the corresponding functions.

2.1.3.1 Policy evaluation

Policy evaluation is the problem of computing the value function V^π or the value of a policy v^π for an MDP M. We describe two common methods used for policy evaluation below:

Monte-Carlo policy evaluation Monte-Carlo estimation is a direct method of estimating the value function of a policy in a given MDP. We refer to it as the direct method as it doesn't require any knowledge of the model of the MDP and uses randomly sampled trajectories for the estimation.

Since the total reward an agent observes is a random variable dependent on the trajectory seen in an episode, we can simply run the policy for different episodes for sufficient number of times and average the total reward: For all states $s \in \mathcal{S}, h \in [H]$, we repeat the following procedure:

1. Collect n trajectories of length $H - h$ starting from state s using policy π.
2. Estimate the value function estimate as $\widehat{V}_h^\pi = \sum_{i=1}^{n} \sum_{h'=h}^{H-1} r_h^{(i)}$.

The accuracy of the estimated value function $\max_{s,h} \left| V_h^\pi(s) - \widehat{V}_h^\pi(s) \right|$ can be established by using standard concentration inequalities like Hoeffding's bound (Theorem A.1) with the bounded random variable in range $[0, H]$. In case we are only interested in v^π, we can simply simulate a number of episodes with the initial state sampled from the initial state distribution μ. The key benefit of using this direct estimation method is that we don't need a sample set whose size scales polynomially with the size of the MDP. We will utilize this fact at various points in this thesis.

Next, we review a class of methods which relies on the Bellman equations for estimating value functions; an approach called *bootstrapping* in RL (estimating unknowns using estimates of unknown quantities).

Iterative policy evaluation When the agent has full knowledge or an estimate of the model of the MDP, we can use a dynamic programming algorithm to compute the value function in the backward order from step H to 0 using operator $\mathcal{T}_h^\pi$. For $h = H-1, H-2 \ldots, 0$, we compute the value function as follows:

$$Q_h^\pi = \mathcal{T}_h^\pi Q_{h+1}^\pi. \tag{2.9}$$

As the reader would note, performing an exact Bellman update requires full knowledge of the reward function and the transition dynamics. When the agent does not have access to a full model of the MDP, a sample based method can be used to compute an approximate value function $\widehat{Q}_h^\pi$ such that: $\widehat{Q}_h^\pi = \widehat{\mathcal{T}}_h^\pi \widehat{Q}_{h+1}$. When we use such an estimate to compute/update value functions with an approximate Bellman backup $\widehat{\mathcal{T}}_h^\pi$ (*bootstrapping*), the accuracy of the computed estimate is also affected by the error in performing the Bellman backup. The following result from Munos (2007) shows how the worst case value error can be bounded in terms of the Bellman backup error:

Lemma 2.1 (Error propagation in policy evaluation (Munos, 2007))**.** *Suppose that for each level* $h \in [H]$*, we have* $\left\| \widehat{\mathcal{T}}_h^\pi \widehat{Q}_{h+1}^\pi - \mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi \right\|_\infty = \sup_{s,a} \left| \left(\widehat{\mathcal{T}}_h^\pi \widehat{Q}_{h+1}^\pi \right)(s,a) - \left(\mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi \right)(s,a) \right| \leq \epsilon$,

then for all $h \in [H]$ we have:

$$\left\|\widehat{Q}_h^\pi - Q_h^\pi\right\|_\infty \leq (H-h)\epsilon.$$

Proof. By definition, we know that $\left\|\widehat{Q}_H^\pi - Q_H^\pi\right\|_\infty = 0$. Now, for any $h \in [H]$, we have:

$$\begin{aligned}
\left\|\widehat{Q}_h^\pi - Q_h^\pi\right\|_\infty &\leq \left\|\widehat{Q}_h^\pi - \mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi\right\|_\infty + \left\|\mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi - \mathcal{T}_h^\pi Q_h\right\|_\infty \\
&\leq \left\|\widehat{\mathcal{T}}_h^\pi \widehat{Q}_{h+1}^\pi - \mathcal{T}_h^\pi \widehat{Q}_{h+1}^\pi\right\|_\infty + \left\|\widehat{Q}_{h+1}^\pi - Q_{h+1}^\pi\right\|_\infty \\
&\leq \epsilon + \left\|\widehat{Q}_{h+1}^\pi - Q_{h+1}^\pi\right\|_\infty.
\end{aligned}$$

Therefore, by unrolling the recursion, we can establish the result for all $h \in [H]$. □

In a range of learning methods, where either an approximate model $\widehat{M} \equiv (\widehat{R}, \widehat{P})$ is used or a direct sample based backup is performed, we will employ this result to bound the approximation error for any given policy.

2.1.3.2 Value iteration

We now move our attention to planning where the agent computes the optimal value function V_h^* and the optimal policy π_h^*. In value iteration, we again use dynamic programming from step $H-1$ to step 0, but now apply the Bellman optimality operator in (2.7). Therefore, with $Q_H^*(\cdot) = 0$, for $h = H-1, H-2 \ldots, 0$, we compute the optimal value function as follows:

$$Q_h^* = \mathcal{T}_h Q_{h+1}^*. \tag{2.10}$$

Similar to the policy evaluation problem, we can show an error bound when an approximate Bellman optimality backup $\widehat{\mathcal{T}}_h(\cdot)$ is used instead of exact updates. By estimating the optimal value function as $\widehat{Q}^*$, we can use the greedy policy with respect to this estimate as our behavior policy. In this case, in addition to the approximation error in the value function, we also quantify the value loss when the greedy policy $\hat{\pi}^* = \pi_{\widehat{Q}^*}$ is used instead of π^*:

Lemma 2.2 (Error propagation and value loss in Q^* (Singh and Yee, 1994; Munos, 2007))**.** *In approximate value iteration, if for all $h \in [H]$, we have $\left\|\widehat{\mathcal{T}}_h \widehat{Q}_{h+1}^* - \mathcal{T}_h \widehat{Q}_{h+1}^*\right\|_\infty \leq \infty$, then the following holds:*

$$\begin{aligned}
\left\|\widehat{Q}_h^* - Q_h^*\right\|_\infty &\leq (H-h)\epsilon \\
\left\|Q_h^{\hat{\pi}^*} - Q_h^*\right\|_\infty &\leq (H-h)^2\epsilon.
\end{aligned}$$

Proof. The first part follows similarly as the previous result in Lemma 2.1. For the second part, note that $\mathcal{T}_h^{\hat{\pi}^*}\widehat{Q}_{h+1}^* = \mathcal{T}_h\widehat{Q}_{h+1}^*$ as it is the greed operator. Thus, we get:

$$\begin{aligned}
\left\|Q_h^{\hat{\pi}^*} - Q_h^*\right\|_\infty &\leq \left\|\mathcal{T}_h^{\hat{\pi}^*}\widehat{Q}_{h+1}^* - \mathcal{T}_h Q_{h+1}^*\right\|_\infty + \left\|\mathcal{T}_h^{\hat{\pi}^*}Q_{h+1}^{\hat{\pi}^*} - \mathcal{T}_h^{\hat{\pi}^*}\widehat{Q}_{h+1}^*\right\|_\infty \\
&\leq \left\|\widehat{Q}_{h+1}^* - Q_{h+1}^*\right\|_\infty + \left\|Q_{h+1}^{\hat{\pi}^*} - Q_{h+1}^*\right\|_\infty + \left\|\widehat{Q}_{h+1}^* - Q_{h+1}^*\right\|_\infty \\
&\leq \left\|Q_{h+1}^{\hat{\pi}^*} - Q_{h+1}^*\right\|_\infty + 2(H-h-1)\epsilon \\
&\leq (H-h)^2\epsilon.
\end{aligned}$$

□

In the reinforcement learning setting, we will repeatedly use an approximate model or simply a regression based approximate Bellman backup to estimate Q^*. As such, Lemma 2.2 will be used often in the subsequent text.

2.1.3.3 Policy Iteration

In policy iteration, we iterate in the policy space using a series of strict policy improvements. Starting from an arbitrary policy π^0 and iterate to improved policies as follows: for any policy iterate π^t, (1) compute the value function V_h^t using any policy evaluation method, and (2) update the policy to $\pi^{t+1} \coloneqq \pi_{V^t}$. We are essentially performing greedy improvements in the policy space using intermediate policy evaluation steps.

We know that if a greedy policy is chosen using the exact value function, then the policy improves monotonically, i.e., $V_h^{t+1}(s) \geq V_h^t(s)$ for all state action pairs and the improvement is strict for at least on (s, a) pair. Using this guarantee, we know that the policy iteration procedure terminates in A^S steps. For further details, refer to Bertsekas and Tsitsiklis (1996) and Scherrer (2016).

2.1.4 State-action representations in an MDP

The representation of the state-action space as used by the agent for learning in a given environment has a strong impact on its sample efficiency. Depending on how the state-action space is processed internally by the agent, the complexity (both computational and statistical) can scale according to the actual size of the MDP, say $O(\text{poly}(S, A, H))$, or the effective size of the MDP as considered by the agent (which can be much smaller than the environment size). Below, we describe the major representational settings from the RL literature we consider in this paper.

2.1.4.1 Tabular representation

When the state action space of a given MDP M is discrete and finite in size, the agent can use a represent different quantities relevant to an MDP on a per state-action pair basis. As maintaining such a set of values is akin to storing a table of inputs and outputs in a table of size $\text{poly}(S, A)$, these methods are referred to as *tabular methods*. In Section 2.2, we will see that the minimax rates for learning in tabular environments scale polynomially with the size of the state-action space. For the tabular case, we can characterize different learning algorithms in the following two broad categories:

Model-based algorithms We call a method, *model-based*, when the learning agent builds an estimate $\widehat{P}_h(\cdot|s,a)$ of the transition model of the MDP $P_h((\cdot|s,a))$ for all $s \in \mathcal{S}, a \in \mathcal{A}, h \in [H]$ (and possibly the reward function for each (s,a) pair). In some cases, instead of maintaining a direct estimate of the model, the agent can store proxy quantities which can be effectively used for a model-based estimate later. In tabular MDPs, this can be simply expressed as a space complexity condition that the agent stores $O(SAH)$-many quantities. A popular example for this approach would be building a *certainty-equivalent* MDP where the agent stores the count of each transition tuple seen so far: $n_h(s,a,s')$ for each tuple (s,a,s',h). Using this, we can build a CE estimate of the MDP as:

$$\widehat{P}_h(s'|s,a) = \frac{n_h(s,a,s')}{\sum_{\bar{s}\in\mathcal{S}} n_h(s,a,\bar{s})}.$$

In later sections and chapters, we will see that in addition to such a model, the agent can also maintain a measure of uncertainty for each (s,a,s') tuple to use that during learning. In this thesis, we use such tabular model-based methods in Chapter 3.

Model-free algorithms Another approach for reinforcement learning in the tabular framework is to directly work with a value function and/or a policy. In that case, the agent maintains a function of form $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ which can be stored in a table of size $O(SA)$. Prominent examples of learning algorithms which use such a strategy are: Q-learning (Watkins and Dayan, 1992), sarsa (Sutton, 1996) and policy gradient methods (Sutton et al., 2000).

Note We include the following brief points on model-based vs. model-free RL below:

(i) Until recently, the folklore assumption was that model-free methods are statistically inefficient when compared to model-based methods, even under tabular settings. However, the recent paper on optimistic Q-learning by Jin et al. (2018) shows that a version of Q-learning which

uses uncertainty-based bonuses achieves near-optimal performance guarantees which had been shown for model-based methods only.

(ii) Noting this recent result, Sun et al. (2019) state that the space complexity interpretation of model-based vs model-free algorithms does not provide any insight in the case where function approximation is used in learning instead of tabular estimates. Consequently, they define model-free algorithms as a class of algorithms which do not access the states $s \in \mathcal{S}$ directly but only through a functional interface of state-action function class $\mathcal{G}$. For such cases, they provide an exponential separation result for factored MDPs (Kearns and Koller, 1999) between model-free and model-based algorithms. In this thesis, in *model-based* methods, we will always build and use a transition model whereas for model-free algorithms, we will directly operate in the value function space (Chapter 5).

(iii) In addition to purely model-based and model-free methods, hybrid methods have also been studied in the literature. The most popular and representative example of such methods is Dyna (Sutton, 1991).

2.1.4.2 State abstractions

When the state space for a given MDP is large, then a polynomial dependence on S is not plausible. A direct way of handling large state spaces is to map the *raw* state space to a smaller finite *abstract* state space. This map ϕ of clustering states into smaller finite groups is referred to as state abstraction and has been studied extensively in the literature. Thus, for an agent using a state abstraction ϕ, any function $f : \phi \circ \mathcal{S} \to \mathbb{R}$ is defined under the state abstraction and will have the same value for states which map to the same abstract state, i.e., if $\phi(s_1) = \phi_{s_2}$, then $f(s_1) = f(s_2)$, where we overload the notation for f to lift the mapping from $\mathcal{S}$ to $\phi \circ \mathcal{S}$.

Clearly, the main benefit of using a state abstraction is the statistical advantage it brings in via controlling the size of the abstract MDP. Noting the behavior of tabular methods, the sample complexity of learning in the abstract MDP would now scale with the size of the abstraction $|\{\phi(s)\}_{s\in\mathcal{S}}|$. However, operating under the abstract MDP can lead to a loss of information and any model, value function or state-action mapping that can be learnt in the abstract MDP will deviate from the true function, thereby, affecting the agent's value. Therefore, improving the statistical efficiency by controlling *estimation errors* is in direct contrast with the *approximation error* suffered under the abstraction.

Ideally, a state abstraction is a map which group states such that the difference in a quantity of interest is sufficiently small. In Li et al. (2006), the author proposes a range of approximation notions for state abstractions which preserve the optimal policy/value or value of all policies or the

transition model and rewards of the MDP. The last notion, known as *model-irrelevance* abstraction is the strongest notion of state abstraction and it defined as follows:

Definition 2.1 (Model-irrelevant abstraction)**.** *For an MDP $M \coloneqq (\mathcal{S}, \mathcal{A}, P, R, H)$, a state abstraction ϕ is considered to be (ϵ_r, ϵ_p)-approximately model-irrelevant if, $\forall s_1, s_2 \in \mathcal{S}$ with $\phi(s_1) = \phi(s_2)$, $a \in \mathcal{A}$ and $x \in \phi(\mathcal{S})$, we have:*

$$|R(s_1, a) - R(s_2, a)| \leq \epsilon_r \quad \textit{and} \quad \left| \sum_{s' \in \phi^{-1}(x)} P(s'|s_1, a) - \sum_{s' \in \phi^{-1}(x)} P(s'|s_2, a) \right| \leq \epsilon_p.$$

This strong notion of model-irrelevance was first proposed by Givan et al. (2003) as *bisimulation* ($\epsilon_r = \epsilon_p = 0$) and an approximate notion was studied later in Ferns et al. (2004). In Chapter 3 and Chapter 4, we use such abstractions in the analysis or show a hardness result about abstraction selection from a given class. Further, in Chapter 5, we give another representation selection result which implies a sample efficiency result where the agent to select a sufficiently accurate abstraction from a given candidate abstraction class.

2.1.4.3 Finite dimensional features as a representation

Another method of handling large state in RL is to use an expressive feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ instead of the raw state action variables. For instance, a state abstraction can be equivalently considered as a $|\phi(\mathcal{S})|$-dimensional feature representation which represents each partition as a one-hot encoding vector.[3]

Similar to the abstraction setting, for these finite dimensional features, depending on the structural properties of the representation, we can learn in a sample efficient way and only pay a sample complexity which depends on the dimension d instead of the actual cardinality of the state-action space. However, the expressivity assumptions required for such a Euclidean feature representation is quite different from abstractions, as the agent still needs to learn in a non-tabular MDP. The most popular way to use a feature representation is to consider a class of value fuctions $\mathcal{F} : \phi \circ (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}^d$ or policy class $\Pi : \phi \circ (\mathcal{S}) \to \Delta(\mathcal{A})$ during learning.

The simplest function approximation class that is considered in the literature is *linear function approximation*. Given a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, we can define a class of linear functions as $\mathcal{F} = \{f(s, a; w) = \langle \phi(s, a), w \rangle : w \in \mathbb{R}^d, \|w\|_2 \leq B\}$ for some bounded set of parameters $w \in \mathbb{R}^d$. Using $\mathcal{F}$, we can estimate the value function of a given policy or use iterative methods to learn a sufficiently accurate approximation of the optimal value function. Reinforcement learning with linear function approximation has been studied extensively over the years (including but not limited

[3] A tabular MDP simply uses an SA-dimensional one-hot encoding as feature representation during learning.

to Bradtke and Barto (1996); Tsitsiklis and Van Roy (1997); Parr et al. (2008); Sutton et al. (2009)) and has garnered the attention of the RL theory community in the past few years.

For online learning in MDPs, depending on the quantity the agent is trying to estimate, we can consider a hierarchy of expressivity assumptions for the representation ϕ during learning: (1) ϕ can approximate the optimal value function Q^* (or V^* or both) with small worst case error $\min_w \|\langle \phi(s,a), w\rangle - Q^*(s,a)\|_\infty \leq \epsilon$, (2) ϕ can approximate the value function Q^π of any policy π with small worst case error $\min_w \max_\pi \|\langle \phi(s,a), w\rangle - Q^\pi(s,a)\|_\infty \leq \epsilon$ or stronger conditions which imply more expressivity. A series of recent works (Du et al., 2019b) has shown that for online learning in an MDP, just being able to approximate Q^π or Q^* is not enough for sample efficient RL as the agent has to incur a sample complexity which can be exponential in the dimension in the worst case scenario. On the other hand, for planning Weisz et al. (2021b) recently showed that query-efficient planning is possible with polynomial query complexity for $O(1)$-sized action space whereas an exponential lower bound can be shown when $A = O(\text{poly}(d, H))$ (Weisz et al., 2021a).

In contrast to these negative results, a few structural assumptions have been proposed for learning in MDPs wherein linear function approximation can be used to learn a near-optimal policy in a sample efficient manner. A recent prominent example is that of a low-rank MDP which makes a linear expressivity assumption in the model space instead of just the value function or policy:

Definition 2.2 (Yang and Wang (2019); Jin et al. (2020))**.** *A given MDP is called a linear/low-rank MDP with embedding dimension d if all the transition operators $P_h : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ admits a low-rank structure. A transition operator is said to admit a low rank representation if there exists a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $\mu : \mathcal{S} \to \mathbb{R}^d$ such that:*

$$\forall s, a \in \mathcal{S} \times \mathcal{A}, \quad P_h(s'|s,a) = \langle \phi(s,a), \mu(s')\rangle \,.$$

In Jin et al. (2020), the authors show that if the feature ϕ for a low-rank MDP is known to the agent, their proposed algorithm LSVI-UCB can learn a near-optimal policy in a sample efficient manner. The key result is that the agent only incurs a sample complexity which is polynomial in the embedding dimension d which can be much smaller than the size of the state space. In Chapter 5, we study a representation learning setting for low-rank MDPs where the agent is not given the true feature ϕ^* but a class of realizable representations Φ. Our work constitutes the first model-free result in this representation learning setting.

Similar to low-rank MDPs, in Chapter 4 we propose another structural assumption, which is now commonly studied under the umbrella term *linear mixture MDPs*. We propose and analyse a sample efficient algorithm for this class of MDPs. In addition to these linearity assumptions in the model space, a few other sufficient conditions are also known for sample efficient feature based RL. We discuss another sufficiency condition in Section 2.2.4. For more details, we refer the reader to

Du et al. (2019b).

2.1.5 Alternative MDP settings

Here, we discuss the aspects of the problem setup which are stylized/specific to this thesis and are often formulated in an alternate manner:

Episodic vs infinite horizon Our definition of value function corresponds to an episodic setting where the agent interacts with the environment in episodes of fixed lengths H. This can also be used to characterize environments with a bounded horizon instead of a fixed one. A common alternative choice is to consider a continuing setting where the agent interacts with the environment for an infinite number of timesteps without any reset to a state sampled from an initial state distribution. The value function for such scenarios can be defined as an average reward $V^{\pi}(s) \coloneqq \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \mathbb{E}_{\pi} \left[r_t(s_t, a_t) \right]$, or as a discounted value function $V^{\pi}(s) \coloneqq \sum_{t=0}^{T} \mathbb{E}_{\pi} \left[\gamma^t r_t(s_t, a_t) \right]$ where $\gamma \in (0, 1)$ is a discount factor. The Bellman optimality and evaluation operators described in the previous sections can be stated for the average reward criteria as well as the discounted reward criteria. For average reward, in order to obtain learning guarantees, additional conditions on the connectivity of the MDP, like *ergodicity* or *communicating* properties are required. Such alternative criteria will not be considered in this thesis.

Reward distribution In our formalization for all chapters, we have assumed that the instantaneous reward lies in the range $[0, 1]$. This is only used for the simplicity of the analysis and can be extended to unbounded rewards with sub-Gaussian or sub-Weibull distributions.

Single reward maximization vs reward free learning In the reinforcement learning problem that we have discussed so far, we only considered the definition of value functions for a single reward and outlined the planning and approximation aspects for a single reward. However, in many learning scenarios, the agent's goal is to learn a sufficiently expressive representation of the environment using exploration such that any reward function $R(\cdot)$ in a given class $\mathcal{R}$ can be optimized later on. This reflects a many-goals setting for the agent and is applicable in many practical scenarios. We refer to this setting as a *reward-free learning* setting and it will be the main focus of Chapter 5.

States vs observations In this thesis, we consider environments which can be modeled as a MDP. The main assumption here is that a perfect 'state' is observed by the agent such that the history can be disregarded in decision making. However, in many scenarios, the environment can mimic a controlled hidden Markov model where an underlying latent state space dictates the evolution of a

given trajectory. The agent, instead of observing the states directly, observes an observation $o \in \mathcal{O}$ which can depend on the underlying latent state. This setting has been studied in detail under the framework of POMDPs, but will not be considered in this thesis. However, we will use a stylized result about a latent state representation for a low-rank MDP in Chapter 5.

2.2 Interaction Protocols and Sample Efficiency Criteria

In this section, we discuss the reinforcement learning problem wherein the agent actively interacts with an environment in order to maximize the utility after or even during the learning period. In the planning problem discussed in the previous section, the agent had full knowledge of the MDP M or access to a sampling sub-routine for (P, R) which allows access to the underlying model for (s, a) pairs. Hence, the key challenge in the planning problem is computational where the agent does not need to learn the model $M \equiv (P, R)$. In this thesis, our focus will be on the learning problem where the agent has to interact with the environment to learn a near-optimal policy.

Since the agent doesn't know the model of the MDP M, in the different interaction protocols that we discuss below, the agent has to adaptively collect data by choosing actions according to some policy. This dissertation considers the statistical aspect of RL in which the desiderata is to behave optimally or compute the optimal policy in this interactive setup in a sample efficient manner. Central to the learning problem is the well known *explore-exploit* tradeoff: the agent has to collect informative data which allows it to estimate the optimal policy and/or value function (read *explore*), while simultaneously balancing it to *exploit* the collected data to behave near-optimally. We will see that different interaction protocols motivate different evaluation metrics, which in turn, lead to various explore-exploit paradigms.

Another key aspect of sample efficient RL is that of *generalization* and *temporal credit assignment*. The generalization aspect arises when the environment that the agent interacts with is rich enough that maintaining a tabular model and/or value function is no longer feasible. In such a scenario, we can utilize any available structure in the state space and the model of the MDP to establish a functional mapping for a policy π and its value function V^π. The agent is then required to *generalize* what it sees on a subset of states to the larger state space. In this thesis, we propose and/or study a range of linear structures which allow us to use such *function approximation* schemes in a statistically efficient manner.

Temporal credit assignment refers to the problem of attribution of the value obtained by the agent in any episode to any (or a sequence of) action(s) taken in an episode. The framework of value functions and dynamic programming allows provides us with the basic tools and techniques for this aspect. During learning, the central challenge will be to adequately deal with the statistical and approximation errors which arise when one uses bootstrapping or Monte-Carlo methods to learn

(approximate) value functions. This is one of the two key aspects, in addition to explore-exploit tradeoff, that separates reinforcement learning from other learning paradigms.

Below, we describe the different interaction protocols that we study in this thesis and the different performance measures associated with them. For each criteria, along with the performance metric and the problem setup, we also outline the general algorithmic approaches used to devise sample efficient algorithms. As some of the use cases for the formal frameworks studied in this thesis are based in personalized healthcare, we will take that as an example to provide an intuition behind each efficiency criteria.

In subsequent chapters, we will build upon these basic templates to propose and analyze provably sample efficient methods in varying problem settings.

2.2.1 Online learning in MDPs

The online interaction protocol is the most general interactive setting considered in the literature. Here, the agent directly interacts with a given MDP M in a sequence of episodes with each episode starting with a varying or a fixed initial state distribution. The key factor here is that the agent is evaluated throughout this interaction and has to adaptively balance exploration and exploitation over all episodes. Therefore, there is no distinction between a data collection phase and the evaluation phase as is done in supervised learning. This evaluation metric models the over-arching goal of RL where an agent situated in any environment can learn to optimally behave in the world without incurring a large average cost. The learning protocol is as follows: For every episode $t = 1, 2, \ldots$

1. Agent chooses a policy π_t using previous trajectories $\tau_{1:t-1}$.
2. Using policy π_t, obtain a trajectory $\tau_t = \{(s_{t,0}, a_{t,0}, r_{t,0}), (s_{t,1}, a_{t,1}, r_{t,1}), \ldots (s_{t,H})\}$.
3. Obtain total reward $\sum_{h=0}^{H-1} r_{t,h}$ for the t-th episode.

We now describe the two common measures for online performance evaluation along with a baseline algorithm below:

2.2.1.1 PAC-efficient online learning

The evaluation criteria for an agent under the online provably approximately correct (PAC) RL framework is to consider the number of *mistakes* made by the agent during the interaction. For each episode t, the agent is considered to have incurred a mistake if the value of the agent's policy in that episode $v_t \equiv v_{M_t}^{\pi_t}$ is more than ϵ sub-optimal compared to the true value $v_t^* \equiv v_{M_t}^*$, i.e., $v_t^* - v_t \geq \epsilon$.

Definition 2.3 (PAC Mistake Bound). *Given a sequence of MDPs M_t for $t = 1, 2, \ldots$, an agent is said to be statistically PAC efficient with mistake bound* $\text{Mistake}(T, \epsilon, \delta)$*, if for every $\epsilon > 0, 0 < \delta < 1$, the agent's policies $\pi_1, \ldots, \pi_T$ satisfies the following:*

$$\sum_{t=1}^{T} \mathbb{1}\left[v_t^* - v_t \geq \epsilon\right] < \text{Mistake}(T, \epsilon, \delta)$$

with probability at least $1 - \delta$.

The above measure has also been referred to as the sample complexity of reinforcement learning in previous works (Kakade, 2003; Strehl and Littman, 2005) and is inspired by the notion of sample complexity from the seminal paper on the algorithm E^3 by Kearns and Singh (2002).

PAC mistake bounds in RL have been studied for tabular MDPs in the RL literature. A popular algorithmic scheme for efficient learning is using the *optimism under uncertainty* principle. Under the framework, the agent maintains confidence intervals for its estimates $\widehat{P}_h(\cdot)$ and $\widehat{R}_h(\cdot)$ and uses them to adaptively explore the important but uncertain state-action pairs. Some algorithms like E^3 have a distinction between exploration and exploitation episodes while most algorithms like RMAX (Brafman and Tennenholtz, 2002), MBIE-EB (Strehl and Littman, 2005), UBEV (Dann et al., 2017) etc. redefine/perturb the reward optimistically to account for the planning error in value iteration. The table below highlight the PAC bounds for influential algorithms from previous literature.[4]

Algorithm	PAC-Bound	Time	Space
RMAX (Brafman and Tennenholtz, 2002)	$\widetilde{O}\left(\frac{S^2AH^5}{\epsilon^3}\right)$	$O(THS^2A)$	$O(S^2A)$
MBIE-EB (Strehl and Littman, 2005)	$\widetilde{O}\left(\frac{S^2AH^6}{\epsilon^3}\right)$		
ORLC (Dann et al., 2019)	$\widetilde{O}\left(\frac{SAH^2}{\epsilon^2}\right)$		
DELAYED Q-LEARNING (Strehl et al., 2006)	$\widetilde{O}\left(\frac{SAH^8}{\epsilon^4}\right)$	$O(TH)$	$O(SA)$

Table 2.1: PAC mistake bounds for popular algorithms. The use of $O(SA)$ space in DELAYED Q-LEARNING shows the model-free nature of the method. Dann et al. (2017) showed that ORLC allows near-optimal performance guarantees for all performance measures in this section. For more details, refer to Section 2.2.3.

In Chapter 3, we will propose sample complexity bounds in RL for a multi-task setting with RMAX as our base algorithm in the analysis. For a comparison between PAC and other efficiency criteria, we refer the reader to Section 2.2.3.

[4]Until recently, PAC-RL bounds have been studied in the infinite horizon setting where a mistake is defined on a per state-action pair basis. Specifically, it counts the number of (s_h, a_h) pairs encountered during the interaction such that the agent's policy at that time for the pair is sub-optimal: $V^{\pi_h}(s_h, a_h) \leq V^*(s_h, a_h) - \epsilon$.

2.2.1.2 Online regret minimization

The mistake bound criteria in the previous section counted the number of sub-optimal steps/episodes as encountered by the agent during its interaction with the environment. Another ubiquitous efficiency criteria from the online learning literature is the notion of *regret*. The total regret incurred by an agent over a sequence of T episodes is the sum of optimality gaps across all episodes. Formally, we can define regret as follows:

Definition 2.4 (Regret bound.)**.** *For $t = 1, 2, \ldots$, given a sequence of MDPs M_t, an agent satisfies a high-probability regret bound, $f(T, \delta)$, if for any $0 < \delta < 1$, the agent's policies $\pi_1, \ldots, \pi_T$ satisfy the following inequality:*

$$\text{Regret}(T) \coloneqq \sum_{t=1}^{T} v_t^* - v_t \leq f(T, \delta),$$

with probability at least $1 - \delta$.

Regret minimization has been studied in the field of online for a long time for a range of full-information feedback to partial-information feedback problems. For MDPs, the first regret like setting was studied by Burnetas and Katehakis (1997) where their proposed approach, based on *index policies*, was shown to incur an asymptotic regret proportional to $\log T$. However, a major advance in the regret analysis of MDPs is due to the seminal work in Jaksch et al. (2010) where the authors propose and analyze the UCRL2 algorithm and show a $O(SH^{3/2}\sqrt{AT})$ regret bound for the average reward criteria in the infinite horizon setting.[5] Below, we describe the major algorithmic approaches for regret minimization in reinforcement learning.

Optimism in the face of uncertainty (OFU) This approach is reminiscent of the UCB algorithm for multi-armed bandits where a confidence interval is used for tackling the explore-exploit tradeoff. The UCRL2 algorithm in Jaksch et al. (2010) is based on the OFU principle wherein, the agent builds a confidence set $\mathcal{P}_{sa}$ for the parameters $P(\cdot|s, a)$ for all state action pairs such that, with high probability, $P(\cdot|s, a) \in \mathcal{P}_{sa}$. Using this confidence set, the agent computes a policy $\hat{\pi}$ such that $\hat{\pi} \coloneqq \pi^*_{\widehat{M}}$ where $\widehat{M} = \text{argmax}_{M \in \mathcal{M}} v^*_M$. Here, $\mathcal{M}$ is the confidence set on the complete MDP built by using the confidence set over the next state distributions and reward functions. This procedure is usually referred to as *optimistic planning*.

Another approach to solve the optimistic planning problem is to add value function bonuses in a value iteration scheme such that with high probability, the final value function $\widehat{Q}_h(s, a) \geq Q^*_h(s, a)$ for all state-action pairs and levels. This technique has recently been used in recent minimax optimal

[5] In infinite horizon problems, a quantity known as the *diameter* of the MDP plays the effective role of the horizon.

algorithms for tabular MDPs like Azar et al. (2017) with construction of improved confidence sets on the value functions. Further, the OFU template has also been used in feature based exploration; for instance, Jin et al. (2020) proposed the LSVI-UCB algorithm for linear MDPs using an elliptic bonus based algorithm based on the OFU algorithms in contextual linear bandits.

Randomized exploration methods Another popular approach for efficient exploration is to use sampling based approaches as first studied in Thompson (1933). For MDPs, Osband et al. (2013) proposed the Posterior sampling in RL (PSRL) algorithm based on the Thompson sampling approach. This Bayesian approach considers the case where the agent uses a prior distribution over the MDP parameters during learning. Specifically, using the data collected in previous episodes, the agent maintains a posterior distribution over MDP parameters. At the beginning of each episode, the agent samples an MDP from this posterior and acts according to the optimal policy of the sampled MDP. The analysis in Osband et al. (2013) shows that the algorithm suffers a Bayesian regret of the same order as UCRL2. Here, the Bayesian regret measures the total sub-optimality across T episodes and averages it according to the prior distribution.

Another approach for randomization, is to use randomized value bonuses instead of sampling from a posterior. These randomized approaches are often considered to be better on an average instance as they do not attempt to account for the worst case behavior during learning like OFU algorithms. Recently, Russo (2019) showed that the randomized exploration algorithm RLSVI incurs a regret of order $\widetilde{O}(\sqrt{T})$ in the worst case. This was the first worst-case regret bound for randomized exploration methods in reinforcement learning. Below, we show regret bounds for popular regret minimizing algorithms in Table 2.2.

Algorithm	PAC-Bound	Template
UCRL2 (Jaksch et al., 2010)	$\widetilde{O}\left(H^{3/2}S\sqrt{AT}\right)$	OFU
ORLC (Dann et al., 2019)	$\widetilde{O}\left(H\sqrt{SAT}\right)$	OFU
UCB-VI (Azar et al., 2017)	$\widetilde{O}\left(H\sqrt{SAT}\right)$	OFU
EULER (Zanette and Brunskill, 2019)	$\widetilde{O}\left(\sqrt{\mathbb{Q}^*SAT}\right)$	OFU
UCB-Q-Bernstein (Jin et al., 2018)	$\widetilde{O}\left(\sqrt{H^3SAT}\right)$	OFU
PSRL (Osband et al., 2013)	$\widetilde{O}\left(H^{3/2}S\sqrt{AT}\right)$	Posterior
RLSVI (Russo, 2019)	$\widetilde{O}\left(H^3S^{3/2}\sqrt{AT}\right)$	Randomized bonus

Table 2.2: Regret guarantees for popular algorithms. In EULER, $\mathbf{Q}^*$ denotes a conditional variance term for the value function and can be much smaller than the horizon H in certain cases. The guarantee for PSRL is a Bayesian regret bound.

We use these two approaches in Chapter 3 in this thesis for proposing regret minimizing al-

gorithms. Specifically, in Section 3.4, we propose both optimistic and randomized exploration algorithms with worst-case regret guarantees for contextual MDPs under generalized linear mappings.

2.2.2 Best policy identification à la the train-then-test paradigm

Another popular evaluation criteria for studying the sample efficiency of reinforcement learning algorithm is inspired from the supervised learning literature. In the PAC framework for supervised learning, an algorithm is called PAC-efficient if the returned hypothesis for any sample training dataset is ϵ-optimal with probability at least $1-\delta$, for a sample size that is of order $O(\text{poly}(d, 1/\epsilon, 1/\delta))$ (d is a problem relevant structural parameter). For MDPs, Fiechter (1994) proposed the first supervised learning stype PAC analysis. In this thesis, we will define PAC criteria as follows:

Definition 2.5 (PAC bound for RL)**.** *An algorithm is called PAC-efficient, if for any given MDP M, it returns a policy π_T after $T = O(\text{poly}(S, A, H, 1/\epsilon, 1/\delta))$ episodes, such that, with probability at least $1-\delta$, we have: $v^{\pi_T} \geq v^* - \epsilon$.*

When the size of the state space is large and a feature $\phi : \mathcal{S} \times \mathcal{A} \to \in \mathbb{R}^d$ is given to the agent for learning, we desire a PAC bound which is polynomial in $(d, H, 1/\epsilon, 1/\delta)$. In Chapter 4, we will see that an expressive feature map coupled with an underlying linear structure, allows us to devise exploration algorithms which are PAC efficient under this criteria. Below, we describe the PAC guarantees for recent methods which have been proposed in the literature.

Algorithm	PAC-Bound	Template
UCFH (Dann and Brunskill, 2015)	$\widetilde{O}\left(\frac{SAH^2}{\epsilon^2}\right)$	Tabular
ORLC (Dann et al., 2019)	$\widetilde{O}\left(\frac{SAH^2}{\epsilon^2}\right)$	Tabular
OLIVE (Jiang et al., 2017)	$\widetilde{O}\left(\frac{M^2AH^3}{\epsilon^3}\right)$	General function approximation

Table 2.3: PAC bounds for popular algorithms. For OLIVE, the authors proposed a structural measure called *Bellman rank* which can be used to characterize the sample complexity for RL when using general function approximation. For tabular MDPs, $M = SA$.

2.2.3 Translating one efficiency criteria to another

It is often desirable to have algorithms which guarantee efficiency for multiple evaluation criteria instead of just one. In Dann et al. (2017), the author proposed the idea of uniform-PAC and studied

the implication of performance guarantee under one criteria to another. Below, we briefly discuss such conversions:

1. Mistake bounds vs. regret: An algorithm which has a regret guarantee, as described in the previous section, cannot be converted to an algorithm with a mistake bound. Since only the total regret is bounded under the regret criteria, the agent can encounter infinitely many episodes such that the sub-optimality Δ_t is higher than a given threshold ϵ.

   On the other hand, if an algorithm has a finite mistake bound of $O(\epsilon^{-1})$, then it satisfies a $O(T^{2/3})$ high-probability regret bound for a specific $T = \Theta(\epsilon^{-3})$. Further, for expected regret, a lower bound of $\Omega(T^{2/3})$ can be shown.

2. Regret vs. PAC bounds: If an algorithm satisfies a PAC guarantee of $O(\epsilon^{-\beta})$, it can be converted to a regret bound of $O(T^{\beta/(1+\beta)})$ for fixed T. The idea here is to run the PAC algorithm for $T_1 = \epsilon^{-\beta}$ episodes and then use the returned policy for the remaining $T - T_1$ steps.

   On the other hand, for an algorithm with a regret guarantee, we can not obtain a high-probability PAC-efficient algorithm. However, the following guarantee can be given: suppose the algorithm satisfies a regret guarantee of $O(T^{1-\alpha})$. Now, if we run this algorithm for T episodes and then select a policy at random from $\pi_1, \ldots \pi_T$, with constant probability, we can guarantee that the policy is $O(T^{-\alpha})$ optimal.

For more details, the reader can refer to the discussion in Dann et al. (2017) and Jin et al. (2018).

2.2.4 Learning from offline data in RL

In many application domains, it is infeasible to run an online algorithm directly in the environment (for instance, healthcare domains). In such settings, we can however use any historical interaction data which has already been collected to optimize and/or evaluate decision making policies. As such, in these cases, instead of requiring an RL algorithm which actively interacts with the environment to collect data and learn a good policy, we instead need sample-based evaluation and planning methods. Formally, in these settings, we assume that a dataset $\mathcal{D}_h := \left\{s^{(i)}, a^{(i)}, r^{(i)}, s'^{(i)}\right\}_{i=1}^{n}$ is given to the agent for all levels. The desiderata is to evaluate and/or learn an optimal policy to sufficient accuracy by using polynomially many samples. Note that, this setting is inherently different from the three criteria discussed previously. Here, the exploration problem is essentially disregarded, and therefore, does not share the same algorithmic structure as the exploration algorithms from the previous sections.

Policy evaluation using batch data Offline policy evaluation considers the problem of evaluating the value of a policy π_e when a dataset of trajectories collected from a behavior policy π_b is given. If the behavior policy π_b is known, we can use importance sampling methods, where, the importance ratios $\rho_h = \pi_e(a_h|s_h)/\pi_b(a_h|s_h)$ are used. The step wise importance sampling estimator can be then used as follows:

$$\hat{v}^e = \sum_{i=1}^{n} \sum_{h=0}^{H-1} \rho_{0:h}^{(i)} r_h^{(i)}$$

where $\rho_{0:h}^{(i)}$ denotes the product of importance ratios from $h = 0$ to h for the i-th trajectory. These IS estimators provide an unbiased estimate of the value of any evaluation policy. However, it is well known that these importance sampling estimates suffer from high variance which can scale exponentially with the horizon.

Another approach is to use regression based estimators, which use the Bellman policy evaluation operator in (2.9) in a recursive manner based on the sampled dataset: let $\widehat{V}_H = 0$ and for $h = H-1, \ldots, 0$, evaluate:

$$\widehat{Q}_h(s,a) = \textsc{Estimate}\left(\mathbb{E}_{P_h}\left[R_h(s,a) + V_{h+1}(s')\right]\right)$$

Here, ESTIMATE can be any procedure based on applying function approximation to fit a model (Jong and Stone, 2007), or approximate dynamic programming methods which directly fit a value function using a chosen value function class (Dann et al., 2014). These estimators do not suffer from an exponential variance issue, but do however, introduce statistical bias in the estimates. Recently a range of methods which combine such function approximation methods with importance sampling schemes have been studied (Jiang and Li, 2016; Thomas and Brunskill, 2016).

Sample-based planning In this case, the goal is to use the given dataset $\{\mathcal{D}_h\}_{h\in[H]}$ to learn a near optimal policy for the reward function. Here, we briefly describe fitted Q-iteration (FQI), which is a sample based approximate dynamic programming method to compute the optimal value function of a policy. The idea here is to evaluate the Bellman optimality operator recursively from $h = H$ to $h = H-1, \ldots, 0$. Specifically, let $f_h \in \mathcal{F}_h$ be the estimate of the value function at step h, where $\mathcal{F}_h$ is the function class being used for approximation. FQI solves for such a function iteratively by minimizing the following squared loss:

$$\mathcal{L}_{\mathcal{D}_h}(f, f_{h+1}) = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(s^{(i)}, a^{(i)}\right) - r_h\left(s^{(i)}, a^{(i)}\right) - \max_{a'} f_{h+1}\left(s'^{(i)}, a'\right)\right)^2$$

In FQI, the agent estimates the value function as $\hat{f}_h \coloneqq \operatorname{argmin}_{f \in \mathcal{F}_h} \mathcal{L}_{\mathcal{D}_h}\left(f, \hat{f}_{h+1}\right)$ for all $h \in [H]$. Finally, it returns the greedy policy $\pi_{\hat{f}}$ with respect to the estimate $\hat{f}$.

For a finite sample analysis of such sample based approximations, various conditions are required for near-optimality (Antos et al., 2008; Chen and Jiang, 2019). Specifically, the following three assumptions are usually taken in the general case:

1. *Realizability*: For all levels $h \in [H]$, the optimal value function lies in the hypothesis class: $Q^*_h \in \mathcal{F}_h$.

2. *Closedness*: For all levels $h \in [H]$, $\mathcal{T}_h f \in \mathcal{F}_h$ for any $f \in \mathcal{F}_{h+1}$.

3. *Exploratory dataset*: The data set $\mathcal{D}_h$ is exploratory in nature such that if the sampling distribution is μ, i.e. $(s_h, a_h) \sim \mu_h$, then there exists a constant C which satisfies: $\frac{\mathbb{P}^\pi_h(s,a)}{\mu_h(s,a)} \leq C$ for any policy π and pair (s, a). Here, $\mathbb{P}^\pi_h(s, a)$ denotes the probability of seeing pair (s, a) at level h when using policy π. The constant C is referred to as the concentrability coefficient in the literature.

Of the three conditions outlined above, realizability is the most natural one and is standard in supervised learning theory as well. The second condition of closedness is something specific to these approximate DP methods where we have to control the per iteration approximation error for $\hat{f}_h$ against the best possible fit of $\mathcal{T}_h \hat{f}_{h+1}$ (see Lemma 2.2). Lastly, the exploratory dataset is required to guarantee that the approximation error at each level is small for any possible roll-in distribution at level h. The stated condition is one of the strongest conditions assumed in the literature and different variations also exist (see (Scherrer, 2014) for a comparison).

Finally, Chen and Jiang (2019) recently provided a simplified analysis which showed that an ϵ-optimal policy can be learnt using FQI with probability at least $1 - \delta$, by using $O\left(\frac{H^8 C^2 \log \frac{|\mathcal{F}|}{\delta}}{\epsilon^4}\right)$ samples. In addition to FQI, a range of approaches have been studied for different settings and assumptions (see Uehara et al. (2021); Xie and Jiang (2021); Jin et al. (2021) and reference therein).

In Chapter 5, we will investigate a representation learning problem for low-rank MDPs where we will use the described FQI algorithm as our main offline planning primitive.

Offline vs off-policy RL The term offline RL is often confused with off-policy RL which constitutes a major class of learning algorithms in RL. In off-policy RL, the agent is not given a dataset which has been collected offline, and instead, interacts with the environment by taking actions. The 'off-policy' aspect here refers to the mismatch between the actions used in making updates in the agent's functions vs the actual actions taken by the current behavior policy. A popular instance for this is the well-known Q-learning method where the update for the Q-value is given as $Q_{t+1}(s_t, a_t) = (1 - \alpha) Q_t(s_t, a_t) + \alpha \left(r_t(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q_t(s_{t+1}, a')\right)$. Here, instead of using

the action a_{t+1} taken by the agent at the next step, the update equation uses the greedy action, and is hence, an off-policy update. Due to this commonly occurring misconception, offline RL is also referred to as batch RL, fixed-dataset policy optimization or learning with offline data in the existing literature.

2.3 Learning and Control in Linear Dynamical Systems

In the previous sections of this chapter, we have discussed the background and different learning settings for fixed-horizon episodic MDPs. (Episodic) MDPs are considered the most basic and fundamental model for reinforcement learning and as such have been a focus for the RL theory community. On the empirical side, one of the key applications of RL has been in the domain of robotics. In these robotics domains, the agent has to learn in a continuous control setting which has been studied extensively in the field of optimal control theory. As such, the RL community has shown a renewed interest in developing a non-asymptotic theory for linear-quadratic control, one of the fundamental problems in optimal control. In Chapter 6, we consider an online learning problem in a multi-task LQR problem. Here, we give a brief background on LQR and discuss recent results on system identification (model-learning) and optimal adaptive control (online exploration) in LQR systems.

Note The problem setting and results for LQR is substantially different from the MDP setting discussed earlier. As such, for the background in this section and the main results in Chapter 6, we use a different notation which conforms to the standard notation used in the optimal control community.

2.3.1 A brief introduction to LQR

Optimal control deals with the problem of operating a dynamical system while incurring minimum total cost. Linear quadratic control is the specialized instance where the dynamical system is governed by a system of linear equations and the cost is a quadratic function of the state variable and control inputs. We consider the problem of controlling a discrete time linear time-invariant system where the system state $x(t) \in \mathbb{R}^{d_x}$ evolves with time as follows:

$$x(t+1) = A_* x(t) + B_* u(t) + \eta(t+1), \tag{2.11}$$

where $u(t) \in \mathbb{R}^{d_u}$ is the control input and $\eta(t) \in \mathbb{R}^{d_x}$ is the noise process. The pair of matrices $A_* \in \mathbb{R}^{d_x \times d_x}$ and $B_* \in \mathbb{R}^{d_x \times d_u}$ are referred to as system (or dynamics) matrices, which we briefly denote as $\Theta_* = [A_* | B_*]$.

The systems matrices are unknown to the learning agent and the goal is to adaptively select the control inputs $u(t)$ to minimize the following average cost[6]:

$$\min \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T} x(t)^\top R_x x(t) + u(t)^\top R_u u(t)\right] \quad \text{s.t.} \quad x(t+1) = A_* x(t) + B_* u(t) + \eta(t+1) \tag{2.12}$$

where $R_x \in \mathbb{R}^{d_x \times d_x}$ and $R_u \in \mathbb{R}^{d_u \times d_u}$ are positive definite cost matrices for the problem. The optimal control policy for this infinite horizon problem is obtained by solving the Hamilton-Jacobi-Bellman equation and is given by a constant linear state feedback:

$$u(t) = K_\infty(A_*, B_*)x(t) \text{ where } K_\infty(A_*, B_*) = (R_u + B_*^\top P_\infty(A_*, B_*)B_*)^{-1}B_*^\top P_\infty(A_*, B_*)A_* \tag{2.13}$$

Here, $P_\infty(A_*, B_*) \in \mathbb{R}^{d_x \times d_x}$ is obtained as the solution of the discrete algebraic Riccati equation (DARE):

$$P = R_x + A_*^\top P A_* - A_*^\top P B_*(R_u + B_*^\top P B_*)^{-1}B_*^\top P A_* \tag{2.14}$$

When the LQR system (A_*, B_*) is *stabilizable* (defined later), it is well-known (eg. Kučera (1972)) that there exists a unique positive semi-definite solution $P_\infty(A_*, B_*)$ to the DARE equation in (2.14). The matrix $P_\infty(A_*, B_*)$ also leads to the value function equation which is given by $\mathcal{J}^*(A_*, B_*) = \min_K \mathcal{J}_{A_*,B_*}(K)$ where $\mathcal{J}_{A_*,B_*}(K)$ is the average cost when a state feedback K is used.

Notation. We will $z(t) \in \mathbb{R}^{d_x+d_u}$ to denote the concatenated vector $z(t)^\top := [x(t)^\top | u(t)^\top]$. We use $K(\Theta_*)$ and $P(\Theta_*)$ for the matrices $K_\infty(\Theta_*)$ and $P_\infty(\Theta_*)$ for simplicity. For any square matrix $A \in \mathbb{R}^{n\times n}$, $\rho(A)$ denotes the spectral radius $\rho(A) := \max(|\lambda_1|, \ldots, |\lambda_n|)$ where $\lambda_i \in \mathbb{C}$ are the eigenvalues.

In addition, we will frequently refer to the following properties of LQR system matrices in our discussion:

Definition 2.6 (Stabilizability)**.** *The linear dynamical system Θ is called stabilizable if there exists a state feedback K such that $\rho(A + BK) < 1$.*

Definition 2.7 (Controllability)**.** *The linear dynamical system Θ is called controllable if the controllability Grammian $[A\ AB\ A^2B\ \ldots\ A^{d_x-1}B]$ has full row rank.*

[6]Note that for infinite horizon stochastic LQR problem, we are considering an average reward criteria, which is different than the total reward criteria in the episodic MDP setting. For more discussions, see Section 2.1.5.

Intuitively, stabilizability implies that there exists a control policy which can prevent the state variable from blowing up and can be used to drive the state down to a 0 value. Similarly, controllability implies that there for any target state x and $T \geq d_x$, there exists a sequence of control inputs $u(0), u(1), \ldots, u(T-1)$ such that $x(T) = x$. Note that stabilizability is weaker than controllability: all controllable systems are stabilizable but the converse is not true.

2.3.2 System identification in LTI systems

A key problem in learning and control for LQR systems is to estimate the parameters of the underlying open-loop ($u(t) = 0$) or closed loop system ($A = A_* + B_*K$), usually referred to as *system identification*. In the next section, we will see that a simple identification procedure for estimating the system matrices (A_*, B_*) can be coupled with a perturbed controller to get near-optimal finite time learning guarantees. Thus, in this section, we discuss the problem of system identification for linear time-invariant dynamical systems (LTIDS) which evolves according to the Vector Auto-regressive (VAR) process:

$$x(t+1) = Ax(t) + \eta(t+1). \tag{2.15}$$

Above, $A \in \mathbb{R}^{d \times d}$ denotes the transition matrix of the system and $\eta(t+1)$ is a mean zero noise process. Note that the above setting includes systems that have longer memories and every state vector depends on multiple previous states. That is thanks to the commonly used concatenation technique that stacks state vectors in a larger new state which follows (6.4) (Faradonbeh et al., 2018a). If there is a lag of size q in the memory of a system of dimension d_0, then the new states will be of dimension $d = qd_0$.

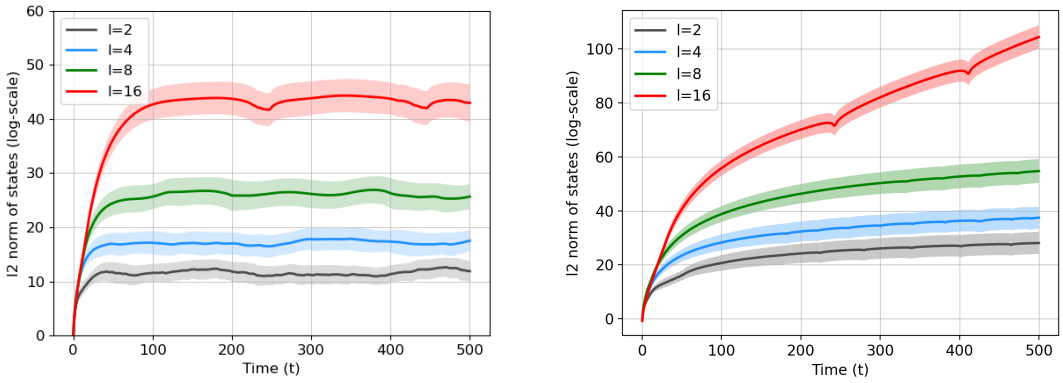
The statistical aspects of a linear system as described in (2.15) is dictated by the Jordan forms of matrix A. For matrix A, its Jordan decomposition is $A = P^{-1}\Lambda P$, where Λ is a block diagonal matrix; $\Lambda = \text{diag}(\Lambda_1, \ldots \Lambda_q)$, and for $i = 1, \ldots q$, each block $\Lambda_i \in \mathbb{C}^{l_i \times l_i}$ is a Jordan matrix of the eigenvalue λ_i. A Jordan matrix of size l for $\lambda \in \mathbb{C}$ is

$$\begin{bmatrix} \lambda & 1 & 0 & \ldots & 0 & 0 \\ 0 & \lambda & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & \lambda & 1 \\ 0 & 0 & 0 & \ldots & 0 & \lambda \end{bmatrix} \in \mathbb{C}^{l \times l}.$$

The temporal evolution of LTIDS introduces a number of technical challenges, not present in iid data. First, for the LTIDS to be able to operate for a decent time period, the state trajectories

in (2.15) need to be non-explosive in the sense that the spectral radius of the transition matrix A can be slightly larger than unit (Juselius et al., 2002). Further spectral properties of the transition matrices such as block-sizes in the Jordan decomposition of A are important as well, as follows. In Figure 2.1, we plot the logarithm of the magnitude of state vectors for linear systems of dimension $d = 32$. The figure on the left depicts stable systems and indicates the effect of the block-sizes l in the Jordan decomposition of the transition matrices for $l = 2, 4, 8, 16$. So, it demonstrates that the state vector scales exponentially with l.

Moreover, the case of transition matrices with eigenvalues close to, or on the unit circle is provided in the right panel in Figure 2.1. It illustrates that in the latter case, the state vectors grow polynomially as a function of time T. Finally, the recursive nature of (2.15) introduces dependencies between the observed data and the noise term, which require careful handling to establish the theoretical results.



(a) $\lambda_{\max}(A) < 1$ (b) $\lambda_{\max}(A) \approx 1$

Figure 2.1: Logarithm of the magnitude of the state vectors vs time, for different block-sizes (l) in the Jordan forms of the transition matrices.

Algorithm 2.1 OLS algorithm for system identification

1: **Input:** Data $z(0), \ldots, z(T-1), x(T)$ for given LQR system.
2: **return** $[\widehat{A}, \widehat{B}]$ and $\Sigma = \sum_{t=0}^{T-1} z(t)z(t)^\top$, where

$$\widehat{A}, \widehat{B} := \underset{A,B}{\operatorname{argmin}} \sum_{t=0}^{t-1} \|x(t+1) - [A|B]\, z(t)\|_2^2 \tag{2.16}$$

2.3.3 Online adaptive control in LQR

During learning, the system matrices are unknown to the agent and the goal is to adaptively choose cnotrol inputs $U(t)$ and use the state observations over T rounds to minimize the incurred cost $\sum_{t=0}^{T-1} = \sum_{t=0}^{T-1} x(t)' R_x x(t) + u(t)' R_u u(t)$. The cost matrices R_x and R_u are known to the agent. In an online stochastic LQR control problem, the goal of the agent is to minimize the regret incurred by the agent as compared to the optimal linear controller:

$$\text{Regret}(A_*, B_*, T) = \left[\sum_{t=0}^{T-1} c(x(t), u(t))\right] - T\mathcal{J}^*(A_*, B_*) \tag{2.17}$$

In the subsequent portion of this section, we briefly outline the two major approaches to achieve near-optimal regret guarantees for online adaptive control in LQR. The results of all discussed papers are summarized in Table 2.4.

Work	Regret	Setting	K_0	Approach
Abbasi-Yadkori and Szepesvári (2011)	$(d_x + d_u)^{(d_x+d_u)}\sqrt{T}$	C	No	OFU
Faradonbeh et al. (2020d)	$\sqrt{T}$	S	Yes	
Lale et al. (2020)	$\text{poly}(d)\sqrt{T}$	S/C	No	
Faradonbeh et al. (2020c)	$\sqrt{T}$	S	Yes	CE
Mania et al. (2019)	$\sqrt{(d_x + d_u)^3 T}$	C	Yes	
Simchowitz and Foster (2020)	$\sqrt{\max(d_x, d_u^2) d_x T}$	S	Yes	

Table 2.4: Notable no-regret results for Online Adaptive LQR.
S := Stabilizable, C := Controllable, K_0 := requires initial stabilizing controller.

2.3.3.1 Algorithms using optimism in the face of uncertainty

In early work on LQR systems, typically a forced exploration scheme was used where exploratory actions are taken according to a fixed and appropriately designed schedule. Campi and Kumar (1996) proposed an exploration method which they call Bet on the Best (BOB) principle where a high-probability confidence set is built for the system parameters and an optimal controller for the pair with the minimum cost is chosen. However, in their analysis, Campi and Kumar (1996) only show an asymptotic result that the optimistic policy converges to the optimal policy in the limit. Using a similar OFU principle, Abbasi-Yadkori and Szepesvári (2011) implement an optimistic controller by constructing a high-probability confidence set similar to linear bandits and MDPs, as described in Section 2.2.1.2. The confidence set is built by using techniques from self-normalized processes (see Lemma A.4) to bound the least squares estimation error for the estimator in Algorithm 2.1. Specifically, using the martingale bounds for a self-normalized martingale process Lemma A.4,

Abbasi-Yadkori and Szepesvári (2011) derived a high probability confidence set of the following form:

$$C_t(\delta) = \left\{\Theta \in \mathbb{R}^{d_x+d_u} : \mathrm{tr}\left((\widehat{\Theta} - \Theta_*)^\top V_t (\widehat{\Theta} - \Theta_*)\right) \leq \beta_t(\delta)\right\}$$

where $V_t = \lambda I + \sum_{s=0}^{t-1} z(s)z(s)^\top$ and we have $\mathbb{P}[\Theta_* \in C_t(\delta), t = 1, 2, \ldots] \geq 1 - \delta$. Given this confidence set, the control policy is selected optimistically: $\tilde{\Theta}_t = \mathrm{argmin}_{\Theta \in C_t(\delta)} \mathcal{J}_*(\Theta)$. Along with the confidence set construction, the authors assume that the parameter Θ_* belongs to a bounded set and each system belonging to this set is controllable.

Faradonbeh et al. (2020d) also propose an optimism based online adaptive control algorithm but enforce a weaker assumption of stabilizability instead of controllability. Another key difference of the algorithm here is that instead of estimating A_* and B_* separately using least squares over input-output pairs $(z(t), x(t+1))$, the closed loop matrix $D_t = A_* + B_* K(\tilde{\Theta}_t)$ is estimated for each round such that $\mathbb{P}\left[\left\|V_t^{1/2}\left(\hat{D}_t - D_t\right)^\top\right\| \geq \beta_t(\delta)\right] \leq \delta$ with $V_t = \sum_{s=0}^{t-1} x(s)x(s)^\top$. In this case, the confidence set is defined as the set Γ_t of all system matrices Θ such that the closed-loop matrix $A + BK_t$ belongs to the confidence set. The overall confidence set is thus updated as $C_t(\delta) = C_{t-1}(\delta) \cap \Gamma_t$ where Γ_t is the high-probability closed-loop matrix confidence set. The regret result for this algorithm is again a high-probability regret bound of order $\widetilde{O}(\sqrt{T})$. In contrast to the algorithm in Abbasi-Yadkori and Szepesvári (2011), the authors here require access to an initial bounded stabilizing set C_0 such that for every $\Theta \in C_0$, the system will be stable if the optimal feedback $K_\infty(\Theta)$ is applied.

Finally, in a recent preprint Lale et al. (2020), the OFU principle is augmented by using additional forced exploration leading to the algorithm ExpOpt. The authors show that ExpOpt can get polynomial dimension dependence in the $\widetilde{O}(\sqrt{T})$ regret bound for both stabilizable and controllable settings. The main idea in their algorithm is to add exploration to the optimistic controller: $u(t) = K(\tilde{\Theta}_t) + g(t)$ where $g_t \sim \mathcal{N}(0, \sigma_g^2)$ is a Gaussian perturbation. This additional persistent excitation is added to the policy for an initial exploration phase T_w leading to a regret guarantee of $\widetilde{O}\left((d_x + d_u)^{d_x+d_u}\sqrt{T_w} + \mathrm{poly}(d)\sqrt{T - T_w}\right)$. The exploration phase length doesn't scale with T, hence, the polynomial in dimension regret guarantee.

Episode schedule In the regret decomposition used in these papers, it has been shown that a large number of changes to the policy K_t lead to larger total regret. As such, all these algorithms proceed in epochs based on a *doubling* scheme. In Abbasi-Yadkori and Szepesvári (2011) and Lale et al. (2020), the optimistic control policy is updated only when the determinant of the design matrix V_t has doubled since the last controller update. It can be easily shown that the total number of policy changes for such a doubling scheme scales as $O(\log T)$. Similar to this, a simpler and direct

doubling scheme is used in Faradonbeh et al. (2020d) where the length of epoch i scales as $\widetilde{O}(\gamma^i)$ for a *reinforcement rate* γ which again implies logarithmic number of policy changes.

2.3.3.2 Certainty equivalent control with persistent excitation

In quite an exciting development, it has recently been shown that for LQR systems, careful exploration via optimism or randomization is not required and one can instead get away with a more straightforward approach: certainty equivalence. In a certainty equivalent controller, a model of the system is fit to observed transition data and a control policy is designed by treating the estimate as the truth.

The first $\widetilde{O}\left(\sqrt{T}\right)$-regret result for a certainty equivalent controller was shown in Faradonbeh et al. (2020c). The authors showed that certainty equivalent controller coupled with ϵ-greedy exploration can achieve $\widetilde{O}\left(\sqrt{T}\right)$ regret when a stabilizing controller K_0 is known apriori. In the proposed perturbed greedy regulator, the control policy is set in epochs of doubling lengths such that $u(t) = K(\hat{\Theta}_t) + g(t)$ where $g_t \sim \mathcal{N}(0, \sigma_t^2 I)$. Here, $\sigma_t^2 \propto \frac{1}{\tau_t^2 \gamma^{\tau_t/2}}$ where τ_t is the epoch length.

A similar result for certainty equivalence was shown in Mania et al. (2019) where a perturbation analysis for the solution to the DARE equation in (2.14) is made. Specifically, it is shown that if the estimates $(\widehat{A}, \widehat{B})$ for the system matrices (A_*, B_*) satisfy $\|\widehat{A} - A_*\| \leq \varepsilon$ and $\|\widehat{B}B - B_*\| \leq \varepsilon$ for some $\varepsilon > 0$, then the nominal controller $K(\widehat{\Theta})$ achieves $\widehat{\mathcal{J}} - \mathcal{J}_* \leq C(A_*, B_*, R_x, R_u)\varepsilon^2$. The aforementioned guarantee is obtained using the following two steps:

1. Perturbation (sensitivity) analysis of DARE solution: use Riccati perturbation theory to show that if $\|\widehat{A} - A_*\| \leq \varepsilon$ and $\|\widehat{B} - B_*\| \leq \varepsilon$, then the solutions to the Riccati equation satisfy $\|\widehat{P} - P\| \leq f(\varepsilon)$ where $f(\varepsilon) = \widetilde{O}(\varepsilon)$.

2. Mismatch between nominal controller and optimal controller: using the perturbation bound on $\hat{P}$, show that we also have $\|\widehat{K} - K\| \leq \widetilde{O}(f(\varepsilon))$ which in turn implies $\mathcal{J}(K) - \mathcal{J}_* \leq (d_x + d_u)C(A_*, B_*, R_x, R_u)f(\varepsilon)^2$.

In the end-to-end learning algorithm, in each epoch, the agent checks if the estimates $(\widehat{A}, \widehat{B})$ of the system matrices are accurate enough for the Riccati perturbation bounds to come into effect. If true, it can be shown that the certainty equivalent controller is stabilizing and incurs low regret. In case of failure, the agent falls back to the stabilizing controller K_0 for the epoch, and adds exploratory noise with constant scale.

Finally, in Simchowitz and Foster (2020), the authors refine the perturbation analysis in Mania et al. (2019) and show a bound on $\|\widehat{P} - P\|$ when the estimates satisfy $\|\widehat{A} - A\|_F^2 \leq \varepsilon$ and $\|\widehat{B} - B\|_F^2 \leq \varepsilon$. Consequently, they also use a perturbed certainty equivalence controller to show a regret bound of $\widetilde{O}\left(\sqrt{\max(d_x, d_u^2) d_x T}\right)$ and therefore show an improved dependence on

the dimensions d_x, d_u and horizon T (ignoring log factors) when $d_u \ll d_x$[7]. In addition to the improved regret bound, their work also relaxes the controllability assumption in Mania et al. (2019) to stabilizability.

In Chapter 6, we study a multi-task LQR control problem where we propose the first joint control procedure for LQR systems, under the assumption that the transition matrices share a common basis. Our control algorithm is based on the certainty equivalent controller, as proposed and analyzed by Simchowitz and Foster (2020).

[7]This upper bound is for the case where the noise process w_t is a martingale process and is not assumed to be independent.

CHAPTER 3

Online Learning in Contextual Markov Decision Processes

Reinforcement learning is considered to be an ideal framework for tackling sequential decision making problems with potential applications ranging from web advertising and portfolio optimization, to healthcare applications like adaptive drug treatment. However, despite the empirical success of RL in simulated domains such as boardgames and video games, it has seen limited use in real world applications because of the inherent trial-and-error nature of the paradigm. In addition to these concerns, for the applications listed above, we have to essentially design adaptive methods for a *population* of users instead of a single system. In this chapter, we address these two concerns in the following manner: (1) we model the interaction via a contextual Markov decision processes framework which accounts for the heterogeneity in the population while utilizing a shared structure and (2) we propose sample efficient online learning algorithms for a variety of structural assumptions on how different environments in the population relate to each other. In this setting, an RL agent interacts with multiple tasks in a sequence and the goal is to adaptively optimize the behavior policy, in such a way that utilizes the shared structure across all tasks, while at the same time, exploits the idiosyncratic properties to adapt to each task individually. We show formally that an agent can utilize a task representation during learning under the assumption that the MDP parameters share a structured mapping via this idiosyncratic representation. The chapter proposes and analyzes provably efficient exploration algorithms under the PAC and regret criteria for varying contextual mappings.

3.1 Introduction

Consider a basic sequential decision making problem in healthcare, namely that of learning a treatment policy for patients to optimize some health outcome of interest. One could model the interaction with every patient as a Markov Decision Process (MDP). In *precision or personalized*

medicine, we want the treatment to be personalized to every patient. At the same time, the amount of data available on any given patient may not be enough to personalize well. This means that modeling each patient via a different MDP will result in severely suboptimal treatment policies. The other extreme of pooling all patients' data results in more data but most of it will perhaps not be relevant to the patient we currently want to treat. We therefore face a trade-off between having a large amount of shared data to learn a single policy, and, finding the most relevant policy for each patient. A similar trade-off occurs in other applications in which the agent's environment involves humans, such as in online tutoring and web advertising.

A key observation is that in many personalized decision making scenarios, some side information is available about individuals which might help in designing personalized policies and also help pool the interaction data across the right subsets of individuals. Examples of such data include laboratory data or medical history of patients in healthcare, user profiles or history logs in web advertising, and student profiles or historical scores in online tutoring. The use of such side information can also be found in RL literature: Ammar et al. (2014) developed a multi-task policy gradient method where the context is used for transferring knowledge between tasks; Killian et al. (2016) used parametric forms of MDPs to develop models for personalized medicine policies for HIV treatment.

Access to such side information should allow learning of better policies even with a limited amount of interaction with individual users. We refer to this side-information as *context* and adopt an augmented model called *Contextual Markov Decision Process* (CMDP) proposed by Hallak et al. (2015). We assume that contexts are fully observed and available before the interaction starts for each new MDP.[1] In this chapter, we study the efficiency of learning in CMDPs when contexts can potentially be generated adversarially. We consider two concrete settings of learning in a CMDP with continuous contexts: (1) individual MDPs vary smoothly with the contexts and (2) a stronger structural assumption, where the MDP parameters are obtained by a (generalized) linear function of the context. Our results highlight the hardness of learning for the general assumption of smoothness and shows that linearity, as a structural assumption, allows us to design provably efficient learning methods for CMDPs.

Chapter outline The remaining chapter is organized as follows: Section 3.2 sets up the contextual MDP problem formally along with preliminary background. In Section 3.3, we study the CMDP setting under the mistake bound criteria (described in Section 2.2.1.1) for two structural assumptions: (1) smoothly varying contextual mappings (Section 3.3.1), and, (2) linearly parameterized CMDPs (Section 3.3.3). Section 3.4 studies the problem under the regret criteria (described in Section 2.2.1.2) for a generalized linear setting and subsumes the linear case analyzed in Section 3.3.3. Finally, we

[1] Hallak et al. (2015) assumes *latent* contexts, which results in significant differences from our work in application scenarios, required assumptions, and results. See detailed discussion in Section 3.5.

conclude the chapter in Section 3.5 with a brief discussion of the results.

3.2 Problem Setup

In this section, we describe the problem formulation, followed by the interaction protocol between the environment and the learner.

Problem setup and notation We consider a contextual episodic MDP problem where the agent receives additional side information about the MDP dynamics before each episode. Below we formally introduce the contextual model, inspired by Hallak et al. (2015):

Definition 3.1 (Contextual MDP)**.** *A contextual Markov Decision Process (CMDP) is defined as a tuple $(\mathcal{X}, \mathcal{S}, \mathcal{A}, \mathcal{M})$ where $\mathcal{X}$ is the context space (assumed to lie in some Euclidean space), $\mathcal{S}$ is the state space, and $\mathcal{A}$ is the action space. $\mathcal{M}$ is a function which maps a context $x \in \mathcal{X}$ to MDP parameters $\mathcal{M}(x) = \{P_x(\cdot|\cdot,\cdot), R_x(\cdot,\cdot), \mu_x(\cdot)\}$.*

The MDP for a context x is denoted by M_x. For simplification, the initial state distribution is assumed to be the same irrespective of the context and rewards are assumed to be bounded between 0 and 1. We denote $|\mathcal{S}|, |\mathcal{A}|$ by S, A respectively. We also assume that the context space is bounded, and for any $x \in \mathcal{X}$ the ℓ_2 norm of x is upper bounded by some constant.

Protocol and Efficiency Criterion We consider the online learning scenario with the following protocol: For $t = 1, 2, \ldots$:

1. Observe context $x_t \in \mathcal{X}$.
2. Choose a policy π_t (based on x_t and previous episodes).
3. Experience an episode in $M_t \equiv M_{x_t}$ using π_t.

Note that we do not make any distributional assumptions over the context sequence. Instead, the context sequence can be chosen in an arbitrary and potentially *adversarial* manner. For conciseness, we will use the notation (P_t, R_t) to denote the contextual mappings P_{x_t}, R_{x_t}.

A natural criteria for judging the efficiency of the algorithm is to look at the number of episodes where it performs sub-optimally. Therefore, we consider the PAC-mistake bound criteria that we discussed in Section 2.2.1.1 and propose a sample complexity analysis for two structural assumptions in Section 3.3. Although, we do give PAC bounds for the algorithms given below, the reader should make note that, we have not made explicit attempts to achieve the tightest possible result. We use the RMAX (Brafman and Tennenholtz, 2002) algorithm as the base of our construction to handle

exploration-exploitation because of its simplicity. Our approach can also be combined with the other PAC algorithms (Strehl and Littman, 2005; Dann and Brunskill, 2015) for improved dependence on S, A and H.

Another way to evaluate an agent in this online CMDP setting would be to look at the regret incurred by the agent (Section 2.2.1.2) and analyze the total suboptimality of the agent's actions over a sequence of T episodes. Hence, in Section 3.4, we consider a contextual MDP setting where the underlying structure is parameterized by a generalized linear map. We propose and analyse provably efficient optimistic/randomized exploration methods and generalize and improve the previously analyzed linear CMDP algorithms.

3.3 Algorithms for CMDPs with Mistake Bound Guarantees

In this section, we consider two concrete settings of learning in a CMDP with continuous contexts. In the first setting, the individual MDPs vary in an arbitrary but smooth manner with the contexts, and we propose our COVER-RMAX algorithm in Section 3.3.1 with PAC mistake bounds. The innate hardness of learning in this general case is captured by our lower bound construction in Section 3.3.2. To show that it is possible to achieve significantly better sample complexity in more structured CMDPs, we study the setting where contexts create linear combinations of a finite set of fixed but unknown MDPs. We use the KWIK (Knows What It Knows) framework to devise the KWIKLR-RMAX algorithm in Section 3.3.3 and also provide a PAC upper bound for the algorithm.

3.3.1 Mistake bounds for smooth contextual MDP

The key motivation for our contextual setting is that sharing information/data among different contexts might be helpful. A natural way to capture this is to assume that the MDPs corresponding to similar contexts will themselves be similar. This can be formalized by the following smoothness assumption:

Definition 3.2 (Smoothness)**.** *Given a CMDP $(\mathcal{X}, \mathcal{S}, \mathcal{A}, \mathcal{M})$ and a distance metric over the context space $\mathrm{dis}(\cdot, \cdot)$, if for any two contexts x_1, $x_2 \in \mathcal{X}$, we have the following constraints:*

$$\|P_{x_1}(\cdot|s,a) - P_{x_2}(\cdot|s,a)\|_1 \leq L_p \mathrm{dis}(x_1, x_2)$$
$$|R_{x_1}(s,a) - R_{x_2}(s,a)| \leq L_r \mathrm{dis}(x_1, x_2)$$

then, the CMDP is referred as a smooth CMDP with smoothness parameters L_p and L_r.

Throughout this section, the distance metric and the constants L_p and L_r are assumed to be known. This smoothness assumption allows us to use a modified version of RMAX (Brafman and

Tennenholtz, 2002) and provide an analysis for smooth CMDPs similar to existing literature on PAC bounds in MDPs (Kearns and Singh, 2002; Strehl et al., 2009; Strehl and Littman, 2005). Given the transition dynamics and the expected reward functions for each state-action pair in a finite MDP, computing the optimal policy is straightforward. The idea of RMAX is to distinguish the state-action pairs as *known* or *unknown*: a state-action pair is known if it has been visited enough number of times, so that the empirical estimates of reward and transition probabilities are near-accurate due to sufficient data. A state s becomes *known* when (s, a) is *known* for all actions a. RMAX then constructs an auxiliary MDP which encourages optimistic behaviour by assigning maximum reward (hence the name RMAX) to the remaining *unknown* states. The optimal policy in the auxiliary MDP ensures that one of the following must happen: 1) it achieves near-optimal value, or, 2) it visits unknown states and accumulates more information efficiently.

Formally, for a set of known states $\mathcal{K}$, we define an (approximate) *induced MDP* $\widehat{M}_{\mathcal{K}}$ in the following manner. Let $n(s, a)$ and $n(s, a, s')$ denote the number of observations of state-action pair (s, a) and transitions (s, a, s') respectively. Also, let $\sum_{i=1}^{n(s,a)} r^{(i)}(s, a)$ denote the total reward obtained from state-action pair (s, a). For each $s \in \mathcal{K}$, define the values

$$\begin{aligned} P_{\widehat{M}_{\mathcal{K}}}(s'|s, a) &= \frac{n(s, a, s')}{n(s, a)}, \\ R_{\widehat{M}_{\mathcal{K}}}(s, a) &= \frac{\sum_{i=1}^{n(s,a)} r^{(i)}(s, a)}{n(s, a)}. \end{aligned} \tag{3.1}$$

For each $s \notin \mathcal{K}$, define the values as $P_{\widehat{M}_{\mathcal{K}}}(s'|s, a) = \mathbb{1}\{s' = s\}$ and $R_{\widehat{M}_{\mathcal{K}}}(s, a) = 1$.

RMAX uses the certainty equivalent policy computed for this induced MDP and performs balanced wandering (Kearns and Singh, 2002) for unknown states. Balanced wandering ensures that all actions are tried equally and fairly for *unknown* states. Assigning maximum reward to the *unknown* states pushes the agent to visit these states and provides the necessary exploration impetus. The generic template of RMAX is given in Algorithm 3.1.

Algorithm 3.1 RMAX Template for CMDP

```
INITIALIZE(S, A, 𝒳, ϵ, δ)
for each episode t = 1, 2, ··· do
    Receive context x_t ∈ 𝒳
    Set 𝒦, M̂_𝒦 using PREDICT(x_t, s, a) for all (s, a). π ← π*_{M̂_𝒦}
    for h = 0, 1, ··· H − 1 do
        if s_h ∈ 𝒦 then
            Choose a_h := π_h(s_h)
        else
            Choose a_h : (s_h, a_h) is unknown
            UPDATE(x_t, s_h, a_h, (s_{h+1}, R_h))
```

For the contextual case, there would be an infinite number of such MDPs. The idea behind our algorithm is that, close enough contexts can be grouped together and be considered as a single MDP. Utilizing the boundedness of the context space $\mathcal{X}$, we can create a *cover* of $\mathcal{X}$ with finitely many balls $B_r(o_i)$ of radius r centered at $o_i \in \mathbb{R}^d$. The bias introduced by ignoring the differences among the MDPs in the same ball can be controlled by tuning the radius r. Doing so allows us to pool together the data from all MDPs in a ball, so that we avoid the difficulty of infinite MDPs and instead only deal with finitely many of them. The size of the cover, i.e. the number of balls, can be measured by the notion of *covering numbers* (see Section 5.1 in Wainwright (2019)), defined as

$$\mathcal{N}(\mathcal{X}, r) = \min\{|\mathcal{Y}| \,:\, \mathcal{X} \subseteq \cup_{y \in \mathcal{Y}}\, B_r(y)\}.$$

The resulting algorithm, COVER-RMAX, is obtained by using the subroutines in Algorithm 3.2, and we state its sample complexity guarantee in Theorem 3.1.

Algorithm 3.2 COVER-RMAX

function INITIALIZE($\mathcal{S}, \mathcal{A}, \mathcal{X}, \epsilon, \delta$)
  Create an r_0-cover of $\mathcal{X}$ with $r_0 = \min\left(\frac{\epsilon}{8HL_p}, \frac{\epsilon}{8L_r}\right)$
  Initialize counts for all balls $\mathcal{B}(o_i)$
function PREDICT(x, s, a)
  Find j such that $x \in \mathcal{B}(o_j)$
  if $n_j(s,a) < m$ **then**
    return $\widehat{P}_x(\cdot|s,a)$ and $\widehat{R}_x(s,a)$ using (3.1)
  else
    return *unknown*
function UPDATE($x, s, a, (s', r)$)
  Find j such that $x \in \mathcal{B}(o_j)$
  if $n_j(s,a) < m$ **then**
    Increment counts and rewards in $\mathcal{B}(o_j)$

Theorem 3.1 (PAC bound for COVER-RMAX)**.** *For any input values $0 < \epsilon, \delta \leq 1$ and a CMDP with smoothness parameters L_p and L_r, with probability at least $1-\delta$, the* COVER-RMAX *algorithm produces a sequence of policies $\{\pi_t\}$ which yield at most*

$$O\left(\frac{NH^4SA}{\epsilon^3}\left(S + \ln\frac{NSA}{\delta}\ln\frac{N}{\delta}\right)\right)$$

non-ϵ-optimal episodes, where $N = \mathcal{N}(\mathcal{X}, r_0)$ and $r_0 = \min\left(\frac{\epsilon}{8H^2L_p}, \frac{\epsilon}{8HL_r}\right)$.

Proof. We first of all carefully adapt the analysis of RMAX by Kakade (2003) to get the PAC bound for an episodic MDP. Let m be the number of visits to a state-action pair after which the model's

estimate $\widehat{P}(\cdot|s,a)$ for $p(\cdot|s,a)$ has an ℓ_1 error of at most $\epsilon/4H^2$ and reward estimate $\widehat{R}(s,a)$ has an absolute error of at most $\epsilon/4H$. We can show that:

Lemma 3.2. *Let M be an MDP with the fixed horizon H. If $\hat{\pi}$ is the optimal policy for $\widehat{M}_{\mathcal{K}}$ as computed by* RMAX*, then for any starting state s_0, with probability at least $1-2\delta$, we have $V_M^{\hat{\pi}} \geq V_M^* - 2\epsilon$ for all but $O(\frac{mHSA}{\epsilon}\ln\frac{1}{\delta})$ episodes.*

Now instead of learning the model for each contextual MDP separately, the algorithm combines the data within each ball. To control the bias induced by sharing data, the radius r for the cover is set to be $r \leq r_0 = \min\left(\frac{\epsilon}{8H^2L_p}, \frac{\epsilon}{8HL_r}\right)$. Further, the value of m, which is the number of visits after which a state becomes *known* for a ball, is set as $m = \frac{128(S\ln 2+\ln\frac{SA}{\delta})H^4}{\epsilon^2}$. This satisfies the assumptions in Lemma 3.2, whereby, we obtain an upper bound on number of non-ϵ episodes in a single ball (generated by COVER-RMAX) as $O\left(\frac{H^5SA}{\epsilon^3}\left(S+\ln\frac{SA}{\delta}\ln\frac{1}{\delta}\right)\right)$ with probability at least $1-\delta$.

Setting the individual failure probability to be $\delta/N(\mathcal{X}, r_0)$ and using the union bound, we get the stated PAC bound. A detailed proof can be found in Section 3.8.1.1. □

The obtained PAC bound has linear dependence on covering number of the context space. In case of a d-dimensional Euclidean metric space, the covering number is of the order $O(\frac{1}{r^d})$. However, we show in Section 3.3.2, that, the dependence on the covering number is at least linear in the worst case and indicate the difficulty of optimally learning in such cases.

3.3.2 Hardness of online learning in smooth contextual MDPs

We prove a lower bound on the number of sub-optimal episodes for any learning algorithm in a smooth CMDP which shows that a linear dependence on the covering number of the context space is unavoidable. As far as we know, there is no existing way of constructing PAC lower bounds for continuous state spaces with smoothness, so we cannot simply augment the state representation to include context information. Instead, we prove our own lower bound in Theorem 3.3 which builds upon the work of Dann and Brunskill (2015) on lower bounds for episodic finite MDPs and of Slivkins (2014) on lower bounds for contextual bandits.

Theorem 3.3 (Lower bound for smooth CMDP)**.** *There exists constants δ_0, ϵ_0, such that for every $\delta \in (0, \delta_0)$ and $\epsilon \in (0, \epsilon_0)$, any algorithm that satisfies a PAC guarantee for (ϵ, δ) and computes a sequence of deterministic policies for each context, there is a hard CMDP $(\mathcal{X}, \mathcal{S}, \mathcal{A}, \mathcal{M})$ with smoothness constant $L_p = 1$, such that*

$$\mathbb{E}[B] = \Omega\left(\frac{\mathcal{N}(\mathcal{X}, \epsilon_1)H^2SA}{\epsilon^2}\right) \tag{3.2}$$

where B is the number of sub-optimal episodes and $\epsilon_1 = \frac{1280\epsilon e^4}{(H-2)}$.

The overall idea is to embed multiple MDP learning problems in a CMDP, such that the agent has to learn the optimal policy in each MDP separately and cannot generalize across them. We show that the maximum number of problems that can be embedded scales with the covering number, and the result follows by incorporating known PAC lower bound for episodic MDPs. We refer the reader to Section 3.8.1.2 for the proof.

3.3.3 KWIKLR-RMAX: A PAC-efficient algorithm for linear CMDPs

From the previous section, it is clear that for a contextual MDP with just smoothness assumptions, exponential dependence on context dimension is unavoidable. Further, the computational requirements of our COVER-RMAX algorithm scales with the covering number of the context space. As such, in this section, we focus on a more structured assumption about the mapping from context space to MDPs and show that we can achieve substantially improved sample and computational efficiency.

The specific assumption we make in this section is that the model parameters of an individual MDP M_x is the linear combination of the parameters of d base MDPs, i.e.,

$$\begin{aligned} P_x(s'|s,a) &= x^\top \begin{bmatrix} P^1(s'|s,a) \\ \vdots \\ P^d(s'|s,a) \end{bmatrix} := x^\top P(s,a,s'), \\ R_x(s,a) &= x^\top \begin{bmatrix} R^1(s,a) \\ \vdots \\ R^d(s,a) \end{bmatrix} := x^\top R(s,a). \end{aligned} \tag{3.3}$$

We use $P(s,a,s')$ and $R(s,a)$ as shorthand for the $d \times 1$ vectors that concatenate the parameters from different base MDPs for the same s, a (and s'). The parameters of the base MDPs (P^i and R^i) are unknown and need to be recovered from data by the learning agent, and the combination coefficients are directly available which is the context vector x itself. This assumption can be motivated in an application scenario where the user/patient responds according to her characteristic distribution over d possible behavioural patterns.

A mathematical difficulty here is that for an arbitrary context vector $x \in \mathbb{R}^d$, $P_x(\cdot|\cdot,\cdot)$ is not always a valid transition function and may violate non-negativity and normalization constraints. Therefore, we require that $x \in \Delta_{d-1}$, that is, x stays in the probability simplex so that $P_x(\cdot|\cdot,\cdot)$ is always valid.

3.3.3.1 KWIKLR-RMAX

We first explain how to estimate the model parameters in this linear setting, and then discuss how to perform exploration properly.

Model estimation Recall that in Section 3.3.1, the COVER-RMAX algorithm treats the MDPs whose contexts fall in a small ball as a single MDP, and estimates its parameters using data from the *local* context ball. In this section, however, we have a *global* structure due to our parametric assumption (d base MDPs that are shared across all contexts). This implies that data obtained at a context may be useful for learning the MDP parameters at another context that is far away, and to avoid the exponential dependence on d we need to leverage this structure and generalize globally across the entire context space.

Due to the linear combination setup, we use linear regression to replace the estimation procedure in (3.1): in an episode with context x, when we observe the state-action pair (s, a), a next-state s_{next} will be drawn from $P_x(\cdot|s, a)$.[2] Therefore, the indicator of whether s_{next} is equal to s' forms an unbiased estimate of $P_x(s'|s, a)$, i.e., $\mathbb{E}_{s_{\text{next}} \sim P_x(\cdot|s,a)} [\mathbb{I}[s_{\text{next}} = s']] = P_x(s'|s, a) = x^\top P(s, a, s')$. Based on this observation, we can construct a feature-label pair

$$(x,\ \mathbb{I}[s_{\text{next}} = s']) \tag{3.4}$$

whenever we observe a transition tuple (s, a, s_{next}) under context x, and their relationship is governed by a linear prediction rule with $P(s, a, s')$ being the coefficients. Hence, to estimate $P(s, a, s')$ from data, we can simply collect the feature-label pairs that correspond to this particular (s, a, s') tuple, and run linear regression to recover the coefficients. The case for reward function is similar, hence, not discussed.

If the data is abundant (i.e., (s, a) is observed many times) and exploratory (i.e., the design matrix that consists of the c vectors for (s, a) is well-conditioned), we can expect to recover $P(s, a, s')$ accurately. But how to guarantee these conditions? Since the context is chosen adversarially, the design matrix can indeed be ill-conditioned.

Observe, however, when the matrix is ill-conditioned and new contexts lie in the subspace spanned by previously observed contexts, we can make accurate predictions despite the inability to recover the model parameters. An *online* linear regression (LR) procedure will take care of this issue, and we choose KWIKLR (Walsh et al., 2009) as such a procedure.

The original KWIKLR deals with scalar labels, which can be used to decide whether the estimate of $P_x(s'|s, a)$ is sufficiently accurate (*known*). A (s, a) pair then becomes known if (s, a, s') is known for all s'. This approach, however, generally leads to a loose analysis, because there is no

[2] Here we use s_{next} to denote the random variable, and s' to denote a possible realization.

need to predict $P_x(s'|s,a)$ for each individual s' accurately: if the estimate of $P_x(\cdot|s,a)$ is close to the true distribution under L_1 error, the (s,a) pair can already be considered as known. We extend the KWIKLR analysis to handle vector-valued outputs, and provide tighter error bounds by treating $P_x(\cdot|s,a)$ as a whole. Below we introduce our extended version of KWIKLR, and explain how to incorporate the knownness information in RMAX skeleton to perform efficient exploration.

Identifying known (s,a) with KWIKLR

The KWIKLR-RMAX algorithm we propose for the linear setting still uses RMAX template (Algorithm 3.1) for exploration: in every episode, it builds the induced MDP $\widehat{M}_{\mathcal{K}}$, and acts greedily according to its optimal policy with balanced wandering. The major difference from COVER-RMAX lies in how the set of known states $\mathcal{K}$ are identified and how $\widehat{M}_{\mathcal{K}}$ is constructed, which we explain below (see pseudocode in Algorithm 3.3).

At a high level, the algorithm works in the following way: when constructing $\widehat{M}_{\mathcal{K}}$, the algorithm queries the KWIK procedure for estimates $\widehat{P}_x(\cdot|s,a)$ and $\widehat{R}_x(s,a)$ for every pair (s,a) using PREDICT(x,s,a). The KWIK procedure either returns $\perp$ (don't know), or returns estimates that are guaranteed to be accurate. If $\perp$ is returned, then the pair (s,a) is considered as unknown and s is associated with $R_{\max}$ reward for exploration. Such optimistic exploration ensures significant probability of observing (s,a) pairs on which the method predicts $\perp$. If such pairs are observed in an episode, KWIKLR-RMAX calls UPDATE with feature-label pairs formed via (3.4) to make progress on estimating parameters for unknown state-action pairs.

Next we walk through the pseudocode and explain how PREDICT and UPDATE work in detail. Then we prove an upper bound on the number of updates that can happen (i.e., the **if** condition holds on line 7), which forms the basis of our analysis of KWIKLR-RMAX.

In Algorithm 3.3, matrices Q and W are initialized for each (s,a) using INITIALIZE$(\cdot)$ and are updated over time. Let $X_t(s,a)$ be the design matrix at episode t, where each row is a context c_τ such that (s,a) was observed in episode $\tau < t$. By matrix inverse rules, we can verify that the update rule on line 6 essentially yields $Q_t(s,a) = (I + X_t^\top X_t)^{-1}$, where $Q_t(s,a)$ is the value of $Q(s,a)$ in episode t. This is the inverse of the (unnormalized and regularized) empirical covariance matrix, which plays a central role in linear regression analysis. The matrix W accumulates the outer product between the feature vector (context) c and the one-hot vector label $y = (\{\mathbb{1}[s_{\text{next}} = s']\}_{\forall s' \in \mathcal{S}})^\top$. It is then obvious that $Q_t(s,a)W_t(s,a)$ is the linear regression estimate of $P(s,a)$ using the data up to episode t. When a new input vector x_t comes, the algorithm checks whether $\|Q(s,a)x_t\|_2$ is below a predetermined threshold α_S (line 3). Recall that $Q(s,a)$ is the inverse covariance matrix, so a small $\|Q(s,a)x_t\|_2$ implies that the estimate $Q_t(s,a)W_t(s,a)$ is close to $P(s,a)$ along the direction of x_t, so it predicts $P_{x_t}(\cdot|s,a) = x_t^\top P(s,a) \approx x_t^\top Q(s,a)W(s,a)$; otherwise returns $\perp$. The KWIK subroutine for rewards is similar hence omitted. To ensure that the estimated transition probability is valid, the estimated vector is projected onto Δ_{S-1}, which can be done efficiently using existing

techniques (Duchi et al., 2008).

Below we state the KWIK bound for learning the transition function; the KWIK bound for learning rewards is much smaller hence omitted here. We use the KWIK bound for scalar linear regression from Walsh et al. (2009) and the property of multinomial samples to get our KWIK bound.

Theorem 3.4 (KWIKLR bound for learning multinomial vectors)**.** *For any $\epsilon > 0$ and $\delta > 0$, if the* KWIKLR *algorithm is executed for probability vectors $P_t(\cdot|s,a)$, with $\alpha_S = \min\left(b_1\frac{\epsilon^2}{d^{3/2}}, b_2\frac{\epsilon^2}{\sqrt{d}\log(d2^S/\delta)}, \frac{\epsilon}{2\sqrt{d}}\right)$ with suitable constants b_1 and b_2, then the number of $\perp$'s where updates take place (see line 7) will be bounded by $O\left(\frac{d^2}{\epsilon^4}\max\left(d^2, S^2\log^2\left(\frac{d}{\delta'}\right)\right)\right)$, and, with probability at least $1-\delta$, $\forall x_t$ where a non-"$\perp$" prediction is returned, $\left\|\widehat{P}_t^{x_t}(\cdot|s,a) - p_t^{x_t}(\cdot|s,a)\right\|_1 \leq \epsilon$.*

Proof. (See full proof in Section 3.8.1.3.) We provide a direct reduction to KWIK bound for learning scalar values. The key idea is to notice that for any vector $v \in \mathbb{R}^S$:

$$\|v\|_1 = \sup_{f\in\{-1,1\}^S} v^\top f.$$

So conceptually we can view Algorithm 3.3 as running 2^S scalar linear regression simultaneously, each of which projects the vector label to a scalar by a fixed linear transformation f. We require every scalar regressor to have $(\epsilon, \delta/2^S)$ KWIK guarantee, and the ℓ_1 error guarantee for the vector label follows from union bound. □

Algorithm 3.3 KWIK learning of $P_x(\cdot|s,a)$

function INITIALIZE(S, d, α_S)
  $Q(s,a) \leftarrow I_d$ for all (s,a)
  $W(s,a) \leftarrow \{0\}^{d\times S}$ for all (s,a)
function PREDICT(x, s, a)
  if $\|Q(s,a)x\|_1 \leq \alpha_S$ **then**
    return $\widehat{P}_x(\cdot|s,a) = x^\top Q(s,a)W(s,a)$
  else
    return $\widehat{P}_x(\cdot|s,a) = \perp$
function UPDATE(x, s, a, s_{next})
  if $\|Q(s,a)x\|_1 > \alpha_S$ ("$\perp$" prediction) **then**
    $Q(s,a) \leftarrow Q(s,a) - \frac{(Q(s,a)x)(Q(s,a)x)^\top}{1+x^\top Q(s,a)x}$
    $y \leftarrow (\{\mathbb{1}[s_{\text{next}} = s']\}_{\forall s'\in\mathcal{S}})^\top$
    $W(s,a) \leftarrow W(s,a) + xy$

With this result, we are ready to prove the formal PAC guarantee for KWIKLR-RMAX.

Theorem 3.5 (PAC bound for KWIKLR-RMAX)**.** *For any input values $0 < \epsilon, \delta \leq 1$ and a linear CMDP model with d number of base MDPs, with probability $1 - \delta$, the* KWIKLR-RMAX *algorithm, produces a sequence of policies $\{\pi_t\}$ which yield at most*

$$O\left(\frac{d^2H^9SA}{\epsilon^5}\log\frac{1}{\delta}\max\left(d^2, S^2\log^2\frac{dSA}{\delta}\right)\right)$$

non-ϵ-optimal episodes.

Proof. When the KWIK subroutine (Algorithm 3.3) makes non-"$\perp$" predictions $\widehat{P}_x(s, a, s')$, we require that

$$\left\|\widehat{P}_x(\cdot|s,a) - P_x(\cdot|s,a)\right\|_1 \leq \epsilon/8H^2.$$

After projection onto Δ_{S-1}, we have:

$$\left\|\Pi_{\Delta_{S-1}}(\widehat{P}_x(s,a)) - P_x(\cdot|s,a)\right\|_1 \leq 2\left\|\widehat{P}_x(\cdot|s,a) - P_x(\cdot|s,a)\right\|_1 \leq \epsilon/4H^2.$$

Further, the update to the matrices Q and W happen only when an unknown state action pair (s, a) is visited and the KWIK subroutine still predicts $\perp$ (line 6). The KWIK bound states that after a fixed number of updates to an unknown (s, a) pair, the parameters will always be known with desired accuracy. The number of updates m can be obtained by setting the desired accuracy in transitions to $\epsilon/8H^2$ and failure probability as δ/SA in Theorem 3.4:

$$m = O\left(\frac{d^2H^8}{\epsilon^4}\max\left(d^2, S^2\log^2\left(\frac{dSA}{\delta}\right)\right)\right)$$

We now use Lemma 3.2 where instead of updating counts for number of visits, we look at the number of updates for unknown (s, a) pairs. On applying a union bound over all state action pairs and using Lemma 3.2, it is easy to see that the sub-optimal episodes are bounded by $O\left(\frac{mSA}{\epsilon}\ln\frac{1}{\delta}\right)$ with probability at least $1 - \delta$. The bound in Theorem 3.5 is obtained by substituting the value of m. □

We see that for this contextual MDP, the linear structure helps us in avoiding the exponential dependence in context dimension d. The combined dependence on S and d is now $O(\max\{d^4S, d^2S^3\})$.

Computational complexity of KWIKLR-RMAX The KWIKLR-RMAX algorithm maintains SA-many matrices of size $O(d^2)$. At the start of each episode, the algorithm requires a vector-matrix multiplication for computing the *knownness* of each state-action pair, which takes a total

of $O(d^2SA)$ operations (line 3). For known state-action pairs, computing $\widehat{P}_x(\cdot|s,a)$ requires a projection step along with vector-matrix multiplications and takes $O(d^2SA + S^2A)$ operations (line 4)[3]. At the end of every episode, it performs rank-1 updates to the matrices Q and W for every *unknown* state-action pair, each taking $O(d^2)$ operations (line 8 and line 10). We, therefore, observe that the KWIKLR-RMAX algorithm takes $O(SA\max\{d^2, S\})$ running time per episode.

3.4 No-regret Exploration in Generalized Linear CMDPs

In Section 3.3, we studied the online learning problem in contextual MDPs under the mistake bound criteria. A key takeaway from the results from that section is that linearity as a structural assumption allows us to design statistically and computationally efficient algorithms for efficient exploration compared to the provably hard case of smoothly varying contextual mappings. In this section, we generalize the linear CMDP structure studied in Section 3.3.3 for the contextual MDP setting to allow a larger class of potentially non-linear maps. Using linear mappings further provides interpretability and explainability properties which are very valuable in our motivating settings. Hence, in this section, we study the class of generalized linear CMDPs defined below under the regret criteria.

3.4.1 Generalized linear models for CMDPs

In this section, we define our structural assumption of generalized linear models for CMDPs and formally state our assumptions. Specifically, we model the categorical output space ($p(\cdot|s,a)$) in a contextual MDP using generalized linear mappings. For each pair $s, a \in \mathcal{S} \times \mathcal{A}$, there exists a weight matrix $W_{sa} \in \mathcal{W} \subseteq \mathbb{R}^{S\times d}$ where $\mathcal{W}$ is a convex set. For any context $x_t \in \mathbb{R}^d$, the next state distribution for the pair is specified by a GLM:

$$P_t(\cdot|s,a) = \nabla\Phi(W_{sa}x_t) \tag{3.5}$$

where $\Phi(\cdot) : \mathbb{R}^S \to \mathbb{R}$ is the link function of the GLM[4]. We will assume that this link function is convex which is always the case for a canonical exponential family (Lauritzen, 1996). For rewards, we assume that each mean reward is given by a linear function[5] of the context: $R_t(s,a) \coloneqq \theta_{sa}^\top x_t$ where $\theta \in \Theta \subseteq \mathbb{R}^d$. In addition, we will make the following assumptions about the link function.

[3]The projection onto Δ_{S-1} takes $O(S)$ operations per state-action pair.
[4]We abuse the term GLM here as we don't necessarily consider a complementary exponential family model in (3.5)
[5]Similar results can be derived for GLM reward functions.

Assumption 3.1. *The function $\Phi(\cdot)$ is α-strongly convex and β-strongly smooth, that is:*

$$\Phi(v) \geq \Phi(u) + \langle \nabla\Phi(u), v-u\rangle + \tfrac{\alpha}{2}\|u-v\|_2^2 \tag{3.6}$$

$$\Phi(v) \leq \Phi(u) + \langle \nabla\Phi(u), v-u\rangle + \tfrac{\beta}{2}\|u-v\|_2^2 \tag{3.7}$$

We will see that this assumption is critical for constructing the confidence sets used in our algorithm. We make another assumption about the size of the weight matrices W_{sa}^* and contexts x_t:

Assumption 3.2. *For all episodes t, we have $\|x_t\|_2 \leq B_x$ and for all state-action pairs (s,a), $\left\|W_{sa}^{(i)}\right\|_2 \leq B_p$ and $\|\theta_{sa}\|_2 \leq B_r$. So, we have $\|Wx_t\|_\infty \leq B_p B_x$ for all $W \in \mathcal{W}$.*

The following two contextual MDP models are special cases of our setting:

Example 3.1 (Multinomial logit model, Agarwal (2013))**.** *Each next state is sampled from a categorical distribution with probabilities*[6]*:*

$$P_x(s_i|s,a) = \frac{\exp(W_{sa}^{(i)}x)}{\sum_{j=1}^{S}\exp(W_{sa}^{(j)}x)}$$

The link function for this case can be given as $\Phi(y) = \log\left(\sum_{i=1}^{S}\exp(y_i)\right)$ which can be shown to be strongly convex with $\alpha = \frac{1}{\exp(BR)S^2}$ and smooth with $\beta = 1$.

Example 3.2 (Linear combination of MDPs, Section 3.3.3)**.** *Each MDP is obtained by a linear combination of d base MDPs $\{(\mathcal{S}, \mathcal{A}, P^i, R^i, H)\}_{i=1}^d$. Here, $x_t \in \Delta_{d-1}$, and $P_t(\cdot|s,a) \coloneqq \sum_{i=1}^d x_t[i]P^i(\cdot|s,a)$. The link function for this can be shown to be:*

$$\Phi(y) = \tfrac{1}{2}\|y\|_2^2$$

which is strongly convex and smooth with parameters $\alpha = \beta = 1$. Moreover, W_{sa} here is the $S \times d$ matrix containing each next state distribution in a column. We have, $B_p \leq \sqrt{d}$, $\|W_{sa}\|_F \leq \sqrt{d}$ and $\|W_{sa}x_t\|_2 \leq 1$.

3.4.2 Online estimation of GLM parameters

In order to obtain a no-regret algorithm for our setting, we will follow the *optimism in the face of uncertainty* (OFU) approach, described in Section 2.2.1.2, which relies on the construction of confidence sets for MDP parameters at the beginning of each episode. We focus on deriving these

[6]Without loss of generality, we can set the last row $W_{sa}^{(S)}$ of the weight matrix to be 0 to avoid an overparameterized system.

confidence sets for the next state distributions for all state action pairs. We assume that the link function Φ and values α, B and R are known a priori. The confidence sets are constructed and used in the following manner in the OFU template for MDPs: at the beginning of each episode $t = 1, 2, \ldots, T$:

- For each (s, a), compute an estimate of transition distribution $\widehat{P}_t(\cdot|s, a)$ and mean reward $\widehat{R}_t(s, a)$ along with confidence sets $\mathcal{P}$ and $\mathcal{R}$ such that $P_t(\cdot|s, a) \in \mathcal{P}$ and $R_t(s, a) \in \mathcal{R}$ with high probability.

- Compute an optimistic policy π_t using the confidence sets and unroll a trajectory in M_t with π_t. Using observed transitions, update the estimates and confidence sets.

Therefore, in the GLM-CMDP setup, estimating transition distributions and reward functions is the same as estimating the underlying parameters W_{sa} and θ_{sa} for each pair (s, a). Likewise, any confidence set $\mathcal{W}_{sa}$ for W_{sa} (Θ_{sa} for θ_{sa}) can be translated into a confidence set of transition distributions.

In our final algorithm for GLM-CMDP, we will use the method from this section for estimating the next state distribution for each state-action pair. The reward parameter θ_{sa} and confidence set Θ_{sa} is estimated using the linear bandit estimator (Lattimore and Szepesvári (2020), Chap. 20). Here, we solely focus on the following online estimation problem without any reference to the CMDP setup. Specifically, given a link function Φ, the learner observes a sequence of contexts $x_t \in \mathcal{X}$ $(t = 1, 2, \ldots)$[7] and a sample y_t drawn from the distribution $p_t \equiv \nabla\Phi(W^* x_t)$ over a finite domain of size S. Here, we use W^* to denote the true parameter for the given GLM model. The learner's task is to compute an estimate W_t for W^* and a confidence set $\mathcal{W}_t$ after any such t samples. We frame this as an online optimization problem with the following loss sequence (based on the negative log-likelihood):

$$l_t(W; x_t, y_t) = \Phi(W x_t) - y_t^\top W x_t \tag{3.8}$$

where y_t is the one-hot representation of the observed sample in round t. This loss function preserves the strong convexity of Φ with respect to $W x_t$ and is a proper loss function (Agarwal, 2013):

$$\operatorname*{argmin}_W \mathbb{E}\left[l_t(W; x_t, y_t)|x_t\right] = W^* \tag{3.9}$$

Since our aim is computational and memory efficiency, we carefully follow the Online Newton Step (Hazan et al., 2007) based method proposed for $0/1$ rewards with logistic link function in

[7] We overload the notation t here to refer to the context index for online optimization, and should not be confused with the episode index.

Zhang et al. (2016). While deriving the confidence set in this extension to GLMs, we use properties of categorical vectors in various places in the analysis which eventually saves a factor of S. The online update scheme is shown in Algorithm 3.4. Interestingly, note that for tabular MDPs, where $d = \alpha = 1$ and $\Phi(y) = \frac{1}{2}\|y\|_2^2$, with $\eta = 1$, we would recover the empirical average distribution as the online estimate. Along with the estimate W_{t+1}, we can also construct a high probability

Algorithm 3.4 Online parameter estimation for GLMs

Input: Φ, α, η
Set $W_1 \leftarrow \mathbf{0}$, $Z_1 \leftarrow \lambda \mathbb{I}_d$
for $t = 1, 2, \ldots$ **do**
  Observe x_t and sample $y_t \sim p_t(\cdot)$
  Compute new estimate W_{t+1}:

$$\underset{W \in \mathcal{W}}{\operatorname{argmin}} \frac{\|W - W_t\|_{Z_{t+1}}^2}{2} + \eta \langle \nabla l_t(W_t x_t) x_t^\top, W - W_t \rangle \tag{3.10}$$

  where $Z_{t+1} = Z_t + \frac{\eta\alpha}{2} x_t x_t^\top$.

confidence set as follows:

Theorem 3.6 (Confidence set for W^*)**.** *In Algorithm 3.4, for all timesteps $t = 1, 2, \ldots$, with probability at least $1 - \delta$, we have:*

$$\|W_{t+1} - W^*\|_{Z_{t+1}} \leq \sqrt{\gamma_{t+1}} \tag{3.11}$$

where

$$\gamma_{t+1} = \lambda B^2 + 8\eta B_p B_x + 2\eta \left[(\tfrac{4}{\alpha} + \tfrac{8}{3} B_p B_x)\tau_t + \tfrac{4}{\alpha} \log \tfrac{\det(Z_{t+1})}{\det(Z_1)} \right] \tag{3.12}$$

with $\tau_t = \log(2\lceil 2 \log St \rceil t^2/\delta)$ and $B = \max_{W \in \mathcal{W}} \|W\|_F$.

Any upper bound for $\|W^*\|_F^2$ can be substituted for B the confidence width in (3.12). The term γ_t depends on the size of the true weight matrix, strong convexity parameter $\frac{1}{\alpha}$ and the log determinant of the covariance matrix. We will later show that the last term grows at a $O(d \log t)$ rate. Therefore, overall γ_t scales as $O(S + \frac{d}{\alpha} \log^2 t)$. The complete proof can be found in Section 3.8.2.1.

Algorithm 3.4 only stores the empirical covariance matrix and solves the optimization problem in (3.10) for the current context. Since $\mathcal{W}$ is convex, this is a tractable problem and can be solved via any off-the-shelf optimizer up to desired accuracy. The total computation time for each context and all (s, a) pairs is $O(\text{poly}(S, A, d))$ with no dependence on t. Furthermore, we only store SA-many matrices of size $S \times d$ and covariance matrices of sizes $d \times d$. Thus, both time and memory complexity of the method are bounded by $O(\text{poly}(S, A, H, d))$ per episode.

3.4.3 GLM-ORL: Optimistic exploration for GLM-CMDPs

In this section, we describe the OFU based online learning algorithm which leverages the confidence sets as described in the previous section. Not surprisingly, our algorithm is similar to the algorithm of Dann et al. (2019) and Abbasi-Yadkori and Neu (2014) and follows the standard format for no-regret bounds in MDPs. In all discussions about CMDPs, we will again use $x_t \in \mathcal{X}$ to denote the context for episode t and use Algorithm 3.4 from the previous section to estimate the corresponding MDP M_t. Specifically, for each state-action pair (s, a), we use all observed transitions to estimate W_{sa} and θ_{sa}. We compute and store the quantities used in Algorithm 3.4 for each (s, a): we use $\widehat{W}_{t,sa}$ to denote the parameter estimate for W_{sa} at the beginning of the t^{th} episode. Similarly, we use the notation $\gamma_{t,sa}$ and $Z_{t,sa}$ for the other terms. Using the estimate $\widehat{W}_{t,sa}$ and the confidence set, we compute the confidence interval for $P_t(\cdot|s, a)$:

$$\begin{aligned}\xi^{(p)}_{t,sa} \coloneqq \left\| P_t(\cdot|s,a) - \widehat{P}_t(\cdot|s,a) \right\|_1 &\leq \beta\sqrt{S} \left\| W_{sa} - \widehat{W}_{t,sa} \right\|_{Z_{t,sa}} \|x_t\|_{Z^{-1}_{t,sa}} \\ &\leq \beta\sqrt{S}\sqrt{\gamma_{t,sa}} \|x_t\|_{Z^{-1}_{t,sa}} \end{aligned} \tag{3.13}$$

where in the definition of $\gamma_{t,sa}$ we use $\delta = \delta_p$. It is again easy to see that for tabular MDPs with $d = 1$, we recover a similar confidence interval as used in Jaksch et al. (2010). For rewards, using the results from linear contextual bandit literature (Lattimore and Szepesvári (2020), Theorem 20.5), we use the following confidence interval:

$$\xi^{(r)}_{t,sa} \coloneqq \left| R_t(s,a) - \widehat{R}_t(s,a) \right| = \underbrace{\left(\sqrt{\lambda d} + \sqrt{\tfrac{1}{4} \log \tfrac{\det Z_{t,sa}}{\delta_r^2 \det \lambda I}} \right)}_{\coloneqq \zeta_{t,sa}} \|x_t\|_{Z^{-1}_{t,sa}} \tag{3.14}$$

In GLM-ORL, we use these confidence intervals to compute an optimistic policy (line 9 and line 15). The computed value function is optimistic as we add the total uncertainty as a bonus (line 11) during each Bellman backup. For any step h, we clip the optimistic estimate between $[0, H - h]$ during Bellman backups (line 13. After unrolling an episode using π_t, we update the parameter estimates and confidence sets for every observed (s, a) pair.

For any sequence of T contexts, we can guarantee the following regret bound:

Theorem 3.7 (Regret of GLM-ORL)**.** *For any $\delta \in (0, 1)$, if Algorithm 3.5 is run with the estimation method Algorithm 3.4, then for all $T \in \mathbb{N}$ and with probability at least $1 - \delta$, the regret* $\mathrm{Regret}(T)$ *is:*

$$\widetilde{O}\left(\left(\frac{\sqrt{d} \max_{s,a} \|W_{sa}\|_F}{\sqrt{\alpha}} + \frac{d}{\alpha} \right) \beta S H^2 \sqrt{AT} \log \frac{1}{\lambda\delta} \right)$$

If $\left\|W^{(i)}\right\|$ is bounded by B_p, we get $\|W_{sa}\|_F^2 \leq SB_p^2$, whereas, for the linear case (Example 3.2), $\|W_{sa}\|_F^2 \leq \sqrt{d}$. Substituting the bounds on $\|W_{sa}\|_F^2$, we get:

Corollary 3.8 (Multinomial logit model)**.** *For Example 3.1, we have* $\|W\|_F \leq B\sqrt{S}$*,* $\alpha = \frac{1}{\exp(BR)S^2}$ *and* $\beta = 1$*. Therefore, the regret bound of* GLM-ORL *is* $\widetilde{O}(dS^3H^2\sqrt{AT})$.

Corollary 3.9 (Regret bound for linear combination case)**.** *For Example 3.2, with* $\|W\|_F \leq \sqrt{d}$*, the regret bound of* GLM-ORL *is* $\widetilde{O}(dSH^2\sqrt{AT})$.

Algorithm 3.5 GLM-ORL (GLM Optimistic Reinforcement Learning)

Input: $\mathcal{S}, \mathcal{A}, H, \Phi, d, \mathcal{W}, \lambda, \delta$
$\delta' = \frac{\delta}{2SA+SH}$, $\widetilde{V}_{t,H+1}(s) = 0 \; \forall s \in \mathcal{S}, t \in \mathbb{N}$
for $t \leftarrow 1, 2, 3, \ldots$ **do**
  Observe current context x_t
  for $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
    $\widehat{P}_t(\cdot|s,a) \leftarrow \nabla\Phi(\widehat{W}_{t,sa}x_t)$
    $\widehat{R}_t(s,a) \leftarrow \left\langle \hat{\theta}_{t,sa}, x_t \right\rangle$
    Compute conf. intervals using eqns. (3.13), (3.14)
  for $h \leftarrow H, H-1, \cdots, 1$, and $s \in \mathcal{S}$ **do**
    for $a \in \mathcal{A}$ **do**
      $\varphi = \left\|\widetilde{V}_{t,h+1}\right\|_\infty \xi_{t,sa}^{(p)} + \xi_{t,sa}^{(r)}$
      $\widetilde{Q}_{t,h}(s,a) = \widehat{P}_{t,sa}^\top \widetilde{V}_{t,h+1} + \widehat{R}_t(s,a) + \varphi$
      $\widetilde{Q}_{t,h}(s,a) = 0 \vee \left(\widetilde{Q}_{t,h}(s,a) \wedge V_h^{\max}\right)$
    $\pi_{t,h}(s) = \operatorname{argmax}_a \widetilde{Q}_{t,h}(s,a)$
    $\widetilde{V}_{t,h}(s) = \widetilde{Q}_{t,h}(s, \pi_{t,h}(s))$
  Unroll a trajectory in M_t using π_t
  Update $\widehat{W}_{sa}$ and $\hat{\theta}_{sa}$ for observed samples.

In Corollary 3.9, the bound is worse by a factor of $\sqrt{H}$ when compared to the $\widetilde{O}(HS\sqrt{ATH})$ bound of UCRL2 for tabular MDPs ($d = 1$). This factor is incurred while bounding the sum of confidence widths in line 3.30 (in UCRL2 it is $O(\sqrt{SATH})$).

3.4.3.1 Mistake bound for GLM-ORL

The regret analysis shows that the total value loss suffered by the agent is sublinear in T, and therefore, goes to 0 on average. However, this can still lead to infinitely many episodes where the sub-optimality gap is larger than a desired threshold ϵ, given that it occurs relatively infrequently. It is still desirable, for practical purposes, to show a mistake bound result as discussed in Section 3.3. For GLM-ORL, we can show the following mistake bound:

Theorem 3.10 (Bound on the number of mistakes)**.** *For any number of episodes T, $\delta \in (0,1)$ and $\epsilon \in (0, H)$, with probability at least $1-\delta$, the number of episodes where* GLM-ORL*'s policy π_t is not ϵ-optimal is bounded by*

$$O\left(\frac{dS^2AH^5\log(TH)}{\epsilon^2}\left(\frac{d\log^2(TH)}{\alpha}+S\right)\right)$$

ignoring $O(poly(\log\log TH))$ terms.

We defer the proof to Section 3.8.2.4. Note that this term depends poly-logarithmically on T and therefore increases with time. The algorithm doesn't need to know the value of ϵ and result holds for all ϵ. This differs from the standard mistake bound style PAC guarantees where a finite upper bound is given. Dann et al. (2019) argued that this is due to the non-shrinking nature of the constructed confidence sets. In order to obtain a finite PAC mistake bound for our setting, a more involved data separation scheme needs to be used, as suggested by He et al. (2021) in their result for contextual linear bandits and linear mixture MDPs.

3.4.4 GLM-RLSVI: Randomized exploration in GLM-CMDPs

Empirical investigations in bandit and MDP literature has shown that optimism based exploration methods typically over-explore, often resulting in sub-optimal empirical performance. In contrast, Thompson sampling based methods which use randomization during exploration have been shown to have an empirical advantage with slightly worse regret guarantees. Recently, Russo (2019) showed that even with such randomized exploration methods, one can achieve a worst-case regret bound instead of the typical Bayesian regret guarantees. In this section, we show that the same is true for GLM-CMDP where a randomized reward bonus can be used for exploration. We build upon their work to propose an RLSVI style method (Algorithm 3.6) and analyze its expected regret. The main difference between Algorithm 3.5 and Algorithm 3.6 is that instead of the fixed bonus φ (line 11) in the former, GLM-RLSVI samples a random reward bonus in line 12 for each (s,a) from the distribution $N(0, HS\varphi^2)$. The variance term φ is set to a sufficiently high value, such that, the resulting policy is optimistic with constant probability. We use a slightly modified version of the confidence sets as follows:

$$\overline{\xi}^{(p)}_{t,sa} := 2 \wedge \left(\beta\sqrt{S}\sqrt{\gamma_{t,sa}}\|x_t\|_{Z^{-1}_{t,sa}}\right)$$
$$\overline{\xi}^{(r)}_{t,sa} := B_r R \wedge \left(\tau_{t,sa}\|x_t\|_{Z^{-1}_{t,sa}}\right)$$

The algorithm, thus, generates exploration policies by using perturbed rewards for planning. Similarly to Russo (2019), we can show the following bound for the expected regret incurred by

Algorithm 3.6 GLM-RLSVI

1: **Input:** $\mathcal{S}, \mathcal{A}, H, \Phi, d, \mathcal{W}, \lambda$
2: $\overline{V}_{t,H+1}(s) = 0 \; \forall s \in \mathcal{S}, t \in \mathbb{N}$
3: **for** $t \leftarrow 1, 2, 3, \ldots$ **do**
4: Observe current context x_t
5: **for** $s \in \mathcal{S}, a \in \mathcal{A}$ **do**
6: $\widehat{P}_t(\cdot|s,a) \leftarrow \nabla\Phi(\widehat{W}_{t,sa} x_t)$
7: $\widehat{R}_t(s,a) \leftarrow \left\langle \hat{\theta}_{t,sa}, x_t \right\rangle$
8: Compute conf. intervals using eqns. (3.13), (3.14)
9: **for** $h \leftarrow H, H-1, \cdots, 1$, and $s \in \mathcal{S}$ **do**
10: **for** $a \in \mathcal{A}$ **do**
11: $\varphi = (H-h)\overline{\xi}^{(p)}_{t,sa} + \overline{\xi}^{(r)}_{t,sa}$
12: Draw sample $b_{t,h}(s,a) \sim N(0, SH\varphi)$
13: $\overline{Q}_{t,h}(s,a) = \widehat{P}^{\top}_{t,sa}\overline{V}_{t,h+1} + \widehat{R}_t(s,a) + b_{t,h}(s,a)$
14: $\pi_{t,h}(s) = \operatorname{argmax}_a \overline{Q}_{t,h}(s,a)$
15: $\overline{V}_{t,h}(s) = \overline{Q}_{t,h}(s, \pi_{t,h}(s))$
16: Unroll a trajectory in M_t using π_t.
17: Update $\widehat{W}_{sa}$ and $\hat{\theta}_{sa}$ for observed samples.

GLM-RLSVI:

Theorem 3.11. *For any contextual MDP with given link function Φ, in Algorithm 3.6, if the MDP parameters for M_t are estimated using Algorithm 3.4, with reward bonuses $b_{t,h}(s,a) \sim N(0, SH\varphi_{t,h}(s,a))$ where $\varphi_{t,h}(s,a)$ is defined in line 11, the algorithm satisfies:*

$$\begin{aligned}\text{Regret}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} V_t^* - V_t^{\pi_t}\right] \\ &= \widetilde{O}\left(\left(\frac{\sqrt{d}\max_{s,a}\|W^*_{sa}\|_F}{\sqrt{\alpha}} + \frac{d}{\alpha}\right)\beta\sqrt{H^7S^3AT}\right)\end{aligned}$$

The proof of the regret bound is given in Section 3.8.2.3. Our regret bound is again worse by a factor of $\sqrt{H}$ when compared to the $\widetilde{O}(H^3S^{3/2}\sqrt{AT})$ bound from Russo (2019) for the tabular case. Therefore, such randomized bonus based exploration algorithms can also be used in the CMDP framework with similar regret guarantees as the tabular case.

3.4.5 Regret lower bound for GLM-CMDP

In this section, we show a regret lower bound by constructing a family of hard instances for the GLM-CMDP problem. We build upon the construction of Osband and Van Roy (2016) and Jaksch et al. (2010) for the analysis:

Theorem 3.12. *For any algorithm* ALG*, there exists CMDP's with S states, A actions, horizon H and $T \geq dSA$ for logit and linear combination case, such that the expected regret of* ALG *(for any sequence of initial states $\in \mathcal{S}^T$) after T episodes is:*

$$\mathbb{E}[R(T; \text{ALG}, M_{1:T}, s_{1:T})] = \Omega(H\sqrt{dSAT})$$

Proof. We start with the lower bound from Jaksch et al. (2010) adapted to the episodic setting.

Theorem 3.13 (Jaksch et al. (2010), Thm. 5)**.** *For any algorithm* ALG$'$*, there exists an MDP M with S states, A actions, and horizon H, such that for $T \geq dSA$, the expected regret of* ALG *after T episodes is:*

$$\mathbb{E}[R(T; \text{ALG}', s, M)] = \Omega(H\sqrt{SAT})$$

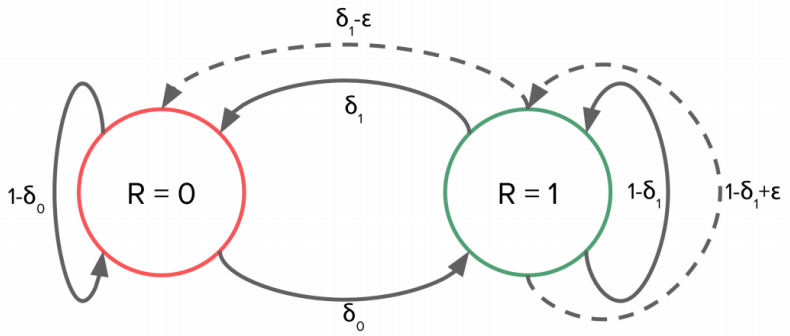


Figure 3.1: Hard 2-state MDP for CMDP regret lower bound. (Osband and Van Roy, 2016)

The lower bound construction is obtained by concatenating $\lceil S/2 \rceil$-copies of a bandit-like 2-state MDP as shown in Figure 3.1[8]. Essentially, state 1 is a rewarding state and all but one action take the agent to state 0 with probability δ_1. The remaining optimal action transits to state 0 with probability $\delta_1 - \epsilon$. This makes the construction similar to a hard Bernoulli multi-armed bandit instance which leads to the lower bound. Now, we will construct a set of such hard instances with the logit link function for transition probabilities. Since, the number of next states is 2, we use a GLM with parameter vector w^* of shape $1 \times d$. Thus, for any context x, the next state probabilities are given as:

$$P(1|1, a; x) = \frac{\exp(w_a^* x)}{1 + \exp(w_a^* x)} = \phi(w_a^* x)$$

[8]The two state MDP is built using $A/2$ actions with the rest used for concatenation. We ignore this as it only leads to a difference in constants.

If $w_a^* x = 0$, the value turns out to be $\frac{1}{2}$ which we choose as $\delta_1 - \epsilon$. For making the probability $\delta_1 = \frac{1}{2} + \epsilon$, we need to have $w_a^* x = \phi^{-1}(\delta_1) = c^*$. We consider the case where for each index i, all but one action has $w_a^*[i] = 0$ and one action a_i^* has $w_{a^*}^*[i] = c^*$. The sequence of contexts given to the algorithm comprises of T/d indicator vectors with 1 at only one index. Therefore, for each episode t, we get an MDP with $P_t(0|1, a^*_{t\%d}) = 1/2$ for one optimal action and $1/2$ for all other actions. Therefore, this is a hard instance as shown in Figure 3.1. The agent interacts with each such MDP $T_i \approx T/d$ times. Further, these MDPs are decoupled as the context vectors are non-overlapping. Therefore, we have:

$$\begin{aligned}\mathbb{E}[R(T; \textsc{Alg}, M_{1:T}, s_{1:T})] &= \sum_{i=1}^{d} \mathbb{E}[R(T_i; \textsc{Alg}, M_{1:T}, s_{1:T})] \\ &\geq \sum_{i=1}^{d} cH\sqrt{SAT/d} = cH\sqrt{dSAT}\end{aligned}$$

□

Linear combination case Similar to the logit case, we need to construct the sequence of hard instances in the linear combination case. It turns out that a similar construction works. Note that, in the linear combination case, each parameter vector w_a^* now directly contains the probability of moving to the rewarding state. In other words, each index of this vector $w_a^*[i]$ corresponds to the next state visitation probability for the base MDP M_i. Therefore, for each index, we again set one action's value to $\frac{1}{2} + \epsilon$ and all others to 0. This maintains the independence argument and using indicator vectors as contexts, we get the same sequence of MDPs. The same lower bound can therefore be obtained for the linear combination case.

The lower bound has the usual dependence on MDP parameters in the tabular MDP case, with an additional $O(\sqrt{d})$ dependence on the context dimension. Thus, our upper bounds have a gap of $O(H\sqrt{dS})$ with the lower bound even in the arguably simpler case of Example 3.2.

3.5 Related Work and Comparison

Transfer in RL with latent contexts The general definition of CMDPs captures the problem of transfer in RL and multi-task RL. See Taylor and Stone (2009) and Lazaric (2011) for surveys of empirical results. Recent papers have also advanced the theoretical understanding of transfer in RL. For instance, Brunskill and Li (2013) and Hallak et al. (2015) analyzed the sample complexity of CMDPs where each MDP is an element of a finite and small set of MDPs, and the MDP label is treated as the *latent* (i.e., unseen) context. Mahmud et al. (2013) consider the problem of transferring

the optimal policies of a large set of known MDPs to a new MDP. All of these recent papers assume that the MDP label (i.e., the context) is *not* observed. Hence, their methods have to initially explore in every new MDP to identify its label, which requires the episode length to be substantially longer than the planning horizon. This can be a problematic assumption in our motivating scenarios, where we interact with a patient / user / student for a limited period of time and the data in a single episode (whose length H *is* the planning horizon) is not enough for identifying the underlying MDP. In contrast to prior work, we propose to leverage observable context information to perform more direct transfer from previous MDPs, and our algorithm works with arbitrary episode length H.

RL in metric space For smooth CMDPs (Section 3.3.1), we pool observations across similar contexts and reduce the problem to learning policies for finitely many MDPs. An alternative approach is to consider an infinite MDP whose state representation is augmented by the context, and apply PAC-MDP methods for metric state spaces (e.g., C-PACE proposed by Pazis and Parr (2013)). However, doing so might increase the sample and computational complexity unnecessarily, because we no longer leverage the structure that a particular component of the (augmented) state, namely the context, remains the same in an episode. Concretely, the augmenting approach needs to perform planning in the augmented MDP over states and contexts, which makes its computational/storage requirement worse than our solution: we only perform planning in MDPs defined on $\mathcal{S}$, whose computational characteristics have no dependence on the context space. In addition, we allow the context sequence to be chosen in an adversarial manner. This corresponds to adversarially chosen initial states in MDPs, which is usually not handled by PAC-MDP methods.

Another line of recent work on RL in metric spaces, has analyzed the regret of model-free exploration algorithms. Song and Sun (2019) adapt the Q-learning with Bernstein bonus algorithm from Jin et al. (2018) with an ϵ-net of the state-action space, to show a regret guarantee of $O\left(H^{5/2}K^{\frac{d+1}{d+2}}\right)$. Improving upon this, Sinclair et al. (2019) show that adaptive discretization can be used in the Q-learning algorithm to only refine those partitions which the agent frequently visits, hence, saving on space and time complexity. Along the same lines, Cao and Krishnamurthy (2020) show a gap-dependent regret analysis and show a dependence on the zooming dimension instead of the covering number of the state-action space.

Contextual MDP To our knowledge, Hallak et al. (2015) first used the term contextual MDPs and studied the case when the context space is finite and the context is not observed during interaction. They propose CECE, a clustering based learning method and analyze its regret. In Section 3.3, we generalized the CMDP framework and proved the PAC exploration bounds under smoothness and linearity assumptions over the contextual mapping. The PAC bound is incomparable to the regret bounds in Section 3.4 as a no-regret algorithm can make arbitrarily many mistakes $\Delta_t \geq \epsilon$ as long

as it does so sufficiently less frequently.

Our work in Section 3.4 can be best compared with Abbasi-Yadkori and Neu (2014) and Dann et al. (2019) who propose regret minimizing methods for CMDPs. Abbasi-Yadkori and Neu (2014) consider an online learning scenario where the values $P_t(s'|s,a)$ are parameterized by a GLM. The authors give a no-regret algorithm which uses confidence sets based on Abbasi-Yadkori et al. (2012). However, their next state distributions are not normalized which leads to invalid next state distributions. Due to these modelling errors, their results cannot be directly compared with our analysis. Even if we ignore their modelling error, in the linear combination case, we get an $\widetilde{O}(S\sqrt{A})$ improvement. Similarly, Dann et al. (2019) proposed an OFU based method ORLC-SI for the linear combination case. Their regret bound is $\widetilde{O}(\sqrt{S})$ worse than our bound for GLM-ORL. In addition, the work also showed that obtaining a finite mistake bound guarantees for such CMDPs requires a non-trivial and novel confidence set construction. In this paper, we show that a $\mathrm{poly}(\log T)$ mistake bound can still be obtained. For a quick comparison, Table 3.1 shows the results from the two papers.

Algorithm	$\mathrm{Regret}^{\textbf{Linear}}(T)$	$\mathrm{Regret}^{\textbf{Logit}}(T)$	$\|P_x\|_1 = 1$
Alg 1 (Abbasi-Yadkori and Neu, 2014)	$\widetilde{O}(dH^3S^2A\sqrt{T})$	✗	✗
ORLC-SI (Dann et al., 2019)	$\widetilde{O}(dH^2S^{3/2}\sqrt{AT})$	✗	✗
GLM-ORL (this work)	$\widetilde{O}(dH^2S\sqrt{AT})$	$\widetilde{O}(dH^2S^3\sqrt{AT})$	✓

Table 3.1: Comparison of regret guarantees for CMDPs. Last column denotes whether the transition dynamics $P_x(\cdot|s,a)$ are normalized in the model or not.

3.6 Discussion

In this chapter, we formalized an RL setting where the agent interacts with a sequence of environments whose dynamics are parameterized by an observable context vector. The use of such side-information for multi-task or sequential interaction protocols have also been explored in the empirical RL literature (Lee et al., 2020; Benjamins et al., 2021; Klink et al., 2020a). However, the utility of such contextual information may not be directly visible and we discuss the a few insights below:

Contextual MDP vs context-augmented state In the contextual MDP setting that we consider, the context sequence breaks the overall interaction into a sequence of static (no variation in next-state dynamics and rewards) MDPs for each episode. Similarly, a general setting can be considered

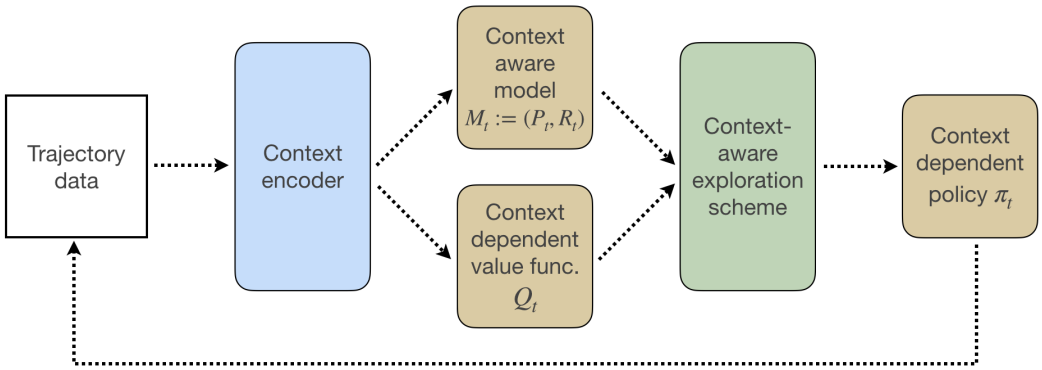


Figure 3.2: Generic context-dependent learning scheme.

where the context (representation of the current dynamics and reward function) evolves with time and can vary smoothly between and within episodes. Such contextual formalization can be quite helpful in building multi-task or continual learning settings as shown in Figure 3.2. In our setting, the context encoder is not needed as we assume that contexts are observed. In various settings, context encoders are learnt (Lee et al., 2020; Sodhani et al., 2021) and can also be framed under the meta-learning framework for quickly encoding the episode context using few-shot learning (Ritter et al., 2018). Further, the planning component essentially operates on a per-episode basis as the context stays constant over the fixed horizon episode and therefore, only computes the (optimistic) policy at the beginning of an episode.

The key factor here is to utilize the context representation to generalize across the tasks and utilize the specific heterogeneity across them by conditioning the model or value function estimates on the context. In our algorithms and analysis, we highlight another key component in contextual RL as described below.

Context dependent exploration In almost all context-dependent RL works (for instance, Lee et al. (2020); Benjamins et al. (2021); Klink et al. (2020a); Sodhani et al. (2021)), the context-dependent exploration component is absent. This follows the typical theory-vs-practice gap where sophisticated and provably efficient exploration schemes have been proposed for learning in MDPs but, in practice, simple tricks like epsilon-greedy exploration schemes are used due to computational concerns and mismatch in assumptions. In our work, we study the theoretical aspects of utilizing the context representation for adaptive exploration in an online learning setting. In the smooth CMDP case, we essentially show that maintaining separate counts for different clusters of contexts allows us to directly use known PAC-efficient algorithms. In practice, it would be interesting to see if the count-based exploration schemes can be extended to more general environments. On a more technical side, we show that if the agent has some prior knowledge about the mapping between the

context and MDP parameters, we can use the degree of uncertainty while learning that mapping to guide exploration and efficiently explore across heterogeneous environments. For empirically feasible algorithms, we can use well-known tricks like: learning context-dependent exploration noise (Klink et al., 2020b), ensemble based techniques for uncertainty estimation can be used. Our work provides a novel theoretical study of this framework and we hope that this can motivate further practical and sample efficient exploration schemes for generalization across tasks.

3.7 Summary

In this chapter, we presented a general setting involving the use of side information, or context, for learning near-optimal policies in a large and potentially infinite number of MDPs. We proposed PAC and regret efficient algorithms for contextual mappings which (1) vary smoothly with the context and (2) are (generalized) linear functions of the context. The results in this chapter also outline potential future directions: close the regret gap for tabular CMDPs, devise an efficient and sparsity aware regret bound when the underlying parameters follow a structured sparsity prior and investigate whether a near-optimal mistake and regret bound can be obtained simultaneously. Lastly, extension of the framework to non-tabular MDPs is an interesting problem for future work.

3.8 Proofs of Main Results

In this section, we provide the proofs of the main results provided in this chapter. In Section 3.8.1, we provide the formal proofs for the results in Section 3.3 and in Section 3.8.2, we provide a formal analysis for the results in Section 3.4.

3.8.1 Proofs of mistake bound results

3.8.1.1 Proofs of mistake bound of COVER-RMAX

In this section, we provide a formal analysis for normalized value functions $V^{\pi}_{M,h}(s) = \frac{1}{H}\sum_{h'=h}^{H-1}\mathbb{E}_{\pi}\left[r_{h'}\right]$. The results in Section 3.3 can be simply obtained by rescaling the error threshold for next state distributions and rewards by a factor of $1/H$ (see Lemma 3.14).

Proof of Lemma 3.2 We adapt the analysis in Kakade (2003) for the episodic case which results in the removal of a factor of H, since complete episodes are counted as mistakes and we do not count every sub-optimal action in an episode. The detailed analysis is reproduced here for completeness. For completing the proof of Lemma 3.2, firstly, we will look at a version of simulation lemma

from Kearns and Singh (2002). Also, for the complete analysis we will assume that the rewards lie between 0 and 1.

Definition 3.3 (Induced MDP)**.** *Let M be an MDP with $\mathcal{K} \subseteq \mathcal{S}$ being a subset of states. Given, such a set $\mathcal{K}$, we define an induced MDP $M_{\mathcal{K}}$ in the following manner. For each $s \in \mathcal{K}$, define the values*

$$P_{M_{\mathcal{K}}}(s'|s,a) = P_M(s'|s,a)$$
$$R_{M_{\mathcal{K}}}(s,a) = R_M(s,a)$$

For all $s \notin \mathcal{K}$, define $P_{M_{\mathcal{K}}}(s'|s,a) = \mathbb{1}\{s' = s\}$ and $R_{M_{\mathcal{K}}}(s,a) = 1$.

Lemma 3.14 (Simulation lemma for episodic MDPs)**.** *Let M and M' be two MDPs with the same state-action space. If the transition dynamics and the reward functions of the two MDPs are such that*

$$\|P_M(\cdot|s,a) - P_{M'}(\cdot|s,a)\|_1 \leq \epsilon_1 \ \forall s \in \mathcal{S}, a \in \mathcal{A}$$

$$|R_M(s,a) - R_{M'}(s,a)| \leq \epsilon_2 \ \forall s \in \mathcal{S}, a \in \mathcal{A}$$

then, for every (non-stationary) policy π the two MDPs satisfy this property:

$$|V_M^\pi - V_{M'}^\pi| \leq \epsilon_2 + H\epsilon_1$$

Proof. Consider $\mathcal{T}_h$ to be the set of all trajectories of length h and let $P_M^\pi(\tau)$ denote the probability of observing trajectory τ in M with the behaviour policy π. Further, let $U_M(\tau)$ the expected average

reward obtained for trajectory τ in MDP M.

$$
\begin{aligned}
|V_M^\pi - V_{M'}^\pi| &= \left| \sum_{\tau \in \mathcal{T}_H} (P_M^\pi(\tau) U_M(\tau) - P_{M'}^\pi(\tau) U_{M'}(\tau)) \right| \\
&\leq \left| \sum_{\tau \in \mathcal{T}_H} (P_M^\pi(\tau) U_M(\tau) - P_M^\pi(\tau) U_{M'}(\tau) + P_M^\pi(\tau) U_{M'}(\tau) - P_{M'}^\pi(\tau) U_{M'}(\tau)) \right| \\
&\leq \left| \sum_{\tau \in \mathcal{T}_H} (P_M^\pi(\tau)(U_M(\tau) - U_{M'}(\tau))) \right| + \left| \sum_{\tau \in \mathcal{T}_H} (U_{M'}(\tau)(P_M^\pi(\tau) - P_{M'}^\pi(\tau))) \right| \\
&\leq \left| \sum_{\tau \in \mathcal{T}_H} P_M^\pi(\tau) \right| \epsilon_2 + \left| \sum_{\tau \in \mathcal{T}_H} (P_M^\pi(\tau) - P_{M'}^\pi(\tau)) \right| \\
&\leq \epsilon_2 + \left| \sum_{\tau \in \mathcal{T}_H} [P_M^\pi(\tau) - P_{M'}^\pi(\tau)] \right|
\end{aligned}
$$

The bound for the second term follows from the proof of Lemma 8.5.4 in Kakade (2003). Combining the two expressions, we get the desired result. □

Lemma 3.15 (Induced inequalities)**.** *Let M be an MDP with $\mathcal{K}$ being the set of known states. Let $M_\mathcal{K}$ be the induced MDP as defined in Definition 3.3 with respect to $\mathcal{K}$ and M. We will show that for any (non-stationary) policy π, all states $s \in \mathcal{S}$,*

$$V_{M_\mathcal{K}}^\pi(s) \geq V_M^\pi(s)$$

and

$$V_M^\pi(s) \geq V_{M_\mathcal{K}}^\pi(s) - P_M^\pi[\textit{Escape to an unknown state} | s_0 = s]$$

where $V_M^\pi(s)$ denotes the value of policy π in MDP M when starting from state s.

Proof. See Lemma 8.4.4 from Kakade (2003). □

Corollary 3.16 (Implicit Explore and Exploit)**.** *Let M be an MDP with $\mathcal{K}$ as the set of known states and $M_\mathcal{K}$ be the induced MDP. If $\pi_{M_\mathcal{K}}^*$ and π_M^* be the optimal policies for $M_\mathcal{K}$ and M respectively, we have for all states s:*

$$V_M^{\pi_{M_\mathcal{K}}^*}(s) \geq V_M^*(s) - P_M^\pi[\textit{Escape to an unknown state} | s_0 = s]$$

Proof. Follows from Lemma 8.4.5 from Kakade (2003). □

Proof. **of Lemma 3.2** Let π_M^* be the optimal policy for M. Also, using the assumption about m,

we have an $\epsilon/2$-approximation of $M_{\mathcal{K}}$ as the MDP $\widehat{M}_{\mathcal{K}}$. RMAX computes the optimal policy for $\hat{M}_{\mathcal{K}}$ which is denoted by $\hat{\pi}$. Then, by Lemma 3.14,

$$\begin{aligned} V^{\hat{\pi}}_{M_{\mathcal{K}}}(s) &\geq V^{\hat{\pi}}_{\widehat{M}_{\mathcal{K}}}(s) - \epsilon/2 \\ &\geq V^{\pi^*_M}_{\widehat{M}_{\mathcal{K}}}(s) - \epsilon/2 \\ &\geq V^{\pi^*_M}_{M_{\mathcal{K}}}(s) - \epsilon \end{aligned}$$

Combining this with Lemma 3.15, we get

$$\begin{aligned} V^{\hat{\pi}}_{M}(s) &\geq V^{\hat{\pi}}_{M_{\mathcal{K}}}(s) - \mathrm{P}^{\pi}_{M}[\text{Escape to an unknown state}|s_0 = s] \\ &\geq V^{\pi^*_M}_{M_{\mathcal{K}}}(s) - \epsilon - \mathrm{P}^{\pi}_{M}[\text{Escape to an unknown state}|s_0 = s] \\ &\geq V^{*}_{M}(s) - \epsilon - \mathrm{P}^{\pi}_{M}[\text{Escape to an unknown state}|s_0 = s] \end{aligned}$$

If this escape probability is less than ϵ, then the desired relation is true. Therefore, we need to bound the number of episodes where this expected number is greater than ϵ. Note that, due to balanced wandering, there can be at most mSA visits to unknown states for the RMAX algorithm. In the execution, the agent may encounter an extra $H-1$ visits as the estimates are updated only after the termination of an episode.

Whenever this quantity is more than ϵ, the expected number of exploration steps in mSA/ϵ such episodes is at least mSA. By the Hoeffding's inequality, for N episodes, with probability, at least $1-\delta$, the number of successful exploration steps is greater than

$$N\epsilon - \sqrt{\frac{N}{2}\ln\frac{1}{\delta}}$$

Therefore, if $N = O(\frac{mSA}{\epsilon}\ln\frac{1}{\delta})$, with probability at least $1-\delta$, the total number of visits to an unknown state is more than mSA. Using the upper bound on such visits, we conclude that these many episodes suffice. □

Proof of Theorem 3.1 We now need to compute the required resolution of the cover and the number of transitions m which will guarantee the approximation for the value functions as required in the previous lemma. The following is a key result:

Lemma 3.17 (Cover approximation)**.** *For a given CMDP and a finite cover, i.e., $\mathcal{X} = \cup_{i=1}^{N(\mathcal{X},r)}\mathcal{B}_i$ such that $\forall i,\ \forall x_1, x_2 \in \mathcal{B}_i$:*

$$\|P_{x_1}(\cdot|s,a) - P_{x_2}(\cdot|s,a)\|_1 \leq \epsilon/8H$$

and

$$|R_{x_1}(s,a) - R_{x_2}(s,a)| \leq \epsilon/8$$

if the agent visit every state-action pair $m = \frac{128(S\ln 2 + \ln\frac{SA}{\delta})H^2}{\epsilon^2}$ *times in a ball* $\mathcal{B}_i$ *summing observations over all* $x \in \mathcal{B}_i$*, then, for any policy* π *and with probability at least* $1 - 2\delta$*, the approximate MDP* $\widehat{M}_i$ *corresponding to* $\mathcal{B}_i$ *computed using empirical averages will satisfy*

$$|V^{\pi}_{M_c} - V^{\pi}_{\widehat{M}_i}| \leq \epsilon/2$$

for all $x \in \mathcal{B}_i$*.*

Proof. With each visit to a state action pair (s,a), a transition to some $s' \in \mathcal{S}$ is observed for context $x_t \in \mathcal{B}_i$ in t_{th} visit with probability $P_{x_t}(s,a)$. Let us encode this by an S-dimensional vector I_t with 0 at all indices except s'. After observing m such transitions, the next state distribution for any $x \in \mathcal{B}_i$ is esitmated as $P_{\widehat{M}_i}(\cdot|s,a) = \frac{1}{m}\sum_{t=1}^{m} I_t$. Now for all $x \in \mathcal{B}_i$,

$$\left\|P_{\widehat{M}_i}(\cdot|s,a) - P_x(\cdot|s,a)\right\|_1 \leq \left\|P_{\widehat{M}_i}(\cdot|s,a) - \frac{1}{m}\sum_{t=1}^{m} P_{x_t}(\cdot|s,a)\right\|_1 + \epsilon/8H$$

For bounding the first term, we use the Hoeffding's bound:

$$\begin{aligned}
&P\left[\left\|P_{\widehat{M}_i}(\cdot|s,a) - \frac{1}{m}\sum_{t=1}^{m} P_{x_t}(\cdot|s,a)\right\|_1 \geq \epsilon\right] \\
&= P\left[\max_{s'\in A\subseteq\mathcal{S}}(P_{\widehat{M}_i}(s'\in A|s,a) - \frac{1}{m}\sum_{t=1}^{m} P_{x_t}(s'\in A|s,a)) \geq \epsilon/2\right] \\
&\leq \sum_{s'\in A\subseteq\mathcal{S}} P\left[(P_{\widehat{M}_i}(s'\in A|s,a) - \frac{1}{m}\sum_{t=1}^{m} P_{x_t}(s'\in A|s,a)) \geq \epsilon/2\right] \\
&\leq (2^S - 2)\exp(-m\epsilon^2/2)
\end{aligned}$$

Therefore, with probability at least $1 - \delta/2$, for all $s \in \mathcal{S}, a \in \mathcal{A}$, we have:

$$\left\|P_{\widehat{M}_i}(\cdot|s,a) - P_x(\cdot|s,a)\right\|_1 \leq \sqrt{\frac{2(S\ln 2 + \ln 2SA/\delta)}{m}} + \epsilon/8H$$

If $m = \frac{128(S\ln 2 + \ln\frac{SA}{\delta})H^2}{\epsilon^2}$, the error becomes $\epsilon/4H$. One can easily verify using similar arguments that, the error in rewards for any context $x \in \mathcal{B}_i$ is less than $\epsilon/4$.

By using the simulation Lemma 3.14, we get the desired result. □

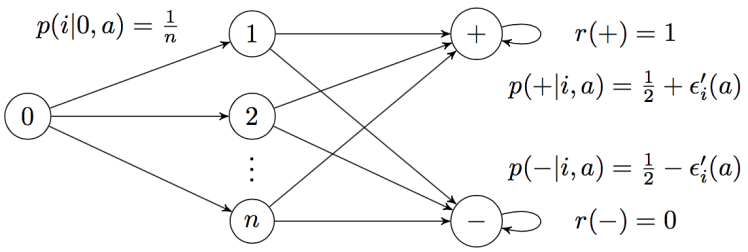


Figure 3.3: Hard instance for mistake lower bound for smooth CMDPs. ((Dann and Brunskill, 2015)) The initial state 0 moves to a uniform distribution over states 1 to n regardless of the action, and states $+/-$ are absorbing with 1 and 0 rewards respectively. States 0 to n have 0 reward for all actions. Each state $i \in [n]$ essentially acts as a hard bandit instance, whose A actions move to $+$ and $-$ randomly. Action a_0 satisfies $p(+|i, a_0) = \frac{1}{2} + \frac{\epsilon'}{2}$ and there is at most one other action a_i with $p(+|i, a_i) = \frac{1}{2} + \epsilon'$. Any other action a_j satisfies $p(+|i, a_j) = \frac{1}{2}$.

3.8.1.2 Lower bound analysis

The key construction in the proof is to embed multiple MDP learning problems in a CMDP, such that the agent has to learn the optimal policy in each MDP separately. We start with the lower bound for learning in episodic MDPs. See Figure 3.3 and its caption for details. The construction is due to Dann and Brunskill (2015) and we adapt their lower bound statement to our setting in Theorem 3.18.

Theorem 3.18 (Lower bound for episodic MDP (Dann and Brunskill, 2015))**.** *There exists constants δ_0, ϵ_0, such that for every $\delta \in (0, \delta_0)$ and $\epsilon \in (0, \epsilon_0)$, any algorithm that satisfies a PAC guarantee for (ϵ, δ) and computes a sequence of deterministic policies, there is a hard instance M_{hard} so that $\mathbb{E}[B] = \Omega\left(\frac{H^2SA}{\epsilon^2}\right)$, where B is the number of sub-optimal episodes. The constants can be chosen as $\delta_0 = \frac{e^{-4}}{80}$, $\epsilon_0 = \frac{H-2}{640e^4}$.*

Now we discuss how to populate the context space with these hard MDPs. Note in Figure 3.3 that, the agent does not know which action is the most rewarding (a_i), and the adversary can choose i to be any element of $[A]$ (which is essentially choosing an instance from a family of MDPs). In our scenario, we would like to allow the adversary to choose the MDP *independently* for each individual packing point to yield a lower bound linear in the packing number. However, this is not always possible due to the smoothness assumption, as committing to an MDP at one point may restrict the adversary's choices at another point.

To deal with this difficulty, we note that any pair of hard MDPs differ from each other by $O(\epsilon')$ in transition distributions. Therefore, we construct a packing of $\mathcal{X}$ with radius $r = 8\epsilon'$, defined as a set of points Z such that any two points in Z are at least r away from each other. The maximum

size of such Z is known as the *packing number*:

$$\mathcal{D}(\mathcal{X}, r) = \max\{|Z| : Z \text{ is an } r\text{-packing of } \mathcal{X}\},$$

which is related to the covering number as $N(\mathcal{X}, r) \leq \mathcal{D}(\mathcal{X}, r)$. The radius r is chosen to be $O(\epsilon')$ so that arbitrary choices of hard MDP instances at different packing points always satisfy the smoothness assumption (recall that $L_p = 1$). Once we fix the MDPs for all $x \in Z$, the MDP for $x \in \mathcal{X} \setminus Z$ is specified as follows: for state i and action a,

$$P_x(+|i, a) = \max_{x' \in Z} \max(1/2, P_{x'}(+|i, a) - \text{dis}(x, x')/2).$$

Essentially, as we move away from a packing point, the transition to $+/-$ become more uniform. We can show that:

Claim 3.19. *The CMDP defined above is satisfies Definition 3.2 with constant $L_p = 1$.*[9]

Proof. We need to prove that the defined contextual MDP, satisfies the constraints in Definition 3.2. Let us assume that the smoothness assumption is violated for a context pair (x_1, x_2). The smoothness constraints for rewards are satisfied trivially for any value of L_r as they are constant. This implies that there exists state $i \in [n]$ and action a such that

$$\begin{aligned} &\|P_{x_1}(\cdot|i, a) - P_{x_2}(\cdot|i, a)\|_1 > \text{dis}(x_1, x_2) \\ \Rightarrow\ &2|P_{x_1}(+|i, a) - P_{x_2}(+|i, a)| > \text{dis}(x_1, x_2) \end{aligned}$$

We know that, $P_c(+|i, a) \in [1/2, 1/2 + \epsilon']$, which shows that $\text{dis}(x_1, x_2) < 2\epsilon'$. Without loss of generality, assume $P_{x_1}(+|i, a) > P_{x_2}(+|i, a)$ which also leads to

$$\begin{aligned} &P_{x_1}(+|i, a) > 1/2 \\ \Rightarrow\ &\exists x_0 \in Z \text{ such that } \text{dis}(x_1, x_0) < 2\epsilon' \end{aligned}$$

By triangle inequality, we have

$$\text{dis}(x_2, x_0) < 4\epsilon'$$

[9]The reward function does not vary with context, hence, reward smoothness is satisfied for all $L_r \geq 0$.

Now, $\forall x_0' \in Z$, such that $\mathrm{dis}(x_0', x_0) \geq 8\epsilon'$, by triangle inequality, we have

$$\mathrm{dis}(x_0', x_1) > 6\epsilon'$$
$$\mathrm{dis}(x_0', x_2) > 4\epsilon'$$

This simplifies the definition of $P_c(+|i,a)$ for $c = x_1, x_2$ to

$$P_c(+|i,a) = \max(1/2, P_{x_0}(+|i,a) - \mathrm{dis}(x_0, c)/2)$$

Now,

$$\begin{aligned}
|P_{x_1}(+|i,a) - P_{x_2}(+|i,a)| &= P_{x_1}(+|i,a) - P_{x_2}(+|i,a) \\
&= P_{x_0}(+|i,a) - \mathrm{dis}(x_0, x_1)/2 \\
&\quad - \max\left(1/2, P_{x_0}(+|i,a) - \mathrm{dis}(x_0, x_2)/2\right) \\
&\leq P_{x_0}(+|i,a) - \mathrm{dis}(x_0, x_1)/2 - (P_{x_0}(+|i,a) - \mathrm{dis}(x_0, x_2)/2) \\
&= \frac{1}{2}\left(\mathrm{dis}(x_0, x_2) - \mathrm{dis}(x_0, x_1)\right) \leq \mathrm{dis}(x_1, x_2)/2
\end{aligned}$$

which leads to a contradiction. □

We choose the context sequence given as input to be repetitions of an arbitrary permutation of Z. Therefore, our construction populates a set of packing points in the context space with hard MDPs. We claim that these instances are independent of each other from the algorithm's perspective. To formalize this statement, let Z be the $8\epsilon'$-packing as before. The adversary makes the choices of the instances at each context $x \in Z$, as follows: Select an MDP from the family of hard instances described in Figure 3.3 where the optimal action from each state in $[n]_+$ is chosen randomly and independently from the other assignments. The parameter ϵ' deciding the difference in optimality of actions in Figure 3.3 is taken as $\frac{160\epsilon e^4}{(H-2)}$. The expression is obtained by using the construction of Theorem 3.18.

We denote these instances by the set $\mathcal{I}$ and an individual instance by $\mathcal{I}_z$. Let $z = \left\{z_1, z_2, \ldots, z_{|Z|}\right\}$ be the random vector denoting the optimal actions chosen for the MDPs corresponding to the packing points. By construction, we have a uniform distribution $\Gamma \equiv \Gamma_1 \times \Gamma_2 \times \ldots \Gamma_{|Z|}$ over these possible set of instances $\mathcal{I}_z$. From Claim 3.19, any assignment of optimal actions to these packing points would define a valid smooth contextual MDP. Further, the independent choice of the optimal actions makes MDPs at each packing point at least as difficult as learning a single MDP. Formally, let the sequence of transitions and rewards observed by the learning agent for all packing points be $T \equiv \left\{\tau_1, \tau_2, \ldots, \tau_{|Z|}\right\}$. Due to the independence between

individual instances, we can see that:

$$P_\Gamma[\,\tau_1|\tau_2,\tau_3,\ldots,\tau_{|Z|}\,] \equiv P_{\Gamma_1}[\,\tau_1\,]$$

where $P_\Gamma[\tau_i]$ denotes the distribution of trajectories τ_i. Thus, we cannot deduce anything about the optimal actions for one point by observing trajectories from MDP instances at other packing points. With respect to this distribution, learning in the contextual MDP is equivalent to or worse than simulating a PAC algorithm for a single MDP at each of these packing points. For any given contextual MDP algorithm ALG, we have:

$$\mathbb{E}_\Gamma[B_i] = \mathbb{E}_{\Gamma_{-i}}\left[\mathbb{E}_{\Gamma_i,\text{ALG}}\left[B_i|T_{-i}\right]\right] \geq \mathbb{E}_{\Gamma_{-i}}\left[\mathbb{E}_{\Gamma_1,\text{ALG}^*}[B_i]\right]$$

where ALG* is an optimal single MDP learning algorithm. The expectation is with respect to the distribution over the instances $\mathcal{I}_z$ and the algorithm's randomness. From Theorem 3.18, we can lower bound the expectation on the right hand side of the inequality by $\Omega\left(\frac{H^2SA}{\epsilon^2}\right)$.

The total number of mistakes is lower bounded as:

$$\mathbb{E}_\Gamma\left[\sum_{i=1}^{|Z|} B_i\right] = \sum_{i=1}^{|Z|}\mathbb{E}_\Gamma[B_i] \geq \Omega\left(\frac{|Z|H^2SA}{\epsilon^2}\right)$$

Setting $|Z| = \mathcal{D}(\mathcal{X},\epsilon_1) \leq \mathcal{N}(\mathcal{X},\epsilon_1)$ gives the stated lower bound with $\epsilon_1 = 8\epsilon'$.

3.8.1.3 Proof of mistake bound in Theorem 3.4

In this section, we present the proof of our KWIK bound for learning transition probabilities. Our proof uses a reduction technique that reduces the vector-valued label setting to the scalar setting, and combines the KWIK bound for scalar labels given by Walsh et al. (2009).

Proof. Fix a state action pair (s,a). Consider a sequence of contexts $x_1, x_2, ...$ for which the transitions were observed for pair (s,a). Given a new context x, we want to estimate:

$$P_x(\cdot|s,a) = x^\top P(s,a)$$

In our KWIKLR algorithm, this is estimated as:

$$\hat{P}_x(\cdot|s,a) = x^\top Q(s,a)W(s,a)$$

where $Q(s,a)$ and $W(s,a)$ are as described in Section 3.3.3.1.

We wish to bound the ℓ_1 error between $\hat{P}_x(\cdot|s,a)$ and $P_x(\cdot|s,a)$ for all x for which a prediction is made. We know that

$$\left\| P_x(\cdot|s,a) - \hat{P}_x(\cdot|s,a) \right\|_1 = \sup_{f \in \{-1,1\}^S} (P_x(\cdot|s,a) - \hat{P}_x(\cdot|s,a))f. \tag{3.15}$$

This representation of ℓ_1-norm can be used to prove a tighter KWIK bound for learning transition probabilities. For every fixed $f \in \{-1,1\}^S$, we formulate a new linear regression problem with feature-label pair:

$$(x, yf).$$

Recall that $y = (\{\mathbb{1}[s_{\text{next}} = s']\}_{\forall s' \in \mathcal{S}})^\top$ is the vector label of real interest, and f projects y to a scalar value. Algorithm 3.3 can be viewed as implicitly running this regression thanks to linearity: since Q only depends on input contexts and W is linear in y, $\hat{P}_x(\cdot|s,a)f$ is simply equal to the linear regression prediction for the problem (x, yf). As a result, the KWIK bound for the problem (x, yf) (which we establish below) automatically applies as a property of $\hat{P}_x(\cdot|s,a)f$. Taking union bound over all $f \in \{-1,1\}^S$ yields the desired ℓ_1 error guarantee for $\hat{P}_x(\cdot|s,a)$ thanks to (3.15).

Now we establish the KWIK guarantee for the new regression problem. The groundtruth (expected) label is

$$P_x(\cdot|s,a)f = x^\top (P(s,a)f) := x^\top \theta^f. \tag{3.16}$$

The noise in the label is then

$$\eta^f := (y - P_x(\cdot|s,a))f. \tag{3.17}$$

This noise has zero-mean and constant magnitude: $|\eta^f| \leq \|y - P_x(\cdot|s,a)\|_1 \|f\|_\infty \leq 2$.

With the above conditions, we can invoke the KWIK bound for scalar linear regression from Walsh et al. (2009):

Theorem 3.20 (KWIK bound for linear regression (Walsh et al., 2009))**.** *Suppose the observation noise in a noisy linear regression problem has zero-mean and its absolute value is bounded by β. Let M be an upper bound on the ℓ_2 norm of the true linear coefficients. For any $\delta' > 0$ and $\epsilon > 0$, if the* KWIK *linear regression algorithm is executed with $\alpha_0 = \min\left(b_1 \frac{\epsilon^2}{dM}, b_2 \frac{\epsilon^2}{M \log(d/\delta')}, \frac{\epsilon}{2M}\right)$, with suitable constants b_1 and b_2, then the number of $\perp$'s will be $O\left(M^2 \max\left(\frac{d^3}{\epsilon^4}, \frac{d \log^2(d/\delta')}{\epsilon^4}\right)\right)$, and with probability at least $1 - \delta'$, for each sample x_t for which a prediction is made, the prediction is ϵ-accurate.*

For our purpose, $\beta = 2$ as $|\eta^f| \leq 2$ and $M = \sqrt{d}$ as $\left\|\theta^f\right\|_2 = \|P(s,a)f\|_2 \leq \sqrt{d}$.

Now set $\delta' = \frac{\delta}{2^S}$ in Theorem 3.20. In the KWIK linear regression algorithm, the *known* status for a context c is checked in the same manner as done in line 3 in Algorithm 3.3. Therefore

$$\begin{aligned}
\mathbb{P}\left[\left\|P_x(\cdot|s,a) - \hat{P}_x(\cdot|s,a)\right\|_1 \geq \epsilon\right] &= \mathbb{P}\left[\sup_{f\in\{-1,1\}^S} (P_x(\cdot|s,a) - \hat{P}_x(\cdot|s,a))f \geq \epsilon\right] \\
&\qquad \text{(Equation (3.15))} \\
&\leq \sum_{f\in\{-1,1\}^S} \mathbb{P}\left[(P_x(\cdot|s,a) - \hat{P}_x(\cdot|s,a))f \geq \epsilon\right] \\
&\qquad \text{(union bound)} \\
&= \sum_{f\in\{-1,1\}^S} \mathbb{P}\left[x^\top\theta^f - x^\top Q(s,a)(W(s,a)f) \geq \epsilon\right] \\
&\qquad \text{(regression w.r.t. } f \text{ implicitly run)} \\
&\leq \sum_{f\in\{-1,1\}^S} \delta/2^S = \delta.
\end{aligned}$$

Substituting the values of M and δ' in Theorem 3.20, we get:

$$\alpha_S = \min\left\{b_1\frac{\epsilon^2}{d^{3/2}}, b_2\frac{\epsilon^2}{\sqrt{d}\log(d/2^S\delta')}, \frac{\epsilon}{2\sqrt{d}}\right\}$$

and number of $\perp$'s is bounded as

$$O\left(\max\left\{\frac{d^4}{\epsilon^4}, \frac{d^2S^2\log^2(d/\delta')}{\epsilon^4}\right\}\right).$$

□

3.8.2 Proofs of main regret bounds

3.8.2.1 Proof of estimation error bounds in Theorem 3.6

We closely follow the analysis from Zhang et al. (2016) and use properties of the categorical output space to adapt it to our case. The analysis is fairly similar, but carefully manipulating the matrix norms saves a factor of $O(S)$ in the confidence widths. For notation, we use $\nabla l_t(W_t)$ to refer to the derivative with respect to the matrix for loss l_t and $\nabla l_t(W_t x_t)$ for the derivative with respect to the projection $W_t x_t$. B_p denotes the upper bound on the ℓ_2-norm of each row $W^{(i)}$ and B_x is the assumed bound on the context norm $\|x\|_2$. Recall that, we are overloading the notation for t here to denote the sample timestep in the online optimization setup described in Section 3.4.2. Thus, it

should not be confused with the episode index in the chapter. Now, using the strong convexity of the loss function l_t with respect to $W_t x_t$, for all t, we have:

$$l_t(W_t) - l_t(W^*) \leq \langle \nabla l_t(W_t x_t), W_t x_t - W^* x_t \rangle - \tfrac{\alpha}{2} \underbrace{\|W^* x_t - W_t x_t\|_2^2}_{:= b_t}$$

Taking expectation with respect to the categorical sample y_t, we get:

$$\begin{aligned} 0 \leq \mathbb{E}_{y_t}[l_t(W_t) - l_t(W^*)] &\leq \mathbb{E}_{y_t}[\langle \nabla l_t(W_t x_t), W_t x_t - W^* x_t \rangle] - \tfrac{\alpha}{2} b_t \\ &\leq \mathbb{E}_{y_t}[\langle \nabla l_t(W_t x_t), W_t x_t - W^* x_t \rangle] - \tfrac{\alpha}{2} b_t \end{aligned} \tag{3.18}$$

where the `lhs` is obtained by using the calibration property from (3.9). Now, for the first term on `rhs`, we have:

$$\begin{aligned} \mathbb{E}_{y_t}[\langle \nabla l_t(W_t x_t), W_t x_t - W^* x_t \rangle] &= \mathbb{E}_{y_t}[\langle \nabla \Phi(W_t x_t) - y_t, W_t x_t - W^* x_t \rangle] \\ &= (\tilde{p}_t - p_t)^\top (W_t - W^*) x_t \\ &= \underbrace{(\tilde{p}_t - y_t)^\top (W_t - W^*) x_t}_{:= \mathbf{I}} + \underbrace{(y_t - p_t)^\top (W_t - W^*) x_t}_{:= c_t} \end{aligned} \tag{3.19}$$

where $\tilde{p}_t = \nabla \Phi(W_t x_t)$ and $\mathbb{E}[y_t] = p_t = \nabla \Phi(W^* x_t)$. We bound the term $\mathbf{I}$ using the following lemma:

Lemma 3.21.

$$\langle \nabla l_t(W_t x_t), W_t x_t - W^* x_t \rangle \leq \frac{\|W_t - W^*\|_{Z_{t+1}}}{2\eta} - \frac{\|W_{t+1} - W^*\|_{Z_{t+1}}}{2\eta} + 2\eta \|x_t\|_{Z_{t+1}^{-1}}^2 \tag{3.20}$$

Proof. To prove this, we go back to the update rule in (3.10) which has the following form:

$$Y = \operatorname*{argmin}_{W \in \mathcal{W}} \frac{\|W - X\|_M^2}{2} + \eta a^\top W b$$

with $Y = W_{t+1}$, $X = W_t$, $a = \nabla l_t(W_t x_t) = \tilde{p}_t - y_t$, $b = x_t$ and $M = Z_{t+1}$. For a solution to any such optimization problem, by the first order optimality conditions, we have:

$$\begin{aligned} \big\langle (Y - X)M + \eta a b^\top, W - Y \big\rangle &\geq 0 \\ \text{or } (Y - X)MW &\geq (Y - X)MY - \eta a^\top (W - Y) b \end{aligned}$$

Using this first order condition, we have

$$\begin{aligned}\|X-W\|_M^2 - \|Y-W\|_M^2 &= \sum_{i=1}^{S} X^i M X^i + W^i M W^i - Y^i M Y^i - W^i M W^i \\ &\quad + 2(Y^i - X^i) M W^i \\ &\geq \|X-Y\|_M^2 - 2\eta a^\top (W-Y) b \\ &= \|X-Y\|_M^2 + 2\eta a^\top (Y-X) b - 2\eta a^\top (W-X) b \\ &\geq \operatorname*{argmin}_{A \in \mathbb{R}^{S\times d}} \|A\|_M^2 + 2\eta a^\top A b - 2\eta a^\top (W-X) b \end{aligned} \tag{3.21}$$

Noting that $a = \tilde{p}_t - y^t$, we get

$$\operatorname*{argmin}_{A \in \mathbb{R}^{S\times d}} \|A\|_M^2 + 2\eta a^\top A b \geq \sum_{i=1}^{S} -\eta^2 a_i^2 \|b\|_{M^{-1}}^2 \geq -4\eta^2 \|b\|_{M^{-1}}^2$$

Substituting this and $W = W^*$ along with other terms in (3.21) proves the stated lemma in (3.20). □

Thus, from eqs. (3.18), (3.19) and (3.20), we have

$$\|W_{t+1} - W^*\|_{Z_{t+1}} \leq \|W_t - W^*\|_{Z_t} - \frac{\eta\alpha}{2} b_t + 2\eta c_t + 4\eta^2 \|x_t\|_{Z_{t+1}^{-1}}^2 \tag{3.22}$$

Bounding the first term on the `rhs` similarly, and telescoping the sum, we get:

$$\begin{aligned}\|W_{t+1} - W^*\|_{Z_{t+1}} + \frac{\eta\alpha}{2} \sum_{i=1}^{t} b_i &\leq \|W^*\|_{Z_1} + 2\eta \sum_{i=1}^{t} c_i + 4\eta^2 \sum_{i=1}^{t} \|x_i\|_{Z_{i+1}^{-1}}^2 \\ &\leq \lambda \|W^*\|_F^2 + 2\eta \sum_{i=1}^{t} c_i + 4\eta^2 \sum_{i=1}^{t} \|x_i\|_{Z_{i+1}^{-1}}^2 \end{aligned} \tag{3.23}$$

We will now bound the sum $\sum_{i=1}^{t} c_i$ in (3.23) using Bernstein's inequality for martingales in the same manner as Zhang et al. (2016):

Lemma 3.22. *With probability at least* $1-\delta$*, we have:*

$$\sum_{i=1}^{t} c_i \leq 4B_p R + \frac{\alpha}{4} \sum_{i=1}^{t} b_i + \left(\frac{4}{\alpha} + \frac{8B_p R}{3}\right) \tau_t \tag{3.24}$$

where $\tau_t = \log(2\lceil 2\log St \rceil t^2/\delta)$.

Proof. The result can be easily derived from the proof of Lemma 5 in Zhang et al. (2016). We

provide the key steps here for completeness.

We first note that c_t is a martingale difference sequence with respect to filtration $\mathcal{F}_t$ induced by the first t rounds including the next context x_{t+1}:

$$\mathbb{E}\left[(y_t - p_t)^\top (W_t - W^*)x_t | \mathcal{F}_{t-1}\right] = \mathbb{E}\left[(y_t - p_t)|\mathcal{F}_{t-1}\right]^\top (W_t - W^*)x_t = 0$$

Further, each term in this martingale series can be bounded as:

$$\begin{aligned} |c_t| = (y_t - p_t)^\top (W_t - W^*)x_t &\leq \|(y_t - p_t)\|_1 \|(W_t - W^*)x_t\|_\infty \\ &\leq 4B_pR \end{aligned}$$

Similarly, for martingale $C_t \coloneqq \sum_{i=1}^t c_i$, we bound the conditional variance as

$$\begin{aligned} \Sigma_t^2 = \sum_{i=1}^t \mathbb{E}_{y_i}\left[\left((y_t - p_t)^\top (W_t - W^*)x_t\right)^2\right] &\leq \sum_{i=1}^t \mathbb{E}_{y_i}\left[\left(y_t^\top (W_t - W^*)x_t\right)^2\right] \\ &\leq \underbrace{\sum_{i=1}^t \|(W_t - W^*)x_t\|_2^2}_{\coloneqq A_t} \end{aligned}$$

Thus, we have a natural upper bound for the conditional variance which is $\Sigma_t^2 \leq 4B_p^2R^2St$. Now, consider two scanarios: `CASE I`: $A_t \geq 4B_p^2R^2/St$ and `CASE II`: $4B_p^2R^2/St \leq A_t \leq 4B_p^2R^2St$.

`CASE I`: Here, we directly bound the sum as

$$\begin{aligned} x_t \leq \sum_{i=1}^t |c_i| &\leq 2\sum_{i=1}^t \|(W_t - W^*)x_t\|_2 \\ &\leq 2\sqrt{t\sum_{i=1}^t \|(W_t - W^*)x_t\|_2^2} \leq 4B_pR \end{aligned}$$

`CASE II`: We directly use the expression after applying Bernstein's inequality along with the peeling technique from Zhang et al. (2016). Using that, we have:

$$\begin{aligned} &P\left[C_t \geq 2\sqrt{A_t\tau_t} + \frac{8B_pR\tau_t}{3}\right] \\ \leq &\sum_{j=-\log S}^{m} P\left[C_t \geq 2\sqrt{A_t\tau_t} + \frac{8B_pR\tau_t}{3}, \frac{4B_pR^2 2^j}{t} \leq A_t \leq \frac{4B_pR^2 2^{j+1}}{t}\right] \\ \leq &\, m' e^{-\tau_t} \end{aligned}$$

where $m = \log St^2$ and $m' = m + \log S = \log S^2t^2$. We set $\tau_t = \log \frac{2m't^2}{\delta}$, we get that with probability at least $1 - \delta/2t^2$, we have:

$$C_t \leq 2\sqrt{A_t \tau_t} + \frac{8B_pR\tau_t}{3}$$

Taking a union bound over $t \geq 0$ and substituting $A_t = \sum_{i=1}^{t} b_i$, with probability at least $1 - \delta$, for all $t \geq 0$, we get:

$$\sum_{i=1}^{t} c_i \leq 4B_pR + 2\sqrt{\tau_t \sum_{i=1}^{t} b_i} + \tfrac{8B_pR}{3}\tau_t$$

Using the RMS-AM inequality, we get the desired expression:

$$\sum_{i=1}^{t} c_i \leq 4B_pR + \frac{\alpha}{4}\sum_{i=1}^{t} b_i + \left(\frac{4}{\alpha} + \frac{8B_pR}{3}\right)\tau_t$$

□

Substituting the high probability upper bound over $\sum_{i=1}^{t} c_i$ in (3.23), we get:

$$\|W_{t+1} - W^*\|_{Z_{t+1}} \leq \lambda\|W^*\|_F^2 + 2\eta\left[4B_pR + \left(\frac{4}{\alpha} + \frac{8}{3}B_pR\right)\tau_t\right] + 4\eta^2\sum_{i=1}^{t}\|x_t\|^2_{Z_{t+1}^{-1}} \tag{3.25}$$

For getting the final result, we now bound the elliptic potential using the following Lemma from Zhang et al. (2016):

Lemma 3.23 (Lemma 6 (Zhang et al., 2016))**.** *The elliptic potential term can be bounded as follows:*

$$\sum_{i=1}^{t}\|x_t\|^2_{Z_{t+1}^{-1}} \leq \frac{2}{\eta\alpha}\log\frac{\det(Z_{t+1})}{\det(Z_1)}$$

3.8.2.2 Proof of regret bound for GLM-ORL (Theorem 3.7)

We now provide a complete proof of Theorem 3.7.

Failure events and bounding failure probabilities To begin with, we write the important failure events for the algorithm $F = F^{(r)} \cup F^{(p)} \cup F^{(O)}$ where each sub-event is defined as follows:

$$F^{(O)} := \left\{ \exists T \in \mathbb{N} : \sum_{t,h,s,a} \left(P_t[s_h, a_h = s, a | s_{t,1}] - \mathbb{1}[s_{t,h} = s, a_{t,h} = a] \right) \geq SH\sqrt{T \log \tfrac{6\log(2T)}{\delta_1}} \right\}$$

$$F^{(p)} := \left\{ \exists\, s \in \mathcal{S}, a \in \mathcal{A}, t \in \mathbb{N} : \left\| W_{sa} - \widehat{W}_{t,sa} \right\|_{Z_{t,sa}} \geq \sqrt{\gamma_{t,sa}} \right\}$$

$$F^{(r)} := \left\{ \exists\, s \in \mathcal{S}, a \in \mathcal{A}, t \in \mathbb{N} : \left\| \theta_{sa} - \hat{\theta}_{t,sa} \right\|_{Z_{t,sa}} \geq \zeta_{t,sa} \right\}$$

Using high-probability guarantees for parameter estimation and concentration of measure, we have the guarantee that:

Lemma 3.24. *The probabilities for failure events $F^{(O)}, F^{(p)}$ and $F^{(r)}$ are bounded bounded by $SH\delta_1$, $SA\delta_p$ and $SA\delta_r$ respectively.*

Proof. The guarantee for $F^{(p)}$ follows from Theorem 3.6 in Section 3.4.2. The failure probability $P(F^{(r)})$ can be bounded by using Theorem 20.5 from Lattimore and Szepesvári (2020).

Lastly, the failure probability $P(F^{(O)})$ is directly taken from Lemma 23 of Dann et al. (2019). □

Regret incurred outside failure events

Lemma 3.25 (Optimism)**.** *If all the confidence intervals as computed in Algorithm 3.5 are valid for all episodes t, then outside of failure event F, for all t and $h \in [H]$ and $s, a \in \mathcal{S} \times \mathcal{A}$, we have:*

$$\widetilde{Q}_{t,h}(s,a) \geq Q^*_{t,h}(s,a)$$

Proof. For every episode, the lemma is true trivially for $H + 1$. Assume that it is true for $h + 1$. For h, we have:

$$\begin{aligned}
&\widetilde{Q}_{t,h}(s,a) - Q^*_{t,h}(s,a) \\
&= \left(\widehat{P}_t(s,a)^\top \widetilde{V}_{t,h+1} + \widehat{R}_t(s,a) + \varphi_{t,h}(s,a) \right) \wedge V_h^{\max} - P_t(s,a)^\top V^*_{t,h+1} - R_t(s,a) \\
&= \widehat{R}_t(s,a) - R_t(s,a) + \widehat{P}_t(s,a)^\top \left(\widetilde{V}_{t,h+1} - V^*_{t,h+1} \right) + \varphi_{t,h}(s,a) - \left(P_t(s,a) - \widehat{P}_t(s,a) \right)^\top V^*_{t,h+1} \\
&\geq -\left| \widehat{R}_t(s,a) - R_t(s,a) \right| + \varphi_{t,h}(s,a) - \left\| P_t(s,a) - \widehat{P}_t(s,a) \right\|_1 \left\| \widetilde{V}_{t,h+1} \right\|_\infty \geq 0
\end{aligned}$$

In the second equality step, we use the fact that when $\widetilde{Q}_{t,h}(s,a) = V_h^{\max}$, the requirement is trivially satisfied. When $\widetilde{Q}_{t,h}(s,a) < V_h^{\max}$, the step follows by definition. The last step uses the guarantee on confidence intervals and the inductive assumption for $h + 1$. Therefore, the estimated Q-values are optimistic by induction. □

Therefore, using the optimism guarantee, we can bound the instantaneous regret Δ_t in episode t as: $V^*_{t,1}(s) - V^{\pi_t}_{t,1}(s) \leq \widetilde{V}_{t,1}(s) - V^{\pi_t}_{t,1}(s)$. Thus, we have:

$$\begin{aligned}
\Delta_t &\leq \widetilde{V}_{t,1}(s) - V^{\pi_t}_{t,1}(s) \\
&\leq \left(\widehat{P}_t(s,a)^\top \widetilde{V}_{t,2} + \widehat{R}_t(s,a) + \varphi\right) \wedge V_1^{\max} - P_t(s,a)^\top V^{\pi_t}_{t,2} - R_t(s,a) \\
&\leq \left(\varphi + \left(\widehat{P}_t(s,a) - P_t(s,a)\right)^\top \widetilde{V}_{t,2} + \widehat{R}_t(s,a) - R_t(s,a)\right) \wedge V_1^{\max} \\
&\quad + P_t(s,a)^\top \left(V^{\pi_t}_{t,2} - \widetilde{V}_{t,2}\right) \\
&\leq 2\varphi \wedge V_1^{\max} + P_t(s,a)^\top \left(V^{\pi_t}_{t,2} - \widetilde{V}_{t,2}\right) \\
&\leq \sum_{h,s,a} \mathbb{P}_t[s_h, a_h = s, a | s_{t,1}] \left(2\varphi(s,a) \wedge V_h^{\max}\right) \qquad (3.26)
\end{aligned}$$

Using Lemma 3.24, we can show the following result:

Lemma 3.26. *Outside the failure event $F^{(O)}$, i.e., with probability at least $1 - SH\delta_1$, the total regret $R(T)$ can be bounded by*

$$\text{Regret}(T) \leq SH^2\sqrt{T \log \tfrac{6 \log 2T}{\delta_1}} + \sum_{t=1}^{T}\sum_{h=1}^{H} \mathbb{1}_{t,h}(s,a) \cdot \left(2\varphi_{t,h}(s_{t,h}, a_{t,h}) \wedge V_h^{\max}\right) \qquad (3.27)$$

Proof.

$$\begin{aligned}
\Delta_t &\leq \sum_{h,s,a} \mathbb{P}_t\left[s_h, a_h = s, a | s_{t,1}\right] \left(2\varphi(s,a) \wedge V_h^{\max}\right) \\
&\leq \sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{s,a} \left(\mathbb{P}_t[s_h, a_h = s, a | s_{t,1}] - \mathbb{1}_{t,h}(s,a)\right) \left(2\varphi(s_{t,h}, a_{t,h}) \wedge V_h^{\max}\right) \\
&\quad + \sum_{t=1}^{T}\sum_{h=1}^{H} \mathbb{1}_{t,h}(s,a)(2\varphi(s_{t,h}, a_{t,h}) \wedge V_h^{\max})
\end{aligned}$$

where $\mathbb{1}_{t,h}(s,a)$ is the indicator function $\mathbb{1}[s_{t,h} = s, a_{t,h} = a]$. From Lemma 3.24, we know that the first term is bounded by $SH\sqrt{T \log \frac{6 \log 2T}{\delta_1}}$ with probability at least $1 - SH\delta_1$. □

Before bounding the second term in (3.27), we state the following Lemma from Abbasi-Yadkori et al. (2011) which is used frequently in our analysis:

Lemma 3.27 (Determinant-Trace inequality)**.** *Suppose $X_1, X_2, \ldots, X_t \in \mathbb{R}^d$ and for any $1 \leq s \leq t$,*

$\|X_s\|_2 \leq L$. *Let* $V_t := \lambda \mathbf{I} + \sum_{s=1}^{t} X_s X_s^\top$ *for some* $\lambda \geq 0$. *Then, we have:*

$$\det(V_t) \leq \left(\lambda + tL^2/d\right)^d$$

The second term in (3.27) can now be bounded as follows:

$$\sum_{t=1}^{T}\sum_{h=1}^{H}(2\varphi(s_{t,h},a_{t,h}) \wedge V_h^{\max}) \leq \sum_{t=1}^{T}\sum_{h=1}^{H}\left(2\xi^{(r)}_{t,s_{t,h},a_{t,h}} \wedge V_h^{\max}\right) + \sum_{t=1}^{T}\sum_{h=1}^{H}\left(2V_{h+1}^{\max}\xi^{(p)}_{t,s_{t,h},a_{t,h}} \wedge V_h^{\max}\right) \tag{3.28}$$

We ignore the reward estimation error in (3.28) as it leads to lower order terms. The second expression can be again bounded as follows:

$$\sum_{t,2}^{T}\sum_{h=1}^{H}(2V_{h+1}^{\max}\xi^{(p)}_{t,s_{t,h},a_{t,h}} \wedge V_h^{\max}) \leq 2\sum_{t,h} V_h^{\max}\left(1 \wedge \beta\sqrt{S\gamma_t(s_{t,h},a_{t,h})}\|x_t\|_{Z^{-1}_{t,sa,h}}\right) \tag{3.29}$$

Using Lemma 3.27, we see that

$$\begin{aligned}
\gamma_t(s,a) &:= f_\Phi(t,\delta_p) + \frac{8\eta}{\alpha}\log\frac{\det(Z_{t,sa})}{\det(Z_{1,sa})} \\
&\leq \frac{\eta\alpha}{2S} + f_\Phi(TH,\delta_p) + \frac{8\eta}{\alpha}\log\frac{\det(Z_{T+1,sa})}{\det(Z_{1,sa})} \\
&\leq \frac{\eta\alpha}{2S} + f_\Phi(TH,\delta_p) + \frac{8\eta d}{\alpha}\log\left(1 + \frac{THR^2}{\lambda d}\right)
\end{aligned}$$

We use $f_\Phi(t,\delta_p)$ to refer to the Z_t independent terms in (3.12). Setting $\bar{\gamma}_T$ to the last expression guarantees that $\frac{2S\bar{\gamma}_T}{\eta\alpha} \geq 1$. We can now bound the term in (3.29) as:

$$2\beta V_1^{\max}\sqrt{\frac{2S\bar{\gamma}_T}{\eta\alpha}}\sum_{t,h}\left(1 \wedge \sqrt{\frac{\eta\alpha}{2}}\|x_t\|_{Z^{-1}_{t,sa,h}}\right) \leq 2\beta V_1^{\max}\sqrt{\frac{2S\bar{\gamma}_T TH}{\eta\alpha}}\sqrt{\sum_{t,h}\left(1 \wedge \frac{\eta\alpha}{2}\|x_t\|^2_{Z^{-1}_{t,sa,h}}\right)} \tag{3.30}$$

Ineq. line 3.30 follows by using Cauchy-Schwarz inequality. We now bound the elliptic potential term inside the square root in line 3.30:

Lemma 3.28. *For any $T \in \mathbb{N}$, we have:*

$$\sum_{t,h}\left(1 \wedge \frac{\eta\alpha}{2}\|x_t\|^2_{Z^{-1}_{t,sa,h}}\right) \leq 2H \sum_{s,a} \log\left(\frac{\det Z_{t+1,sa}}{\det Z_{t,sa}}\right)$$

Proof. Note that, instead of summing up the weighted operator norm with changing values of $Z_{t,h}$ for each observed transition of a pair (s,a), we keep the matrix same for all observations in an episode. Note that, Z_t denotes the matrix at the beginning of episode t and therefore, does not include the terms $x_t x_t^\top$. Thus, for any episode t:

$$\begin{aligned}
\sum_{h=1}^{H}\left(1 \wedge \frac{\eta\alpha}{2}\|x_t\|^2_{Z^{-1}_{t,sa,h}}\right) &\leq 2\sum_{s,a}\sum_{h=1}^{H} \mathbb{1}_{t,h}(s,a)\log\left(1+\frac{\eta\alpha}{2}\|x_t\|^2_{Z^{-1}_{t,sa}}\right) \\
&= 2\sum_{s,a} N_t(s,a)\log\left(1+\frac{\eta\alpha}{2}\|x_t\|^2_{Z^{-1}_{t,sa}}\right) \\
&\leq 2\sum_{s,a} N_t(s,a)\log\left(1+N_t(s,a)\frac{\eta\alpha}{2}\|x_t\|^2_{Z^{-1}_{t,sa}}\right) \\
&= 2H\sum_{s,a}\log\left(\frac{\det Z_{t+1,sa}}{\det Z_{t,sa}}\right)
\end{aligned}$$

where in the last step, we have used the following:

$$Z_{t+1} = Z_t^{1/2}\left(1+\frac{\eta\alpha}{2}N_t Z_t^{-1/2}x_t x_t^\top Z_t^{-1/2}\right)Z_t^{1/2}$$

and then bound the determinant ratio using

$$\det Z_{t+1} = \det Z_t\left(1+N_t\frac{\eta\alpha}{2}\|x_t\|^2_{Z_t^{-1}}\right)$$

□

Finally, by using Lemma 3.27, we can bound the term as

$$\sum_{t=1}^{T}\sum_{h=1}^{H}(2V_{h+1}^{\max}\xi^{(p)}_{t,s_{t,h},a_{t,h}} \wedge V_h^{\max}) \leq 4\beta V_1^{\max}\sqrt{\frac{2S\bar{\gamma}_T TH}{\eta\alpha}}\sqrt{2HSAd\log\left(1+\frac{THR^2}{\lambda d}\right)}$$

Now, we set each individual failure probability$\delta_1 = \delta_p = \delta_r = \delta/(2SA+SH)$. Upon taking a union bound over all events, we get the total failure probability as δ. Therefore, with probability at

least $1-\delta$, we can bound the regret of GLM-ORL as

$$\text{Regret}(T) = \widetilde{O}\left(\left(\frac{\sqrt{d}\max_{s,a}\|W^*_{sa}\|_F}{\sqrt{\alpha}} + \frac{d}{\alpha}\right)\beta SH^2\sqrt{AT}\right)$$

where $\max_{s,a}\|W^*_{sa}\|_F$ is replaced by the problem dependent upper bound assumed to be known apriori.

3.8.2.3 Proof of regret bound for GLM-RLSVI (Theorem 3.11)

Our analysis will closely follow the proof from Russo (2019). We start by writing the concentration result for estimating MDP M_t by using Algorithm 3.4 and the linear bandit estimators. For notation, we use $\widehat{M}_t$ to denote the MDP constructed using the estimates $\widehat{W}_t$ and $\widehat{\theta}_t$. The perturbed MDP used in the algorithm is denoted by $\overline{M}_t$ and $\widetilde{M}_t$ will denote an MDP constructed using another set of *i.i.d.* reward bonuses as $\overline{M}_t$. Specifically, we have:

Lemma 3.29. *Let $\mathcal{M}_t$ be the following set of MDPs:*

$$\mathcal{M}_t := \{(P', R') : \forall(h,s,a),\ |(R'(s,a) - R_t(s,a)) + \langle P'(s,a) - P_t(s,a), V_{t,h+1}\rangle| \le \varphi_{t,h}(s,a)\}$$

where $\varphi^2_{t,h}(s,a) = \left(\beta\sqrt{S\gamma_{t,sa}}(H-h) + \zeta_{t,sa}\right)\|x_t\|_{Z^{-1}_{t,sa}}$. If we choose $\delta_p = \delta_r = \pi^2/SA$, then, we have:

$$\sum_{t\in\mathbb{N}} \mathbb{P}\left[\widehat{M}_t \notin \mathcal{M}_t\right] \le \frac{\pi^2}{6}$$

Proof. The proof follows from the analysis in Section 3.8.2.2 where the union bound over all (s,a) pairs gives the total failure probability to be $\frac{\pi^2}{6}$. □

Given the concentration result, Lemma 4 from Russo (2019) directly applies to the CMDP setting in the following form:

Lemma 3.30. *Let π^*_t be the optimal policy for MDP M_t. If $\widehat{M}_t \in \mathcal{M}_t$ and reward bonuses $b_{t,h}(s,a) \sim N(0, HS\varphi^2_{t,h}(s,a))$, then we have*

$$\mathbb{P}\left[v^{\pi_t}_{\overline{M}_t} \ge v^{\pi^*}_{M_t} | \mathcal{H}_{t-1}\right] \ge \mathbb{F}(-1)$$

where $\widehat{M}_t$ is the estimated MDP, $\overline{M}_t$ is the MDP obtained after perturbing the rewards and $\mathbb{F}(\cdot)$ is the cdf for the standard normal distribution.

In a similar fashion, the following result can also be easily verified:

Lemma 3.31. *For an absolute constant $c = \mathbb{F}(-1)^{-1} \leq 6.31$, we have:*

$$\begin{aligned}\text{Regret}(T) &:= \mathbb{E}_{\text{ALG}}\left[\sum_{t=1}^{T} v_t^*(s_{t,1}) - v_t^{\pi_t}(s_{t,1})\right] \\ &\leq (c+1)\mathbb{E}\left[\sum_{t=1}^{T}\left|v_{\overline{M}_t}^{\pi_t} - v_{M_t}^{\pi_t}\right|\right] + c\mathbb{E}\left[\sum_{t=1}^{T}\left|v_{\widetilde{M}_t}^{\pi_t} - v_{M_t}^{\pi_t}\right|\right] + H\frac{\pi^2}{6}\end{aligned}$$

We will now bound the first term on the `rhs` of Lemma 3.31 to get the final regret bound. The second term can be bounded in the same manner. For each episode, the summand in the first term can be written as:

$$\begin{aligned}&v_{\overline{M}}^{\pi_t}(s_{t,1}) - v_{M_t}^{\pi_t}(s_{t,1}) \\ =&\mathbb{E}\left[\sum_{h=1}^{H}\left(\left\langle P_t(s_{t,h},a_{t,h}) - \widehat{P}_t(s_{t,h},a_{t,h}), \overline{V}_{t,h+1}\right\rangle\right)\Big|\mathcal{H}_{t-1}\right] \\ &+\mathbb{E}\left[\sum_{h=1}^{H}\left(\widehat{R}_t(s_{t,h},a_{t,h}) - R_t(s_{t,h},a_{t,h}) + b_{t,h}(s_{t,h},a_{t,h})\right)\Big|\mathcal{H}_{t-1}\right] \\ \leq&\mathbb{E}\left[\sum_{h=1}^{H}\left\langle P_t(s_{t,h},a_{t,h}) - \widehat{P}_t(s_{t,h},a_{t,h}), \overline{V}_{t,h+1}\right\rangle\right] + \mathbb{E}\left[\sum_{h=1}^{H} R_t(s_{t,h},a_{t,h}) - \widehat{R}_t(s_{t,h},a_{t,h})|\mathcal{H}_{t-1}\right] \\ &+\mathbb{E}\left[\sum_{h=1}^{H}|b_{t,h}(s_{t,h},a_{t,h})|\,\Big|\mathcal{H}_{t-1}\right] \qquad (3.31)\end{aligned}$$

where $\overline{V}_{t,h+1}$ denotes the h^{th}-step value of policy π_t in $\overline{M}_t$. We will now bound each term individually where we ignore the reward term and the variance component due to reward uncertainty as both lead to lower order terms. Specifically, we directly consider $\varphi_{t,h}^2(s,a) = 2\left(\beta\sqrt{S\gamma_{t,sa}}(H-h)\right)\|x_t\|_{Z_{t,sa}^{-1}}$. For the last expression in (3.31), we focus on the first and third terms (the reward bonuses lead to lower order terms in the final regret bound).

Lemma 3.32. *We have:*

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}|b_{t,h}(s_{t,h},a_{t,h})|\,\Big|\mathcal{H}_{t-1}\right] = \widetilde{O}\left(\left(\frac{\sqrt{d}\max_{s,a}\|W_{sa}^*\|_F}{\sqrt{\alpha}} + \frac{d}{\alpha}\right)\beta S^{3/2}H^{5/2}\sqrt{AT}\right)$$

Proof. We write $b_{t,h}(s_{t,h},a_{t,h}) = \sqrt{HS}\varphi_{t,h}(s_{t,h},a_{t,h})\xi_{t,h}(s_{t,h},a_{t,h})$ where $\xi_{t,h}(s_{t,h},a_{t,h}) \sim N(0,1)$.

Therefore, by using Holder's inequality, we have:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}|b_{t,h}(s_{t,h},a_{t,h})|\,\middle|\,\mathcal{H}_{t-1}\right] \leq \mathbb{E}\left[\max_{t,h,s,a}\xi_{t,h}(s,a)\right]\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}\sqrt{HS}\varphi_{t,h}(s_{t,h},a_{t,h})\right]$$

By using (sub)-Gaussian maximal inequality, we know that

$$\mathbb{E}\left[\max_{t,h,s,a}\xi_{t,h}(s,a)\right] = O\left(\log(HSAT)\right) \tag{3.32}$$

For the second expression, we have:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}\sqrt{HS}\varphi_{t,h}(s_{t,h},a_{t,h})\right] &\leq \sqrt{HS}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}\varphi_{t,h}(s_{t,h},a_{t,h})\right] \\ &\leq 2H^{3/2}\sqrt{S}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}1\wedge\left(\beta\sqrt{S}\sqrt{\gamma_{t,sa}}\|x_t\|_{Z_{t,sa}^{-1}}\right)\right]\end{aligned}$$

where we used the definition of $\overline{\xi}_{t,h}^{(p)}$ used in Section 3.4.4. Using the upper bound above along with Lemma 3.27 and Lemma 3.28, we obtain the bound:

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{h=1}^{H}\sqrt{HS}\varphi_{t,h}(s_{t,h},a_{t,h})\right] = O\left(\beta H^{5/2}S^{3/2}\sqrt{\frac{dA\bar{\gamma}_T T}{\eta\alpha}}\sqrt{\log\left(1+\frac{THR^2}{\lambda d}\right)}\right) \tag{3.33}$$

We get the final bound on the term by combining eqs. (3.32) and (3.33). □

We now bound the first term in (3.31):

Lemma 3.33. *With the ONS estimation method and the used randomized bonus, we have:*

$$\begin{aligned}&\mathbb{E}\left[\sum_{t,h}\left|\left\langle P_t(s_{t,h},a_{t,h})-\widehat{P}_t(s_{t,h},a_{t,h}),\overline{V}_{t,h+1}\right\rangle\right|\right] \\ &= \widetilde{O}\left(\left(\frac{\sqrt{d}\max_{s,a}\|W_{sa}^*\|_F}{\sqrt{\alpha}}+\frac{d}{\alpha}\right)\beta\sqrt{H^7S^3AT}\right)\end{aligned}$$

Proof. We first rewrite the expression:

$$\mathbb{E}\left[\sum_{t,h}\left|\left\langle P_t(s_{t,h},a_{t,h})-\widehat{P}_t(s_{t,h},a_{t,h}),V_{t,h+1}\right\rangle\right|\right] \leq \mathbb{E}\left[\sum_{t,h}\|\epsilon_t^p(s_{t,h},a_{t,h})\|_1\|V_{t,h+1}\|_\infty\right]$$

where $\epsilon_t^p(s_{t,h},a_{t,h}) = P_t(s_{t,h},a_{t,h})-\widehat{P}_t(s_{t,h},a_{t,h})$. Using Cauchy-Schwarz inequality, we rewrite

this as:

$$\sqrt{\mathbb{E}\left[\sum_{t,h}\|\epsilon_t^p(s_{t,h},a_{t,h})\|_1^2\right]}\sqrt{\mathbb{E}\left[\sum_{t,h}\|V_{t,h+1}\|_\infty^2\right]}$$

For bounding the sum of values under the second square root, we can directly use the Lemma 8 from Russo (2019):

$$\sqrt{\mathbb{E}\left[\sum_{t,h}\|V_{t,h+1}\|_\infty^2\right]} = \widetilde{O}(H^3\sqrt{ST}) \tag{3.34}$$

For bounding the expected estimation error, we consider two events: $F^{(p)}$ when the confidence widths are incorrect and $(F^{(p)})_x$ when the confidence intervals are valid for all (s,a), t and h. Therefore, we have:

$$\begin{aligned}\mathbb{E}\left[\sum_{t,h}\|\epsilon_t^p(s_{t,h},a_{t,h})\|_1^2\right] &= \mathbb{E}\left[\sum_{t,h}\|\epsilon_t^p(s_{t,h},a_{t,h})\|_1^2|F^{(p)}\right]P(F^{(p)}) \\ &\quad+\mathbb{E}\left[\sum_{t,h}\|\epsilon_t^p(s_{t,h},a_{t,h})\|_1^2|(F^{(p)})_x\right]P((F^{(p)})_x)\end{aligned}$$

Setting $\delta_p = 1/TH$, we can bound the sum under failure event to a constant. For the other term, we see that it is equivalent to:

$$\begin{aligned}\mathbb{E}\left[\sum_{t,h}\|\epsilon_t^p(s_{t,h},a_{t,h})\|_1^2|(F^{(p)})_x\right]P((F^{(p)})_x) &\le \mathbb{E}\left[\sum_{t,h}\left(1\wedge\beta\sqrt{S\gamma_t(s_{t,h},a_{t,h})}\|x_t\|_{Z_{t,sa,h}^{-1}}\right)^2\right] \\ &\le \frac{2\beta^2 S\bar{\gamma}_T}{\eta\alpha}\mathbb{E}\left[\sum_{t,h}\left(1\wedge\frac{\eta\alpha}{2}\|x_t\|_{Z_{t,sa,h}^{-1}}^2\right)\right] \\ &= \widetilde{O}\left(\left(\frac{d\max_{s,a}\|W_{sa}^*\|_F^2}{\alpha}+\frac{d^2}{\alpha^2}\right)\beta^2S^2AH\right)\end{aligned} \tag{3.35}$$

Combining eqs. (3.34) and (3.35), we get the desired result. □

The final regret guarantee can be obtained by adding terms from Lemma 3.32 and Lemma 3.33.

3.8.2.4 Proof of Mistake Bound of GLM-ORL

In order to prove the mistake bound, we need to bound the number of episodes where the policy's value is more than ϵ-suboptimal. We start with the inequality in (3.26):

$$V^*_{t,1}(s) - V^{\pi_t}_{t,1}(s) \leq \sum_{h,s,a} \mathbb{P}_t[s_h, a_h = s, a | s_{t,1}](2\varphi_{t,h}(s,a) \wedge V^{\max}_h)$$

We note that if $\varphi_{t,h}(s,a) \leq \frac{\epsilon}{2H}$ for all t, h and (s,a), then we have

$$V^*_{t,1}(s) - V^{\pi_t}_{t,1}(s) \leq \sum_{h,s,a} \mathbb{P}_t[s_h, a_h = s, a | s_{t,1}] \frac{\epsilon}{H} \leq \epsilon$$

In order to satisfy the constraint, we bound each error term as: $\xi^{(p)} \leq \frac{\epsilon}{4H^2}$ and $\xi^{(r)} \leq \frac{\epsilon}{4H}$.

We bound the number of episodes where this constraint is violated. For simplicity, we consider that the rewards are known and only consider the transition probabilities in the analysis:

$$\begin{aligned}
\sum_{t \in [T]_+} \mathbb{1}\left[\exists (s,a) \text{ s.t. } \xi^{(p)}_{t,sa} \geq \frac{\epsilon}{4H^2}\right] &\leq \sum_{t \in [T]_+} \sum_{s,a} \mathbb{1}\left[\beta \sqrt{S} \sqrt{\gamma_{t,sa}} \|x_t\|_{Z^{-1}_{t,sa}} \geq \frac{\epsilon}{4H^2}\right] \\
&\leq \sum_{t \in [T]_+} \sum_{s,a} \frac{16\beta^2 S H^4 \gamma_{t,sa}}{\epsilon^2} \|x_t\|^2_{Z^{-1}_{t,sa}} \\
&\leq \frac{16\beta^2 S H^4 \gamma_{T+1}}{\epsilon^2} \sum_{t \in [T]_+} \sum_{s,a} \|x_t\|^2_{Z^{-1}_{t,sa}} \\
&\leq \frac{16\beta^2 H^4 \gamma_{T+1}}{\epsilon^2} \sum_{s,a} \sum_{t \in [T]_+} \|x_t\|^2_{Z^{-1}_{t,sa}} \qquad (3.36)
\end{aligned}$$

where in the intermediate steps, we have used the nature of the indicator function and the fact that minimum is upper bounded by the average. Assuming that $N_{t,sa}$ denotes the number of visits to pair (s,a) in episode t, we rewrite the inner term as:

$$\begin{aligned}
\|x_t\|^2_{Z^{-1}_{t+1,sa}} &= x_t^\top (Z_t + N_{t,sa} x_t x_t^\top)^{-1} x_t \\
&= x_t^\top Z_{t,sa} x_t - \frac{N_{t,sa} x_t^\top Z^{-1}_{t,sa} x_t x_t^\top Z^{-1}_{t,sa} x_t}{1 + N_{t,sa} x_t^\top Z^{-1}_{t,sa} x_t} \\
&= \|x_t\|^2_{Z^{-1}_{t,sa}} - \frac{N_t \|x_t\|^4_{Z^{-1}_{t,sa}}}{1 + N_{t,sa} \|x_t\|^2_{Z^{-1}_{t,sa}}}
\end{aligned}$$

With this setup, we get:

$$\begin{aligned}\|x_t\|^2_{Z^{-1}_{t,sa}} &= \frac{\|x_t\|^2_{Z^{-1}_{t+1,sa}}}{1 - N_{t,sa}\|x_t\|^2_{Z^{-1}_{t+1,sa}}} \\ &\leq \frac{\lambda + H}{\lambda}\|x_t\|^2_{Z^{-1}_{t+1,sa}} \\ &\leq \frac{\lambda + H}{\lambda}\langle Z^{-1}_{t+1,sa}, N_{t,sa}x_t x_t^\top\rangle\end{aligned}$$

Using Lemma 11 from Hazan et al. (2007), the inner sum in (3.36), can be bounded as:

$$\frac{\lambda + H}{\lambda}\sum_{t\in[T]_+}\|x_t\|^2_{Z^{-1}_{t+1}} \leq d\log\left(\frac{R^2TH}{\lambda} + 1\right)$$

Combining all these bounds, we get:

$$\sum_{t\in[T]_+}\mathbb{1}\left[\exists(s,a)\text{ s.t. }\xi^{(p)}_{t,sa} \geq \frac{\epsilon}{4H^2}\right] \leq \frac{16(\lambda + H)\beta^2 dS^2AH^4\gamma_{T+1}}{\lambda\epsilon^2}\log\left(\frac{R^2TH}{\lambda} + 1\right)$$

Noting that $\gamma_{T+1} = O\left(\frac{d\log^2 TH}{\alpha} + S\right)$, we get the final mistake bound as:

$$O\left(\frac{dS^2AH^5\log TH}{\epsilon^2}\left(\frac{d\log^2 TH}{\alpha} + S\right)\right)$$

ignoring $O(\text{poly}(\log\log TH))$ terms.

CHAPTER 4

Best Policy Identification in Linear Mixture MDPs

In the previous chapter, we considered a multi-task RL problem where the agent interacts with a sequence of tabular environments. One of the main results considered a linear setting where an MDP in a given sequence is obtained as a linear combination of set of basis models and the combination coefficients parameterized the tasks. Our results showed that such a structure can be exploited to behave near-optimally in an online learning scenario by iteratively learning the underlying basis. In this chapter, we consider the flip of the structural assumption for a single task where the environment can be complex but can be expressed as linear combination of basis models.

Specifically, in this chapter, we consider a setting where we have access to an ensemble of models, which can be thought of as pre-trained and possibly inaccurate simulators (models). We approximate the real environment (which can be arbitrarily complex) using a state-dependent linear combination of the ensemble, where the coefficients are determined by the given state features and some unknown parameters. Our proposed algorithm provably learns a near-optimal policy with a sample complexity polynomial in the number of unknown parameters, and incurs no dependence on the size of the state (or action) space. As an extension, we also consider the more challenging problem of model selection, where the state features are unknown and can be chosen from a large candidate set. We provide exponential lower bounds that illustrate the fundamental hardness of this problem, and develop a provably efficient algorithm under additional natural assumptions.

4.1 Introduction

A common aspect in many of the success stories of RL, for instance, games (Mnih et al., 2015; Silver et al., 2016), simulated control problems (Todorov et al., 2012; Lillicrap et al., 2015; Mordatch et al., 2016) and a range of robotics tasks (Christiano et al., 2016; Tobin et al., 2017), is the use of simulation. Arguably, given a simulator of the real environment, it is possible to use RL to learn a near-optimal policy from (usually a large amount of) simulation data. If the simulator is highly accurate, the learned policy should also perform well in the real environment.

Apart from some cases where the true environment and the simulator coincide (e.g., in game playing) or a nearly perfect simulator can be created from the law of physics (e.g., in simple control problems), in general we will need to construct the simulator using data from the real environment, making the overall approach an instance of *model-based RL*[1]. As the algorithms for learning from simulated experience mature (which is what the RL community has mostly focused on), the bottleneck has shifted to the creation of a good simulator. *How can we learn a good model of the world from interaction experiences?*

A possible approach for meeting this challenge, is to learn using a wide variety of simulators, which imparts robustness and adaptivity to the learned policies. Recent works have demonstrated the benefits of using such an ensemble of models, which can be used to either transfer policies from simulated to real-world domains, or to simply learn robust policies (Andrychowicz et al., 2020; Tobin et al., 2017; Rajeswaran et al., 2017). Borrowing the motivation from these empirical works, we notice that the process of learning a simulator inherently includes various choices like inductive biases, data collection policy, design aspects etc. As such, instead of relying on a sole approximate model for learning in simulation, interpolating between models obtained from different sources can provide better approximation of the real environment. From a statistical perspective too, incorporating these different design choices in building the set of models allows us to adapt to differing sizes of available interaction data. Previous works like Buckman et al. (2018); Lee et al. (2019); Kurutach et al. (2018) have also demonstrated the effectiveness of using an ensemble of models for decreasing modelling error, or its effect thereof, during learning.

In this chapter, we consider building an approximate model of the real environment from interaction data using a set (or *ensemble*) of possibly inaccurate models, which we will refer to as the *base models*. The simplest way to combine the base models is to take a weighted combination, but such an approach is rather limited. For instance, each base model might be accurate in certain regions of the state space, in which case it is natural to consider a state-dependent mixture. We consider the problem of learning in such a setting, where one has to identify an appropriate combination of the base models through real-world interactions, so that the induced policy performs well in the real environment. The data collected through interaction with the real world can be a precious resource and, therefore, we need the learning procedure to be sample-efficient. Our main result is an algorithm that enjoys polynomial sample complexity guarantees, where the polynomial has no dependence on the size of the state and action spaces. We also study a more challenging setting where the featurization of states for learning the combination is unknown and has to be discovered from a large set of candidate features.

[1]Model-free RL agorithms provide an approach to circumvent the issue of learning a near-accurate model, but are marred with sample efficiency issues in practice. Hence, we focus on a theoretical model which utilizes model-based structural assumptions.

Outline. We formally set up the problem and notation in Section 4.2. The main algorithm is introduced in Section 4.3, together with its sample complexity guarantees. We then proceed to the feature selection problem in Section 4.4 and show a fundamental hardness of learning in the unknown feature case. In Section 4.9, we provide a detailed proof of the main result, and finally, conclude in Section 4.6 with a brief discussion of related work and results.

4.2 Problem Setup

In this chapter, we consider a setting where the agent interacts with an episodic MDP with arbitrarily large state-action spaces. The agent is given access to a set of K base MDPs $\{M_1, \dots, M_K\}$ which share the same $\mathcal{S}, \mathcal{A}, H, P_1$, and only differ in P and R. In addition, a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \Delta_{d-1}$ is given which maps state-action pairs to d-dimensional real vectors. Given these two objects, we consider the class of all models which can be obtained from the following state-dependent linear combination of the base models:

Definition 4.1 (Linear Combination of Base Models)**.** *For given model ensemble $\{M_1, \dots, M_K\}$ and the feature map $\phi : \mathcal{S} \times \mathcal{A} \to \Delta_{d-1}$, we consider models parameterized by W with the following transition and reward functions:*

$$P^W(\cdot|s,a) = \sum_{k=1}^{K} \left(W\phi(s,a)\right)[k] \cdot P^k(\cdot|s,a),$$
$$R^W(\cdot|s,a) = \sum_{k=1}^{K} \left(W\phi(s,a)\right)[k] \cdot R^k(\cdot|s,a).$$

We will use $M(W)$ to denote such a model for any parameter $W \in \mathcal{W}$ with $\mathcal{W}_0 \equiv \{W \in [0,1]^{K \times d} : \sum_{i=1}^{K} W_{ij} = 1$ for all $j \in [d]_+\}$.

For now, let's assume that there exists some W^* such that $M^* = M(W^*)$, i.e., the true environment can be captured by our model class; we will relax this assumption shortly. For notation, we use "$s_{h:H} \sim M$" to imply that the sequence of states are generated according to the dynamics of M. A policy is said to be optimal for M if it maximizes the value v_M^π. We denote such a policy as π_M and its value as v_M. We use π^* and v^* as shorthand for π_{M^*} and v_{M^*}, respectively.

To develop intuition, consider a simplified scenario where $d = 1$ and $\phi(s,a) \equiv 1$. In this case, the matrix W becomes a $K \times 1$ stochastic vector, and the true environment is approximated by a linear combination of the base models.

Example 4.1 (Global convex combination of models)**.** *If the base models are combined using a set of constant weights $w \in \Delta_{K-1}$, then this is a special case of Definition 4.1 where $d = 1$ and each*

state's feature vector is $\phi(s,a) \equiv 1$.

In the more general case of $d > 1$, we allow the combination weights to be a linear transformation of the features, which are $W\phi(s,a)$, and hence obtain more flexibility in choosing different combination weights in different regions of the state-action space. A special case of this more general setting is when ϕ corresponds to a partition of the state-action space into multiple groups, and the linear combination coefficients are constant within each group.

Example 4.2 (State space partition)**.** *Let* $\mathcal{S} \times \mathcal{A} = \bigcup_{i \in [d]_+} \mathcal{X}_i$ *be a partition (i.e.,* $\{\mathcal{X}_i\}$ *are disjoint). Let* $\phi_i(s,a) = \mathbb{1}[(s,a) \in \mathcal{X}_i]$ *for all* $i \in [d]_+$ *where* $\mathbb{1}[\cdot]$ *is the indicator function. This* ϕ *satisfies the condition that* $\phi(s,a) \in \Delta_{d-1}$*, and when combined with a set of base models, forms a special case of Definition 4.1.*

Goal. We consider the best policy identification problem (described in Section 2.2.2) in this chapter: with probability at least $1-\delta$, the algorithm should output a policy π with value $v^\pi_{M^*} \geq v^* - \epsilon$ by collecting $\mathrm{poly}\,(d, K, H, 1/\epsilon, \log(1/\delta))$ episodes of data. Importantly, here the sample complexity is not allowed to depend on $|\mathcal{S}|$ or $|\mathcal{A}|$. However, the assumption that M^* lies the class of linear models can be limiting and, therefore, we will allow some approximation error in our setting as follows:

$$\theta \coloneqq \min_{W \in \mathcal{W}} \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} \left\| P^*(\cdot|s,a) - P^W(\cdot|s,a) \right\|_1 + \left| R^*(\cdot|s,a) - R^W(\cdot|s,a) \right| \tag{4.1}$$

We denote the optimal parameter attaining this value by W^*. The case of $\theta = 0$ represents the *realizable* setting where $M^* = M(W^*)$ for some $W^* \in \mathcal{W}$. When $\theta \neq 0$, we cannot guarantee returning a policy with the value close to v^*, and will have to pay an additional penalty term proportional to the approximation error θ, as is standard in RL theory.

Further Notations Let π_W be a shorthand for $\pi_{M(W)}$, the optimal policy in $M(W)$. When referring to value functions and state-action distributions, we will use the superscript to specify the policy and use the subscript to specify the MDP in which the policy is evaluated. For example, we will use $V^W_{W',h}$ to denote the value of π_W (the optimal policy for model $M(W)$) when evaluated in model $M(W')$ starting from timestep h. The term $d^W_{W',h}$ denotes the state-action distribution induced by policy π_W at timestep h in the MDP $M(W')$. Furthermore, we will write $V^W_{M^*,h}$ and $d^W_{M^*,h}$ when the evaluation environment is M^*. For conciseness, $V_{W,h}$ and $Q_{W,h}$ will denote the optimal (state- and Q-) value functions in model $M(W)$ at step h (e.g., $V_{W,h}(s) \equiv V^W_{W,h}(s)$). The expected return of a policy π in model $M(W)$ is defined as:

$$v^\pi_W = \mathbb{E}_{s \sim P_0}\left[V^\pi_{M(W),0}(s)\right]. \tag{4.2}$$

We assume that the total reward $\sum_{h=0}^{H-1} r_h$ lies in $[0,1]$ almost surely in all MDPs of interest and under all policies. Further, whenever used, any value function at step H (e.g., $V^{\pi}_{W,H}$) evaluates to 0 for any policy and any model.

4.3 Algorithm and Main Result

In this section we introduce the main algorithm that learns a near-optimal policy in the aforementioned setup with a $\mathrm{poly}(d, K, H, 1/\epsilon, \log(1/\delta))$ sample complexity. We will first give the intuition behind the algorithm, and then present the formal sample complexity guarantees. The complete detailed analysis is deferred to Section 4.9. For simplicity, we will describe the intuition for the realizable case with $\theta = 0$ ($P^* \equiv P^{W^*}$). The pseudocode (Algorithm 4.1) and the results are, however, stated for the general case of $\theta \neq 0$.

At a high level, our algorithm proceeds in iterations $t = 1, 2, \ldots$, and gradually refines a *version space* $\mathcal{W}_t$ of plausible parameters. Our algorithm follows an *explore-or-terminate* template and in each iteration, either chooses to explore with a carefully chosen policy or terminates with a near-optimal policy. For exploration in the t-th iteration, we collect n trajectories $\left\{\left(s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}, \ldots, s_{H-1}^{(i)}, a_{H-1}^{(i)}, r_{H-1}^{(i)}\right)\right\}_{i\in[n]}$ following some exploration policy π_t (line 7). A key component of the algorithm is to extract knowledge about W^* from these trajectories. In particular, for every h, the bag of samples $\left\{s_{h+1}^{(i)}\right\}_{i\in[n]}$ may be viewed as an unbiased draw from the following distribution

$$\frac{1}{n}\sum_{i=1}^{n} P^{W^*}\left(\cdot|s_h^{(i)}, a_h^{(i)}\right). \tag{4.7}$$

The situation for rewards is similar and will be omitted in the discussion. So in principle we could substitute W^* in (4.7) with any candidate W, and if the resulting distribution differs significantly from the real samples $\left\{s_{h+1}^{(i)}\right\}_{h\in[H],i\in[n]}$, we can assert that $W \neq W^*$ and eliminate W from the version space. However, the state space can be arbitrarily large in our setting, and comparing state distributions directly can be intractable. Instead, we project the state distribution in (4.7) using a (non-stationary) discriminator function $\{f_{t,h}\}_{h=0}^{H-1}$ (which will be chosen later) and consider the following scalar property

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1} \mathbb{E}_{\substack{r\sim R^{W^*}\left(\cdot|s_h^{(i)},a_h^{(i)}\right),\\ s'\sim P^{W^*}\left(\cdot|s_h^{(i)},a_h^{(i)}\right)}}\left[r + f_{t,h+1}(s')\right], \tag{4.8}$$

Algorithm 4.1 PAC Algorithm for Linear Model Ensembles

Input: $\{M_1, \dots M_K\}, \epsilon, \delta, \phi(\cdot,\cdot), \mathcal{W}_0$

for $t \to 1, 2, \dots$ **do**

Compute *optimistic model* W_t and set π_t to π_{W_t}

$$W_t \leftarrow \underset{W \in \mathcal{W}_{t-1}}{\operatorname{argmax}} V_W$$

Estimate the value of π_t using n_{eval} trajectories:

$$\hat{v}_t \coloneqq \frac{1}{n_{\text{eval}}} \sum_{h=0}^{H-1} r_h^{(i)} \tag{4.3}$$

if $v_{W_t} - \hat{v}_t \leq 3\epsilon/4 + (3\sqrt{dK}+1)H\theta$ **then**

Terminate and output π_t

Collect n trajectories using $\pi_t : a_h \sim \pi_t(s_h)$

Estimate the matrix $\widehat{Z}_t$ and $\hat{y}_t$ as

$$\widehat{Z}_t \coloneqq \frac{1}{n} \sum_{i=1}^{n} \sum_{h=0}^{H-1} \overline{V}_{t,h}\left(s_h^{(i)}, a_h^{(i)}\right) \phi\left(s_h^{(i)}, a_h^{(i)}\right)^\top \tag{4.4}$$

$$\hat{y}_t \coloneqq \frac{1}{n} \sum_{i=1}^{n} \sum_{h=0}^{H-1} r_{h+1}^{(i)} + V_{t,h+1}\left(s_{h+1}^{(i)}\right) \tag{4.5}$$

Update the version space to $\mathcal{W}_t$ as the set:

$$\left\{ W \in \mathcal{W}_{t-1} : \left| \hat{y}_t - \left\langle W, \widehat{Z}_t \right\rangle \right| \leq \tfrac{\epsilon}{12\sqrt{dK}} + H\theta \right\} \tag{4.6}$$

which can be effectively estimated by

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{h=0}^{H-1} \left(r_h^{(i)} + f_{t,h+1}\left(s_{h+1}^{(i)}\right) \right). \tag{4.9}$$

Since we have projected states onto $\mathbb{R}$, (4.9) is the average of scalar random variables and enjoys state-space-independent concentration. Now, in order to test the validity of a parameter W in a given version space, we compare the estimate in (4.9) with the prediction given by $M(W)$, which is:

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{h=0}^{H-1} \mathbb{E}_{\substack{r \sim R^W\left(\cdot | s_h^{(i)}, a_h^{(i)}\right), \\ s' \sim P^W\left(\cdot | s_h^{(i)}, a_h^{(i)}\right)}} \left[r + f_{t,h+1}(s') \right]. \tag{4.10}$$

As we consider a linear model class, by using linearity of expectations, (4.10) may also be written as:

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\left[W\phi\left(s_h^{(i)},a_h^{(i)}\right)\right]^\top\left[\overline{V}_{t,h}\left(s_h^{(i)},a_h^{(i)}\right)\right]\\
&=\left\langle W,\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\overline{V}_{t,h}\left(s_h^{(i)},a_h^{(i)}\right)\,\phi\left(s_h^{(i)},a_h^{(i)}\right)^\top\right\rangle,
\end{aligned}\tag{4.11}$$

where $\langle A,B\rangle$ denotes $\mathrm{tr}(A^\top B)$ for any two matrices A and B. In (4.11), $\overline{V}_{t,h}$ is a function that maps (s,a) to a K dimensional vector with each entry being

$$\left[\overline{V}_{t,h}(s,a)\right]_k \coloneqq \mathbb{E}_{\substack{r\sim R^k(\cdot|s,a),\\ s'\sim P^k(\cdot|s,a)}}\left[r+f_{t,h+1}(s')\right].\tag{4.12}$$

The intuition behind (4.11) is that for each fixed state-action pair $\left(s_h^{(i)},a_h^{(i)}\right)$, the expectation in (4.8) can be computed by first taking expectation of $r+f_{t,h+1}(s')$ over the reward and transition distributions of each of the K base models—which gives $\overline{V}_h$—and then aggregating the results using the combination coefficients. Rewriting lhs of (4.11) as its rhs, we see that (4.8) can also be viewed as a linear measurement of W^*, where the measurement matrix is again $\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\overline{V}_h\left(s_h^{(i)},a_h^{(i)}\right)\,\phi\left(s_h^{(i)},a_h^{(i)}\right)^\top$. Therefore, by estimating this measurement matrix and the outcome in (4.9), we obtain an approximate linear equality constraint over $\mathcal{W}_{t-1}$ and can eliminate any candidate W that violates such constraints. By using a finite sample concentration bound over the inner product, we get a linear inequality constraint to update the version space (see (4.6)).

The remaining concern is to choose the exploration policy π_t and the discriminator function $\{f_{t,h}\}$ to ensure that the linear constraint induced in each iteration is significantly different from the previous ones and induces deep cuts in the version space. We guarantee this by choosing $\pi_t \coloneqq \pi_{W_t}$ and $f_{t,h} \coloneqq V_{W_t,h}$[2], where W_t is the *optimistic model* as computed in line 3. That is, W_t predicts the highest optimal value among all candidate models in $\mathcal{W}_{t-1}$. Following a terminate-or-explore argument, we show that as long as π_t is suboptimal, the linear constraint induced by our choice of π_t and $\{f_{t,h}\}$ will significantly reduce the volume of the version space, and the iteration complexity can be bounded as $\mathrm{poly}(d,K)$ by an ellipsoid argument similar to that of Jiang et al. (2017). Similarly, the sample size needed in each iteration only depends polynomially on d and K and incurs no dependence on $|\mathcal{S}|$ or $|\mathcal{A}|$, as we have summarized high-dimensional objects such as $f_{t,h}$ (function over states) using low-dimensional quantities such as $\overline{V}_{t,h}$ (vector of length K).

The bound on the number of iterations and the number of samples needed per iteration leads to

[2]We use the simplified notation $V_{t,h}$ for $V_{W_t,h}$.

the following sample complexity result:

Theorem 4.1 (PAC bound for Algorithm 4.1)**.** *In Algorithm 4.1, if* $n_{\text{eval}} := \frac{32H^2}{\epsilon^2} \log \frac{4T}{\delta}$ *and* $n = \frac{1800d^2KH^2}{\epsilon^2} \log \frac{8dKT}{\delta}$ *where* $T = dK \log \frac{2\sqrt{2K}H}{\epsilon} / \log \frac{5}{3}$*, with probability at least* $1 - \delta$*, the algorithm terminates after using at most*

$$\widetilde{O}\left(\frac{d^3K^2H^2}{\epsilon^2} \log \frac{1}{\delta}\right) \tag{4.13}$$

trajectories and returns a policy π_T *with a value* $v^T \geq v^* - \epsilon - (3\sqrt{dK} + 2)H\theta$*.*

By setting d and K to appropriate values, we obtain the following sample complexity bounds as corollaries:

Corollary 4.2 (Sample complexity for partitions)**.** *Since the state-action partitioning setting (Example 4.2) is subsumed by the general setup, the sample complexity is again:*

$$\widetilde{O}\left(\frac{d^3K^2H^2}{\epsilon^2} \log \frac{1}{\delta}\right) \tag{4.14}$$

Corollary 4.3 (Sample complexity for global convex combination)**.** *When base models are combined without any dependence on state-action features (Example 4.1), the setting is special case of the general setup with* $d = 1$*. Thus, the sample complexity is:*

$$\widetilde{O}\left(\frac{K^2H^2}{\epsilon^2} \log \frac{1}{\delta}\right) \tag{4.15}$$

Our algorithm, therefore, satisfies the requirement of learning a near-optimal policy without any dependence on the $|\mathcal{S}|$ or $|\mathcal{A}|$. Moreover, we can also account for the approximation error θ but also incur a cost of $(3\sqrt{dK} + 1)H\theta$ in the performance guarantee of the final policy. As we use the projection of value functions through the linear model class, we do not model the complete dynamics of the environment. This leads to an additive loss of $3\sqrt{dK}H\theta$ in value in addition to the best achievable value loss of $2H\theta$ (see Corollary 4.8 in Section 4.9).

Optimality of the bound. Our main result in Theorem 4.1 shows an upper bound of $\widetilde{O}\left(\frac{d^3K^2}{\epsilon^2}\right)$ where the $1/\epsilon^2$ is the expected dependence similar to other minimax-optimal bounds. For the dimension dependence, compared to a supervised learning based upper bound of $O(dK)$, we incur additional factors of $O(dK)$ due to the exploration setting ($O(d)$ vs. $O(d^2)$ in linear bandit problems) and an additional $O(d)$ term due to the size of the parameter W^*.

Comparison to OLIME (Sun et al., 2019) Our Algorithm 4.1 shares some structural similarity with the OLIME algorithm proposed by Sun et al. (2019), but there are also several important differences. First of all, OLIME in each iteration will pick a time step and take uniformly random actions during data collection, and consequently incur polynomial dependence on $|\mathcal{A}|$ in its sample complexity. In comparison, our main data collection step (line 7) never takes a random deviation, and we do not pay any dependence on the cardinality of the action space. Secondly, similar to how we project the transition distributions onto a discriminator function (see (4.7) and (4.8)), OLIME projects the distributions onto a *static discriminator class* and uses the corresponding integral probability metric (IPM) as a measure of model misfit. In our setting, however, we find that the most efficient and elegant way to extract knowledge from data is to use a *dynamic* discriminator function, $V_{W_t,h}$, which changes from iteration to iteration and depends on the previously collected data. Such a choice of discriminator function allows us to make direct cuts on the parameter space $\mathcal{W}$, whereas OLIME can only make cuts in the value prediction space.

Computational Characteristics In each iteration, our algorithm computes the optimistic policy within the version space. Therefore, we rely on access to the following *optimistic planning oracle*:

Assumption 4.1 (Optimistic planning oracle)**.** *We assume that when given a version space $\mathcal{W}_t$, we can obtain the optimistic model through a single oracle call for $W_t = \operatorname{argmax}_{W \in \mathcal{W}_t} V_W$.*

It is important to note that any version space $\mathcal{W}_t$ that we deal with is always an intersection of half-spaces induced by the linear inequality constraints. Therefore, one would hope to solve the optimistic planning problem in a computationally efficient manner given the nice geometrical form of the version space. However, even for a finite state-action space, we are not aware of any efficient solutions as the planning problem induces bilinear and non-convex constraints despite the linearity assumption. Many recently proposed algorithms also suffer from such a computational difficulty (Jiang et al., 2017; Dann et al., 2018; Sun et al., 2019).

Further, we also assume that for any given W, we can compute the optimal policy π_W and its value function: our elimination criteria in (4.6) uses estimates $\widehat{Z}_t$ and $\hat{y}_t$ which in turn depend on the value function. This requirement corresponds to a standard planning oracle, and aligns with the motivation of our setting, as we can delegate these computations to any learning algorithm operating in the simulated environment with the given combination coefficient. Our algorithm, instead, focuses on careful and systematic exploration to minimize the sample complexity in the real world.

4.4 Feature Selection for Linear Model Ensembles

In the previous section we showed that a near-optimal policy can be PAC-learned under our modeling assumptions, where the feature map $\phi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is given along with the approximation error θ. In this section, we explore the more interesting and challenging setting where a realizable feature map ϕ is unknown, but we know that the realizable ϕ belongs to a candidate set $\{\phi_i\}_{i=1}^N$, i.e., the true environment satisfies our modeling assumption in Definition 4.1 under $\phi = \phi_{i^*}$ for some $i^* \in [N]$ with $\theta_{i^*} = 0$. Note that Definition 4.1 may be satisfied by multiple ϕ_i's; for example, adding redundant features to an already realizable ϕ_{i^*} still yields a realizable feature map. In such cases, we consider ϕ_{i^*} to be the most succinct feature map among all realizable ones, i.e., the one with the lowest dimensionality. Let d_i denote the dimensionality of ϕ_i, and $d^* = d_{i^*}$.

One obvious baseline in this setup is to run Algorithm 4.1 with each ϕ_i and select the best policy among the returned ones. This leads to a sample complexity of roughly $\sum_{i=1}^N d_i^3$ (only the dependence on $\{d_i\}_{i=1}^N$ is considered), which can be very inefficient: When there exists j such that $d^* \ll d_j$, we pay for d_j^3 which is much greater than the sample complexity of d^*; When $\{d_i\}$ are relatively uniform, we pay a linear dependence on N, preventing us from competing with a large set of candidate feature maps.

So the key result we want to obtain is a sample complexity that scales as $(d^*)^3$, possibly with a mild multiplicative overhead dependence on d^* and/or N (e.g., $\log d^*$ and $\log N$).

4.4.1 Hardness result for unstructured partitions

Unfortunately, we show that this is impossible when $\{\phi_i\}$ is unstructured via a lower bound. In the lower bound construction, we have an exponentially large set of candidate feature maps, all of which are state space partitions. Each of the partitions has trivial dimensionalities ($d_i = 2$, $K = 2$), but the sample complexity of learning is exponential, which can only be explained away as $\Omega(N)$.

Proposition 4.1. *For the aforementioned problem of learning an ϵ-optimal policy using a candidate feature set of size N, no algorithm can achieve* $\mathrm{poly}(d^*, K, H, 1/\epsilon, 1/\delta, N^{1-\alpha})$ *sample complexity for any constant* $0 < \alpha < 1$.

On a separate note, besides providing formal justification for the structural assumption we will introduce later, this proposition is of independent interest as it also sheds light on the hardness of model selection with state abstractions. We discuss the further implications in Section 4.5.

Proof of Proposition 4.1. We construct a linear class of MDPs with two base models M_1 and M_2 in the following way: Consider a complete tree of depth H with a branching factor of 2. The vertices forming the state space of M_1 and M_2 and the two outgoing edges in each state are the available

actions. Both MDPs share the same deterministic transitions and each non-leaf node yields 0 reward. Every leaf node yields $+1$ reward in M_1 and 0 in M_2. Now we construct a candidate partition set $\{\phi_i\}$ of size 2^H: for ϕ_i, the i-th leaf node belongs to one equivalence class while all other leaf nodes belong to the other. (Non-leaf nodes can belong to either class as M_1 and M_2 agree on their transitions and rewards.)

Observe that the above model class contains a finite family of 2^H MDPs, each of which only has 1 rewarding leaf node. Concretely, the MDP whose i-th leaf is rewarding is exactly realized under the feature map ϕ_i, whose corresponding W^* is the identity matrix: the i-th leaf yields $+1$ reward as in M_1, and all other leaves yield 0 reward as in M_2. Learning in this family of 2^H MDPs is provably hard (Krishnamurthy et al., 2016), as when the rewarding leaf is chosen adversarially, the learner has no choice but to visit almost all leaf nodes to identify the rewarding leaf as long as ϵ is below a constant threshold. The proposition follows from the fact that in this setting $d^* = 2$, $K = 2$, $1/\epsilon$ is a constant, $N = 2^H$, but the sample complexity is $\Omega(2^H)$. □

This lower bound shows the necessity of introducing structural assumptions in $\{\phi_i\}$. Below, we consider a particular structure of *nested partitions* that is natural and enables sample-efficient learning. Similar assumptions have also been considered in the state abstraction literature (e.g., Jiang et al., 2015).

4.4.2 Nested partitions as a structural assumption

Consider the case where every ϕ_i is a partition. W.l.o.g. let $d_1 \leq d_2 \leq \ldots \leq d_N$. We assume $\{\phi_i\}$ is nested, meaning that $\forall (s,a),(s',a')$,

$$\phi_i(s,a) = \phi_i(s',a') \implies \phi_j(s,a) = \phi_j(s',a'),\ \forall i \leq j.$$

While this structural assumption almost allows us to develop sample-efficient algorithms, it is still insufficient as demonstrated by the following hardness result.

Proposition 4.2. *Fixing $K = 2$, there exist base models M_1 and M_2 and* nested *state space partitions ϕ_1 and ϕ_2, such that it is information-theoretically impossible for any algorithm to obtain* $\mathrm{poly}\,(d^*, H, K, 1/\epsilon, 1/\delta)$ *sample complexity when an adversary chooses an MDP that satisfies our environmental assumption (Definition 4.1) under either ϕ_1 or ϕ_2.*

Proof. We will again use an exponential tree style construction to prove the lower bound. Specifically, we construct two MDPs M and M' which are obtained by combining two base MDPs M_1 and M_2 using two different partitions ϕ_1 and ϕ_2. The specification of M_1 and M_2 is exactly the same as in the proof of Proposition 4.1. We choose ϕ_1 to be a partition of size $d_1 = 1$, where all nodes

are grouped together. ϕ_2 has size $d_2 = 2^H$, where each leaf node belongs to a separate group. (As before, which group the inner nodes belong to does not matter.) ϕ_1 and ϕ_2 are obviously nested. We construct M that is realizable under ϕ_2 by randomly choosing a leaf and setting the weights for the convex combination as $(1/2 + 2\epsilon, 1/2 - 2\epsilon)$ for that leaf; for all other leafs, the weights are $(1/2, 1/2)$. This is equivalent to randomly choosing M from a set of 2^H MDPs, each of which has only one *good* leaf node yielding a random reward drawn from $\mathrm{Ber}(1/2 + 2\epsilon)$ instead of $\mathrm{Ber}(1/2)$. In contrast, M' is such that all leaf nodes yield $\mathrm{Ber}(1/2)$ reward, which is realizable under ϕ_1 with weights $(1/2, 1/2)$.

Observe that M and M' are exactly the same as the constructions in the proof of the multi-armed bandit lower bound by (Auer et al., 2002) (the number of arms is 2^H), where it has been shown that distinguishing between M and M' takes $\Omega(2^H/\epsilon^2)$ samples. Now assume towards contradiction that there exists an algorithm that achieves $\mathrm{poly}\,(d^*, H, K, 1/\epsilon, 1/\delta)$ complexity; let f be the specific polynomial in its guarantee. After $f(1, H, 2, 1/\epsilon, 1/\delta)$ trajectories are collected, the algorithm must stop if the true environment is M' to honor the sample complexity guarantee (since $d^* = 1$, $K = 2$), and proceed to collect more trajectories if M is the true environment (since $d^* = 2^H$). Making this decision essentially requires distinguishing between M' and M using $f(1, H, 2, 1/\epsilon, 1/\delta) = \mathrm{poly}(H)$ trajectories, which contradicts the known hardness result from Auer et al. (2002). This proves the statement. □

Essentially, the lower bound creates a situation where $d_1 \ll d_2$, and the nature may adversarially choose a model such that either ϕ_1 or ϕ_2 is realizable. If ϕ_1 is realizable, the learner is only allowed a small sample budget and cannot fully explore with ϕ_2, and if ϕ_1 is not realizable the learner must do the opposite. The information-theoretic lower bound shows that it is fundamentally hard to distinguish between the two situations: Once the learner explores with ϕ_1, she cannot decide whether she should stop or move on to ϕ_2 without collecting a large amount of data.

This hardness result motivates our last assumption in this section, that the learner knows the value of $v^\star$ (a scalar) as side information. This way, the learner can compare the value of the returned policy in each round to $v^\star$ and effectively decide when to stop. This naturally leads to our Algorithm 4.2 that uses a doubling scheme over $\{d_i\}$, with the following sample complexity guarantee.

Theorem 4.4. *When Algorithm 4.2 is run with the input v^*, with probability at least $1 - \delta$, it returns a near-optimal policy π with $v^\pi \geq v^* - \epsilon$ using at most $\widetilde{O}\left(\frac{d^{*3}K^2H^2}{\epsilon^2} \log d^* \log \frac{N}{\delta}\right)$ samples.*

Proof. In Algorithm 4.2, for each partition i, we run Algorithm 4.1 until termination or until the sample budget is exhausted. By union bound it is easy to verify that with probability at least $1 - \delta$, all calls to Algorithm 4.1 will succeed and the Monte-Carlo estimation of the returned policies will

Algorithm 4.2 Model Selection with Nested Partitions

Input: $\{\phi_1, \phi_2, \ldots, \phi_N\}$, $\{M_1, \ldots, M_K\}$, ϵ, δ, $v^\star$.
$i \to 0$
while `True` **do**
  Choose ϕ_i such that d_i is the largest among $\{d_j : d_j \leq 2^i\}$.
  Run Algorithm 4.1 on Φ_i with $\epsilon_i = \frac{\epsilon}{2}$ and $\delta_i = \frac{\delta}{2N}$.
  Terminate the sub-routine if $t > d_i K \log \frac{2\sqrt{2KH}}{\epsilon} / \log \frac{5}{3}$.
  Let π_i be the returned policy (if any). Let $\hat{v}_i$ be the estimated return of π_i using $n_{\text{eval}} = \frac{9}{2\epsilon^2} \log \frac{2N}{\delta}$ Monte-Carlo trajectories.
  if $\hat{v}_i \geq v^* - \frac{2\epsilon}{3}$ **then**
    Terminate with output π_i.

be $(\epsilon/3)$-accurate, and we will only consider this success event in the rest of the proof. When the partition under consideration is realizable, we get $v^{\pi_i}_{M^*} \geq v^* - \epsilon/3$, therefore

$$\hat{v}^i \geq v^{\pi_i} - \tfrac{\epsilon}{3} \geq v^* - \tfrac{2\epsilon}{3},$$

so the algorithm will terminate after considering a realizable ϕ_i. Similarly, whenever the algorithm terminates, we have $v^{\pi_i} \geq v^* - \epsilon$. This is because

$$v^{\pi_i} \geq \hat{v}^i - \tfrac{\epsilon}{3} \geq v^* - \epsilon,$$

where the last inequality holds thanks to the termination condition of Algorithm 4.2, which relies on knowledge of $v^\star$. The total number of iterations of the algorithm is at most $O(\log d^*)$. Therefore, by taking a union bound over all possible iterations, the sample complexity is

$$\sum_{i=1}^{J} \widetilde{O}\left(\frac{d_i^3 K^2 H^2}{\epsilon^2}\right) \leq \widetilde{O}\left(\frac{d^{*3} K^2 H^2}{\epsilon^2}\right). \qquad \square$$

Discussion. Model selection in online learning—especially in the context of sequential decision making—is generally considered very challenging. There has been relatively limited work in the generic setting until recently for some special cases. For instance, Foster et al. (2019) consider the model selection problem in linear contextual bandits with a sequence of nested policy classes with dimensions $d_1 < d_2 < \ldots$. They consider a similar goal of achieving sub-linear regret bounds which only scale with the optimal dimension d_{m^*}. In contrast to our result, they do not need to know the achievable value in the environment and give no-regret learning methods in the *knowledge-free* setting. However, this is not contradictory to our lower bound: Due to the extremely delayed reward signal, our construction is equivalent to a multi-armed bandit problem with 2^H arms. Our negative result (Proposition 4.2) shows a lower bound on sample complexity which is exponential in horizon,

therefore eliminating the possibility of sample efficient and knowledge-free model selection in MDPs.

4.5 The Implication of Proposition 4.1 on the Hardness of Learning State Abstractions

Here we show that the proof of Proposition 4.1 can be adapted to show a related hardness result for learning with state abstractions. A state abstraction is a mapping ϕ that maps the raw state space $\mathcal{S}$ to some finite abstract state space $\mathcal{S}_\phi$, typically much smaller in size. When an abstraction ϕ with good properties (e.g., preserving reward and transition dynamics) is known, one can leverage it in exploration and obtain a sample complexity that is polynomial in $|\mathcal{S}_\phi|$ instead of $|\mathcal{S}|$. Among different types of abstractions, *bisimulation* (Whitt, 1978; Givan et al., 2003) is a very strict notion that comes with many nice properties (Li et al., 2006).

An open problem in state abstraction literature has been whether it is possible to perform model selection over a large set of candidate abstractions, i.e., designing an algorithm whose sample complexity only scales sublinearly (or ideally, logarithmically) with the cardinality of the candidate set. Using the construction from Proposition 4.1, we show that this is impossible without further assumptions:

Proposition 4.3. *Consider a learner in an MDP that is equipped with a set of state abstractions, $\{\phi_1, \phi_2, \ldots, \phi_N\}$. Each abstraction ϕ_i maps the raw state space $\mathcal{S}$ to a finite abstract state space $\mathcal{S}_{\phi_i}$. Even if there exists $i^* \in [N]$ such that $\phi^* = \phi_i$ is a bisimulation, no algorithm can achieve* $\mathrm{poly}\left(|\mathcal{S}_{\phi^*}|, |\mathcal{A}|, H, 1/\epsilon, 1/\delta, N^{1-\alpha}\right)$ *sample complexity for any $\alpha > 0$.*

Proof. Following the proof of Proposition 4.1, we consider the family of MDPs that share the same deterministic transition dynamics with a complete tree structure, where each MDP only has one rewarding leaf. Let $N = 2^H$ be the number of leaves, and let the MDPs be $\{M_i\}$ where the index indicates the rewarding leaf. We then construct a set of N abstractions where one of them will always be a bisimulation regardless of which MDP we choose from the family. Consider the i-th abstraction, ϕ_i. At each level h, ϕ_i aggregates the state on the optimal path in its own equivalence class, and aggregates all other states together. It does not aggregate states across levels. It is easy to verify that ϕ_i is a bisimulation in MDP M_i, and $|\mathcal{S}_{\phi_i}| = 2H$. If the hypothetical algorithm existed, it would achieve a sample complexity sublinear in N, as $N = 2^H$ and all other relevant parameters (e.g., $|\mathcal{S}_{\phi_i}|$ and H) are at most polynomial in H. However, the set of abstractions $\{\phi_i\}$ is uninformative in this specific problem and does not affect the $\Omega(2^H)$ sample complexity, which completes the proof. □

4.6 Related Work

Related structural assumptions In this chapter, we studied how a linear mixture assumption in the model space can be utilized to design provably efficient exploration algorithms. As a follow-up to our work, the structural assumption in Definition 4.1 has been studied extensively under the name 'linear-mixture MDP'. Firstly, the algorithmic scheme we propose in Algorithm 4.1 has been generalized under the regret framework in Ayoub et al. (2020). The authors show that an optimistic value targeted regression algorithm, UCRL-VTR, guarantees near-optimal regret for model classes with bounded Eluder dimension (which includes Definition 4.1). Further, extensions of results to different settings (Zhou et al., 2021b) and improvements have also been proposed (Zhou et al., 2021a).

A closely related structured model is the class of low-rank MDPs, have been considered by Yang and Wang (2019); Jin et al. (2020). As described in Definition 2.2, in low-rank MDPs, the transition matrices admit a low-rank factorization, and the left matrix in the factorization are known to the learner as state-action features (corresponding to our ϕ). Their environmental assumption is a special case of ours, where the transition dynamics of each base model $P^k(\cdot|s,a)$ is *independent* of s and a, i.e., each base MDP can be fully specified by a single density distribution over $\mathcal{S}$. This special case enjoys many nice properties, such as the value function of any policy is also linear in state-action features, and the linear value-function class is closed under the Bellman update operators, which are heavily exploited in their algorithms and analyses. In contrast, none of these properties hold under our more general setup, yet we are still able to provide sample efficiency guarantees. That said, we do note that the special case allows these recent works to obtain stronger results: their algorithms are both statistically and computationally efficient (ours is only statistically efficient), and some of these algorithms work without knowing the K base distributions.[3]

Model ensembles in practice On the empirical side, the closest work to our setting is the multiple model-based RL (MMRL) architecture proposed by Doya et al. (2002) where they also decompose a given domain as a convex combination of multiple models. However, instead of learning the combination coefficients for a given ensemble, their method trains the model ensemble and simultaneously learns a mixture weight for each *base model* as a function of state features. Their experiments demonstrate that each model specialized for different domains of the state space where the environment dynamics is predictable, thereby, providing a justification for using convex combination of models for simulation. Further, the idea of combining different models is inherently present in Bayesian learning methods where a posterior approximation of the real environment is

[3] In our setting, not knowing the base models immediately leads to hardness of learning, as it is equivalent to learning a general MDP without any prior knowledge even when $d = K = 1$. This requires $\Omega(|\mathcal{S}||\mathcal{A}|)$ sample complexity (Azar et al., 2012), which is vacuous as we are interested in solving problems with arbitrarily large state and action spaces.

iteratively refined using interaction data. For instance, Rajeswaran et al. (2017) introduce the EPOpt algorithm which uses an ensemble of simulated domains to learn robust and generalizable policies. During learning, they adapt the ensemble distribution (convex combination) over source domains using data from the target domain to progressively make it a better approximation. Similarly, Lee et al. (2019) combine a set of parameterized models by adaptively refining the mixture distribution over the latent parameter space. Here, we study a relatively simpler setting where a finite number of such base models are combined and give a frequentist sample complexity analysis for our method.

Results on model-selection The problem of model selection in the bandit literature has received a lot of recent attention, in a range of results from linear to general stochastic contextual bandits (Agarwal et al., 2017; Foster et al., 2019; Pacchiano et al., 2020b; Bibaut et al., 2020; Pacchiano et al., 2020a; Arora et al., 2021). In many such algorithms, the idea is to run a master algorithm which manages different instances of a base algorithm with the different hypothesis classes. The model selection subroutine uses the sample efficiency guarantees for the base algorithm as a hypothesis test and eliminates bad hypothesis classes. On the other hand, for reinforcement learning, devising model selection algorithms is a problem of current interest and hasn't seen much work. In one instance, Pacchiano et al. (2020a) show regret guarantees for learning in linear MDPs with a given set of features classes. However, the final result depends polynomially on the number of candidate features which was not the desired guarantee in Section 4.4. Motivated by this, in the next chapter, we consider a representation learning problem for linear MDPs and show a sample complexity bound which depends logarithmicaly on the size of the candidate feature class.

Computational efficiency Lastly, in this chapter, we study the statistical efficiency of learning under a linear combination assumption for arbitrarily large MDPs. We, however, disregard the computational complexity of learning in such MDPs by using an oracle assumption in Assumption 4.1. For the regret criteria, the algorithm proposed by Zhou et al. (2021a) uses elliptic potential based bonuses in addition to least-squares value iteration as the algorithmic template. Similar procedures can be used to devise PAC-efficient algorithms via a regret to PAC conversion method described in Section 2.2.3.

4.7 Discussion

The main motivation considered in this was the use of multiple simulators of varying fidelity for simulation purposes. In our structural model, we studied the case where a linear combination of such models can be used to learn a more-accurate model. Although the main results in this chapter

are stylized for the linear structure, there are various fundamental insights which potentially carry over to more general cases:

Sufficiency of small Bellman error for exploration In our exploration scheme in Algorithm 4.1, our main idea is to use the error in predicting Bellman backups of a sequence of discriminator functions as constraints for updating the version space of plausible models. This has two key benefits which can be used in practice as well: (1) instead of learning a model which has small error in the next-state distributions, we can easily restrict ourselves to models which have small prediction error for value functions, hence, leading to a regression based model learning objective compared to likelihood-based objectives and (2) the set of these value functions can be chosen optimistically by choosing the optimal value functions of the best model in the current version space. Using a value function based model learning objective also allows us to combine hybrid approaches where both model and value function approximation is used. Specifically, this can be seen as an instance of value-aware model learning (Farahmand et al., 2017) where the objective is the Bellman error for value functions in a given class. For provable correctness of our algorithmic setup, we need to solve the optimistic planning problem which is computationally hard and we assume access to an oracle solution. However, in practice, such problems can be approximately solved by look-ahead based planning and tree-based search like MCTS with optimistic model selection on each branching step. For one step, the problem can be reduced to a simpler constrained optimisation objective which is linear in our case. Further, instead of maintaining all sets of value functions and optimistic policies, we can simply the algorithmic setup by using a replay buffer as is typical in most deep reinforcement learning algorithms. Lastly, it will be an interesting question for numerical simulations when multiple models are also learnt as described below.

Learning with model ensembles As described in the previous related work section, model ensembles are used in practice in different forms. The cited works of Doya et al. (2002), Rajeswaran et al. (2017) and Lee et al. (2019) use model ensembles for robust and generalizable RL. These techniques are typically motivated from a Bayesian perspective whereas we take a frequentist route via our linear structural assumption. However, our main techniques can also be explored in their setting by using a setup similar to other bootstrap sampling techniques Osband et al. (2016). Specifically, randomization in the training data, learning rates, internal algorithmic choices can be used to learn multiple models and then later be combined using a feature based linear combination. This will increase the overall expressivity of the parameterized model where the base models can be trained at a slower timescale (or frozen after learning a coarse model) with faster updates for the last linear combination coefficients. In general, we anticipate that using an optimistic combination of models can be much more efficient for exploration as compared to randomized or bootstrap

schemes.

Model selection The final part of this chapter considers a feature selection problem and highlights a challenging and intriguing nature of RL problems. In the hardness results we have shown, the construction outlines a setting where the agent has to identify whether the current environment is a simple environment (hence a structurally simpler policy/model/value function) or a complex one (hence complex policy/model/value function). However, this aspect of identification forces the agent to test for the possibility of a complex environment as well, thereby, increasing the necessary sample size. This is in direct contrast with the supervised learning setting as learning with iid data enables us to directly evaluate performance of any learner using empirical risk, unlike RL. Our sample efficient solution requires additional prior knowledge for such a test but further work is required to address this challenging and important question under weaker assumptions.

4.8 Summary

In this chapter, we proposed a sample efficient model based algorithm which learns a near-optimal policy by approximating the true environment via a feature dependent convex combination of a given ensemble. Our algorithm offers a sample complexity bound which is independent of the size of the environment and only depends on the number of parameters being learnt. In addition, we also consider a model selection problem, show exponential lower bounds and then give sample efficient methods under natural assumptions. The proposed algorithm and its analysis relies on a linearity assumption and shares this aspect with existing exploration methods for rich observation MDPs. Lastly, our work also revisits the open problem of coming up with a computational and sample efficient model based learning algorithm.

4.9 Proof of Main Result

In this section, we provide a detailed proof and the key ideas used in the analysis. The proof uses an optimism based template which guarantees that either the algorithm terminates with a near-optimal policy or explores in the environment. We can show a polynomial sample complexity bound as the algorithm explores for a bounded number of iterations and the number of samples required in each iteration is polynomial in the desired parameters. We start with the key lemmas used in the analysis in Section 4.9.1 with the final proof of the main theorem in Section 4.9.2.

Notation. As in the main text, we use $\langle X, Y \rangle$ for $\mathrm{tr}(X^\top Y)$. The notation $\|A\|_F$ denotes the Frobenius norm $\mathrm{tr}(A^\top A)$. For any matrix $A = (A^1 A^2 \dots A^n)$ in $\mathbb{R}^{m \times n}$ with columns $A^i \in \mathbb{R}^m$, we

will use $\|A\|_{p,q}$ as the group norm: $\left\|(\|A^1\|_p, \|A^2\|_p, \ldots, \|A^n\|_p)\right\|_q$.

4.9.1 Key lemmas used in the analysis

For our analysis, we first define a term $\mathcal{E}(W,h)$ for any parameter W which intuitively quantifies the model error at step h as follows:

$$\mathbb{E}_{d^W_{M^*,h}}\left[\mathbb{E}_{M(W)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] - \mathbb{E}_{M^*}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right]\right] \tag{4.16}$$

We start with the following lemma which allows us to express the value loss by using a model $M(W)$ in terms of these per-step quantities.

Lemma 4.5 (Value decomposition)**.** *For any $W \in \mathcal{W}$, we can write the difference in two values:*

$$v_W - v^W_{M^*} = \mathcal{E}(W) \coloneqq \sum_{h=0}^{H-1} \mathcal{E}(W,h) \tag{4.17}$$

Proof. We start with the value difference on the lhs:

$$\begin{aligned}
v_W - v^W_{M^*} &= \mathbb{E}_{d^W_{M^*,0}}\left[\mathbb{E}_{M(W)}\left[r_0 + V_{W,1}(s_1)\middle|s_0,a_0\right] - \mathbb{E}_{M^*}\left[r_0 + V^W_{M^*,1}(s_1)\middle|s_0,a_0\right]\right] \\
&= \mathbb{E}_{d^W_{M^*,0}}\left[\mathbb{E}_{M(W)}\left[r_0 + V_{W,1}(s_1)\middle|s_0,a_0\right]\right] \\
&\quad - \mathbb{E}_{d^W_{M^*,0}}\left[\mathbb{E}_{M^*}\left[r_0 + V_{W,1}(s_1) - V_{W,1}(s_1) + V^W_{M^*,1}(s_1)\middle|s_0,a_0\right]\right] \\
&= \mathbb{E}_{d^W_{M^*,0}}\left[\mathbb{E}_{M(W)}\left[r_0 + V_{W,1}(s_1)\middle|s_0,a_0\right] - \mathbb{E}_{M^*}\left[r_0 + V_{W,1}(s_1)\middle|s_0,a_0\right]\right] \\
&\quad + \mathbb{E}_{d^W_{M^*,1}}\left[V_{W,1}(s_1) - V^W_{M^*,1}(s_1)\right] \\
&= \mathcal{E}(W,0) + \mathbb{E}_{d^W_{M^*,2}}\left[\mathbb{E}_{M(W)}\left[r_1 + V_{W,2}(s_2)\middle|s_1,a_1\right] - \mathbb{E}_{M^*}\left[r_1 + V^W_{M^*,2}(s_2)\middle|s_1,a_1\right]\right]
\end{aligned}$$

Unrolling the second expected value similarly till H leads to the desired result. □

At various places in our analysis, we will use the well-known simulation lemma to compare the value of a policy π across two MDPs:

Lemma 4.6 (Simulation Lemma (Kearns and Singh, 2002; Modi et al., 2018))**.** *Let M_1 and M_2 be two MDPs with the same state-action space. If the transition dynamics and reward functions of the two MDPs are such that:*

$$\begin{aligned}
\left\|P^1(\cdot|s,a) - P^2(\cdot|s,a)\right\|_1 &\leq \epsilon_p \qquad \forall s \ \in \mathcal{S}, a \in \mathcal{A} \\
\left|R^1(\cdot|s,a) - R^2(\cdot|s,a)\right| &\leq \epsilon_r \qquad \forall s \ \in \mathcal{S}, a \in \mathcal{A}
\end{aligned}$$

then, for every policy π, we have:

$$\left|v^{\pi}_{M_1} - v^{\pi}_{M_2}\right| \leq H\epsilon_p + \epsilon_r \tag{4.18}$$

Now, we will first use the assumption about linearity to prove the following key lemma of our analysis:

Lemma 4.7 (Decomposition of $\mathcal{E}(W)$)**.** *If θ is the approximation error defined in* (4.1)*, then the quantity $\mathcal{E}(W)$ can be bounded as follows:*

$$\mathcal{E}(W) \leq \left\langle W - W^*, \sum_{h=0}^{H-1} \mathbb{E}_{d^W_{M^*,h}}\left[\overline{V}_{W,h}(s_h,a_h)\phi(s_h,a_h)^\top\right]\right\rangle + H\theta \tag{4.19}$$

where $\overline{V}_{W,h}(s_h,a_h) \in [0,1]^K$ is a vector with the k^{th} entry as $\mathbb{E}_{M_k}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right]$.

Proof. Using the definition of $\mathcal{E}(W,h)$ from (4.16), we rewrite the term as:

$$\begin{aligned}
\mathcal{E}(W) &= \sum_{h=0}^{H-1} \mathcal{E}(W,h) \\
&= \mathbb{E}_{d^W_{M^*,h}}\left[\mathbb{E}_{M(W)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] - \mathbb{E}_{M^*}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right]\right] \\
&= \sum_{h=0}^{H-1} \mathbb{E}_{d^W_{M^*,h}}\left[\mathbb{E}_{M(W)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] - \mathbb{E}_{M(W^*)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right]\right] \\
&\quad + \mathbb{E}_{d^W_{M^*,h}}\left[\mathbb{E}_{M(W^*)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] - \mathbb{E}_{M^*}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right]\right] \\
&\leq \sum_{h=0}^{H-1} \mathbb{E}_{d^W_{M^*,h}}\left[\mathbb{E}_{M(W)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] - \mathbb{E}_{M(W^*)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right]\right] \\
&\quad + \mathbb{E}_{d^W_{M^*,h}}[\theta]
\end{aligned}$$

Here, we rewrite the inner expectation as:

$$\begin{aligned}
\mathbb{E}_{M(W)}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] &= \sum_{k=1}^{K} \left(W\phi(s_h,a_h)\right)[k]\mathbb{E}_{M_k}\left[r_h + V_{W,h+1}(s_{h+1})\middle|s_h,a_h\right] \\
&= \mathbb{E}_{d^W_{M^*,h}}\left[\left\langle W, \phi(s_h,a_h), \overline{V}_{W,h}(s_h,a_h)\right\rangle\right] \\
&= \left\langle W, \mathbb{E}_{d^W_{M^*,h}}\left[\overline{V}_{W,h}(s_h,a_h)\phi(s_h,a_h)^\top\right]\right\rangle
\end{aligned}$$

where $\overline{V}_{W,h}(s_h,a_h) \in [0,1]^K$ is a vector with the k^{th} entry as $\mathbb{E}_{M_k}[r_h + V_{W,h+1}(s_{h+1})|s_h,a_h]$.

Therefore, we can finally upper bound $\mathcal{E}(W)$ by:

$$\mathcal{E}(W) \leq \left\langle W - W^*, \sum_{h=0}^{H-1} \mathbb{E}_{d^W_{M^*,h}} \left[\overline{V}_{W,h}(s_h, a_h) \phi(s_h, a_h)^\top \right] \right\rangle + H\theta$$

□

For conciseness, we use the notation $V_{t,h}$ for the vector $V_{W_t,h}$. We write the matrix $\sum_{h=0}^{H-1} \mathbb{E}_{d^W_{M^*,h}} \left[\overline{V}_{W,h}(s_h, a_h) \phi(s_h, a_h)^\top \right]$ as Z_W and further use Z_t for Z_{W_t} which results in the bound:

$$\mathcal{E}(W) \leq \langle W - W^*, Z_W \rangle + H\theta$$

Further, using Lemma 4.6, one can easily see the following result which we later use in Lemma 4.9:

Corollary 4.8. *For the true environment and the MDP $M(W^*)$, we have:*

$$\mathcal{E}(W^*) \leq \left| v_{W^*} - v^{W^*}_{M^*} \right| \leq H\theta \tag{4.20}$$

$$v^{W^*}_{M^*} \geq v^* - 2H\theta \tag{4.21}$$

Proof. Equation (4.20) directly follows through from the assumption and Lemma 4.6. For (4.21), we have:

$$\begin{aligned} v^{W^*}_{M^*} &\geq v_{W^*} - H\theta \\ &\geq v^{\pi^*}_{W^*} - H\theta \\ &\geq v^* - 2H\theta \end{aligned}$$

□

By Lemma 4.5, we see that if the model-misfit error is controlled at each timestep, we can directly get a bound on the value loss incurred by using the greedy policy π_W. In Algorithm 4.1, we choose the optimistic policy W_t as the exploration policy which has the following property:

Lemma 4.9 (Explore-or-terminate)**.** *If the estimate $\hat{v}_t$ from (4.3) satisfies the following inequality:*

$$\left| \hat{v}_t - v^{W_t}_{M^*} \right| \leq \tfrac{\epsilon}{4} \tag{4.22}$$

throughout the execution of the algorithm and W^* is not eliminated from any $\mathcal{W}_t$ (version space is valid), then either of these two statements hold:

(i) the algorithm terminates with output π_t such that $v_{W^*}^{\pi_t} \geq v^* - (3\sqrt{dK}+2)H\theta - \epsilon$

(ii) the algorithm does not terminate and

$$\mathcal{E}(W_t) \geq \frac{\epsilon}{2} + 3\sqrt{dK}H\theta + H\theta$$

Proof. If the algorithm doesn't terminate, then by the condition on line 5 and the assumption, we know that:

$$\mathcal{E}(W_t) = v_{W_t} - v_{W^*}^{W_t} \geq v_{W_t} - \hat{v}_t - \tfrac{\epsilon}{4} \geq \tfrac{\epsilon}{2} + 3\sqrt{dK}H\theta + H\theta$$

If the algorithm does terminate at step T, we have:

$$\begin{aligned}
v_{M^*}^{\pi_T} &\geq \hat{v}_t - \epsilon/4 && \text{(4.22)} \\
&\geq v_{W_T} - (3\sqrt{dK}+1)H\theta - \epsilon && \text{Algorithm 4.1, line 5)} \\
&\geq v_{W^*} - (3\sqrt{dK}+1)H\theta - \epsilon && \text{(Optimism)} \\
&\geq v^* - (3\sqrt{dK}+2)H\theta - \epsilon && \text{(Corollary 4.8)}
\end{aligned}$$

□

Lemma 4.9 shows that either the algorithm terminates with a (near-)optimal policy or guarantees large model-misfit error for W_t. For bounding the number of iterations, we use a volumetric argument similar to Jiang et al. (2017). We will use the following Lemma to show the exponential rate of reduction in the volume of the version space:

Lemma 4.10 (Volume reduction for MVEE, (Jiang et al., 2017))**.** *Consider a closed and bounded set $V \subset \mathbb{R}^p$ and a vector $a \in \mathbb{R}^p$. Let B be any enclosing ellipsoid of V that is centered at the origin, and we abuse the same symbol for the symmetric positive definite matrix that defines the ellipsoid, i.e., $B = \{v \in \mathbb{R}^p : v^\top B^{-1} v \leq 1\}$. Suppose there exists $u \in V$ with $|a^\top u| \geq \kappa$ and define B_+ as the minimum volume enclosing ellipsoid of $\{v \in B : |a^\top v| \leq \gamma\}$. If $\gamma/\kappa \leq 1/\sqrt{p}$, we have*

$$\frac{\mathrm{vol}(B_+)}{\mathrm{vol}(B)} \leq \sqrt{p}\frac{\gamma}{\kappa}\left(\frac{p}{p-1}\right)^{(p-1)/2}\left(1-\frac{\gamma^2}{\kappa^2}\right)^{(p-1)/2} \tag{4.23}$$

Further, if $\gamma/\kappa \leq \frac{1}{3\sqrt{p}}$, the RHS of (4.23) *is less than* 0.6.

In the following lemma, we now show that the exploration step can happen only a finite number of times:

Lemma 4.11 (Bounding the number of iterations)**.** *If the estimates $\widehat{Z}_t$ and $\hat{y}_t$ in* (4.4) *and* (4.5) *satisfy:*

$$\left|\hat{y}_t - \sum_{h=0}^{H-1} \mathbb{E}_{d^{W_t}_{M^*,h}}\left[\mathbb{E}_{M^*}\left[r_h + V_{t,h+1}(s_{h+1})\middle|s_h, a_h\right]\right]\right| + \left|\left\langle W, \widehat{Z}_t\right\rangle - \langle W, Z_t\rangle\right| \leq \frac{\epsilon}{12\sqrt{dK}} \quad (4.24)$$

for all $W \in \mathcal{W}$, for all iterations in Algorithm 4.1, then W^* is never eliminated. Moreover, the number of exploration iterations of Algorithm 4.1 is at most $T = dK \log \frac{2d\sqrt{K}H}{\epsilon} / \log \frac{5}{3}$.

Proof. By definition, $W^* \in \mathcal{W}_0$. We first show that W^* is never eliminated from the version space $\mathcal{W}_t$. Let $\alpha_t \coloneqq \sum_{h=0}^{H-1} \mathbb{E}_{d^{W_t}_{M^*,h}}\left[\mathbb{E}_{M^*}\left[r_h + V_{t,h+1}(s_{h+1})\middle|s_h, a_h\right]\right]$ and by definition, we have $\langle W^*, Z_t\rangle \coloneqq \sum_{h=0}^{H-1} \mathbb{E}_{d^{W_t}_{M^*,h}}\left[\mathbb{E}_{M(W^*)}\left[r_h + V_{t,h+1}(s_{h+1})\middle|s_h, a_h\right]\right]$. Then, we have

$$\begin{aligned}
\left|\hat{y}_t - \left\langle W^*, \widehat{Z}_t\right\rangle\right| &\leq \left|\hat{y}_t - \alpha_t + \alpha_t - \langle W^*, Z_t\rangle + \langle W^*, Z_t\rangle - \left\langle W^*, \widehat{Z}_t\right\rangle\right| \\
&\leq \frac{\epsilon}{12\sqrt{dK}} + |\alpha_t - \langle W^*, Z_t\rangle| \\
&\leq \frac{\epsilon}{12\sqrt{dK}} + H\theta
\end{aligned}$$

Therefore, W^* always satisfies the update in (4.6) and is never eliminated.

Now, we argue that the *volume* of the version space decreases at an exponential rate with each exploration iteration. To set up the volume reduction analysis, we first notice that:

$$\begin{aligned}
\|W - W^*\|_F &\leq \sqrt{\sum_{i=1}^{d} \|W^i - W^{*i}\|_2^2} \\
&\leq \sqrt{2d}
\end{aligned}$$

Therefore, the initial volume of a ball covering the space of flattened vectors in $\mathcal{W}_0$ is at most $c_{dK}(\sqrt{2d})^{dK}$. We will now use Lemma 4.10 by considering the flattened versions of the parameter matrices W in the dimension $p = dK$[4]. Firstly, in each iteration until termination, we find a matrix W_t such that $\mathcal{E}(W_t) \geq \frac{\epsilon}{2} + (3\sqrt{dK} + 1)H\theta$. From Lemma 4.7, we have:

$$\begin{aligned}
\langle W_t - W^*, Z_t\rangle &\geq \mathcal{E}(W_t) - H\theta \\
&\geq \frac{\epsilon}{2} + 3\sqrt{dK}H\theta
\end{aligned}$$

[4]For avoiding ambiguity, we will directly use the matrix notations for inner products and norms.

We will apply Lemma 4.10 with $W_t - W^*$ as the vector u and Z_t as the direction vector a. For the updated version space, $\mathcal{W}_t$, we have:

$$\begin{aligned}|\langle W - W^*, Z_t\rangle| &= \left|y_t - \hat{y}_t + \left\langle W, Z_t - \widehat{Z}_t\right\rangle + \widehat{Z}_t - \hat{y}_t\right| \\ &\leq \frac{\epsilon}{12\sqrt{dK}} + H\theta + \frac{\epsilon}{12\sqrt{dK}} \\ &\leq \frac{\epsilon}{6\sqrt{dK}} + H\theta\end{aligned}$$

Denoting B_{t-1} as the MVEE of the version space $\mathcal{W}_{t-1}$, we consider the MVEE B'_t of the set of vectors $\mathcal{W}'_t \equiv \left\{W \in B_{t-1} : |\langle W - W^*, Z_t\rangle| \leq \epsilon/6\sqrt{dK} + H\theta\right\}$. Clearly, we have $\mathcal{W}_{t-1} \subseteq B_{t-1}$, and hence, $\mathcal{W}_t \subseteq \mathcal{W}'_t$. By setting $\kappa = \frac{\epsilon}{2}$ and $\gamma = \frac{\epsilon}{3\sqrt{dK}}$ in Lemma 4.10, we have:

$$\frac{\text{vol}(B_t)}{\text{vol}(B_{t-1})} \leq \frac{\text{vol}(B'_t)}{\text{vol}(B_{t-1})} \leq 0.6$$

This shows that the volume of the MVEE of the version spaces $\mathcal{W}_t$ decreases with at least a constant rate. We now argue that the procedure stops after reaching a version space with sufficiently small volume.

For any $W \in \mathcal{W}_t$ and $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have:

$$\begin{aligned}\left\|P^W(\cdot|s,a) - P^{W^*}(\cdot|s,a)\right\|_1 &\leq \left\|\sum_{k=1}^{K} \left[(W - W^*)\phi(s,a)\right]_k P^k(\cdot|s,a)\right\|_1 \\ &\leq \left\|(W - W^*)\phi(s,a)\right\|_1 \\ &\leq \sqrt{K}\left\|(W - W^*)\phi(s,a)\right\|_2 \\ &\leq \sqrt{dK}\left\|W - W^*\right\|_F\end{aligned}$$

Also, using Lemma 4.6, if the worst case error in next state transition estimates is bounded by $\frac{\epsilon}{2H} + \theta$, the optimistic value $V^{\pi_W}_{W^*}$ is $\epsilon + H\theta$-optimal. Therefore, we only need to identify the matrix W to within $\frac{\epsilon}{2H\sqrt{dK}}$ distance of W^*. Consequently, the terminating MVEE B_T satisfies:

$$B_T \supseteq \left\{W : \left\|W - W^*\right\|_F \leq \frac{\epsilon}{2\sqrt{dK}H}\right\}$$

Therefore, we have $\mathrm{vol}(B_T) \geq c_{dK}\left(\epsilon/2\sqrt{dK}H\right)^{dK}$, and :

$$\frac{c_{dK}\left(\epsilon/2\sqrt{dK}H\right)^{dK}}{c_{dK}\left(\sqrt{2d}\right)^{dK}} \leq \frac{\mathrm{vol}(B_T)}{\mathrm{vol}(B_0)} \leq 0.6^T$$

By solving for T, we get that:

$$\begin{aligned} dK \log \frac{2\sqrt{2K}H}{\epsilon} &\geq T \log \frac{5}{3} \\ T &\leq dK \log \frac{2\sqrt{2K}H}{\epsilon} / \log \frac{5}{3} \end{aligned} \tag{4.25}$$

□

We now derive the number of trajectories required in each step to satisfy the validity requirements in Lemma 4.9 and Lemma 4.11:

Lemma 4.12 (Concentration for MC estimate $\hat{v}_t$)**.** *For any $W_t \in \mathcal{W}$ with probability at least $1-\delta_1$, we have:*

$$\left|\hat{v}_t - v_{M^*}^{W_t}\right| \leq \tfrac{\epsilon}{4} \tag{4.26}$$

if we set $n_{\text{eval}} \geq \frac{8}{\epsilon^2}\log\frac{2}{\delta_1}$.

Proof. Note that $\hat{v}_t$ is an unbiased estimate of $v_{M^*}^{W_t}$. From our assumption on the expected sum of rewards, the return of each trajectory is bounded by 1 for all policies. Thus, the range of each summand for the estimate $\hat{v}_t$ is $[0,1]$. Then, the result follows from standard application of Hoeffding's inequality. □

Lemma 4.13 (Concentration of the model misfit error)**.** *If $n \geq \frac{1800d^2KH^2}{\epsilon^2}\log\frac{4dK}{\delta_2}$ in Algorithm 4.1, then for a given t, each $W \in \mathcal{W}$ and with probability at least $1-\delta_2$, we have:*

$$\left|\hat{y}_t - \sum_{h=0}^{H-1} \mathbb{E}_{d_{M^*,h}^{W_t}}\left[\mathbb{E}_{M^*}\left[r_h + V_{t,h+1}(s_{h+1})\middle|s_h,a_h\right]\right]\right| + \left|\left\langle W, Z_t - \widehat{Z}_t\right\rangle\right| \leq \frac{\epsilon}{12\sqrt{dK}} \tag{4.27}$$

Proof. To bound the first term, we note that $\hat{y}_t$ is an unbiased estimate of $\sum_{h=0}^{H-1}\mathbb{E}_{d_{M^*,h}^{W_t}}\left[\mathbb{E}_{M^*}\left[r_h + V_{t,h+1}(s_{h+1})\middle|s_h,a_h\right]\right]$ and is bounded between $[0,H]$. Applying Hoeffding's inequality on the estimand $\hat{y}_t$ when using n trajectories, with probability at least $1-\delta'$,

we get:

$$\left|\hat{y}_t - \sum_{h=0}^{H-1} \mathbb{E}_{d_{M^*,h}^{W_t}} \left[\mathbb{E}_{M^*}\left[r_h + V_{t,h+1}(s_{h+1}) \middle| s_h, a_h\right]\right]\right| \leq H\sqrt{\frac{1}{2n}\log\frac{2}{\delta'}}$$

For bounding the second term, using Holder's inequality with matrix group norm (Agarwal et al., 2008), we first see:

$$\begin{aligned}\left|\left\langle W, \widehat{Z}_t\right\rangle - \langle W, Z_t\rangle\right| &\leq \|W\|_{1,\infty}\left\|\widehat{Z}_t - Z_t\right\|_{\infty,1} \\ &\leq \left\|\widehat{Z}_t - Z_t\right\|_{\infty,1}\end{aligned}$$

We will now bound the estimation error for $\widehat{Z}_t$:

$$\begin{aligned}&\left\|\widehat{Z}_t - Z_t\right\|_{\infty,1} \\ &= \left\|\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\overline{V}_{t,h}\left(s_h^{(i)}, a_h^{(i)}\right)\phi\left(s_h^{(i)}, a_h^{(i)}\right)^\top - \mathbb{E}\left[\sum_{h=0}^{H-1}\overline{V}_{t,h}(s_h, a_h)\,\phi(s_h, a_h)^\top\right]\right\|_{\infty,1} \\ &= \sum_{j=1}^{d}\left\|\frac{1}{n}\sum_{i=1}^{n}\sum_{h=0}^{H-1}\overline{V}_{t,h}\left(s_h^{(i)}, a_h^{(i)}\right)\phi\left(s_h^{(i)}, a_h^{(i)}\right)[j] - \mathbb{E}\left[\sum_{h=0}^{H-1}\overline{V}_{t,h}(s_h, a_h)\,\phi(s_h, a_h)[j]\right]\right\|_{\infty}\end{aligned}$$

We can consider each trajectory as a random sample from the distribution of trajectories induced by π_t. Therefore, by definition, each summand in the estimate of $\widehat{Z}_t$ over n trajectories is an unbiased estimate of Z_t. Moreover, we know that each term in the entry $\widehat{Z}_t[i,j]$ of the matrix $\widehat{Z}_t$ is bounded by H. Using Bernstein's inequality for each term in the error matrix, and a union bound over all entries, with probability at least $1-\delta'$, for all i, j we have:

$$\left|\widehat{Z}_t[i,j] - Z_t[i,j]\right| \leq \sqrt{\frac{2\mathrm{Var}\left[\sum_{h=0}^{H-1}\overline{V}_{t,h}[i]\phi(s_h,a_h)[j]\right]\log\frac{2dK}{\delta'}}{n}} + \frac{2H\log\frac{2dK}{\delta'}}{n}$$

Summing up the maximum elements i_j of each column, we have:

$$
\begin{aligned}
norm\widehat{Z}_t - Z_{t\infty,1} &\leq \sum_{j=1}^{d} \sqrt{\frac{2\mathrm{Var}\left[\sum_{h=0}^{H-1} \overline{V}_{t,h}[i_j]\phi(s_h,a_h)[j]\right]\log\frac{2dK}{\delta'}}{n}} + \frac{2dH\log\frac{2dK}{\delta'}}{n} \\
&= \sum_{j=1}^{d} \sqrt{\frac{2\mathbb{E}\left[\left(\sum_{h=0}^{H-1}\phi(s_h,a_h)[j]\right)^2\right]\log\frac{2dK}{\delta'}}{n}} + \frac{2dH\log\frac{2dK}{\delta'}}{n} \\
&\leq \sqrt{\frac{2d\mathbb{E}\left[\sum_{j=1}^{d}\left(\sum_{h=0}^{H-1}\phi(s_h,a_h)[j]\right)^2\right]\log\frac{2dK}{\delta'}}{n}} + \frac{2dH\log\frac{2dK}{\delta'}}{n} \\
&\leq \sqrt{\frac{2d\mathbb{E}\left[\left(\sum_{j=1}^{d}\sum_{h=0}^{H-1}\phi(s_h,a_h)[j]\right)^2\right]\log\frac{2dK}{\delta'}}{n}} + \frac{2dH\log\frac{2dK}{\delta'}}{n} \\
&\leq H\sqrt{\frac{2d\log\frac{2dK}{\delta'}}{n}} + \frac{2dH\log\frac{2dK}{\delta'}}{n}
\end{aligned}
$$

Here, for the first step, we have used the property that $\overline{V}_{t,h}[i_j] \leq 1$, the fact that variance is bounded by the second moment. The next step can be obtained by using Cauchy-Schwartz inequality. The last second step uses that property that for non-negative a_j, $\sum_j a_j^2 \leq (\sum_j a_j)^2$. Now, if $\frac{2d\log\frac{2}{\delta'}}{n} \leq 1$, the above is bounded by $2\sqrt{\frac{2d\log\frac{2}{\delta'}}{n}}$.

Therefore, summing up the two terms with $\delta' = \delta_2/2$, with probability at least $1-\delta_2$ and for all $W \in \mathcal{W}$, we have:

$$
\left|\hat{y}_t - \sum_{h=0}^{H-1}\mathbb{E}_{d_{M^*,h}^{W_t}}\left[\mathbb{E}_{M^*}\left[r_h + V_{t,h+1}(s_{h+1})\middle|s_h,a_h\right]\right]\right| + \left|\left\langle W, Z_t - \widehat{Z}_t\right\rangle\right| \leq \sqrt{\frac{8dH^2\log\frac{4dK}{\delta_2}}{n}} + \sqrt{\frac{H^2\log\frac{4}{\delta_2}}{2n}}
$$

With some algebra, it can be verified that setting $n = \frac{1800d^2KH^2}{\epsilon^2}\log\frac{4dK}{\delta_2}$ makes the total error bounded by $\frac{\epsilon}{12\sqrt{dK}}$ with failure probability δ_2. □

4.9.2 Proof of Theorem 4.1

Proof. With the key lemmas in Section 4.9.1, we can now prove the main result. For the main theorem, we need to ensure that the requirements in Lemma 4.9 and Lemma 4.11 are satisfied.

Since, our method maintains a version space of plausible weights, the validity of each iteration depends on every previous iteration being valid. Therefore, for a total failure probability of δ for the algorithm, we assume:

(i) Estimation of $\hat{\mathcal{E}}(W_t)$ for all iterations: total failure probability $\delta/2$

(ii) Updating the version space $\mathcal{W}_t$: total failure probability $\delta/2$

We set $\delta_1 = \delta/2T$ and $\delta_2 = \delta/2T$ in Lemma 4.12 and Lemma 4.13 respectively with $T = dK \log \frac{2\sqrt{2K}H}{\epsilon} / \log \frac{5}{3}$. By taking a union bound over maximum number of iterations, the total failure probability is bounded by δ. Thus, the total number of trajectories unrolled by the algorithm is:

$$\begin{aligned} T(n_{\text{eval}} + n) &\leq \left(dK \log \frac{2\sqrt{2K}H}{\epsilon} / \log \frac{5}{3} \right) \left(\frac{8}{\epsilon^2} \log \frac{4T}{\delta} + \frac{1800 d^2 K H^2}{\epsilon^2} \log \frac{8dKT}{\delta} \right) \\ &= \widetilde{O}\left(\frac{d^3 K^2 H^2}{\epsilon^2} \log \frac{1}{\delta} \right) \end{aligned}$$

Therefore, by the termination guarantee in Lemma 4.9, we arrive at the desired upper bound on the number of trajectories required to guarantee a policy with value $v^{\pi}_{M^*} \geq v^* - (3\sqrt{dK} + 1)H\theta$. $\square$

CHAPTER 5

Model-Free Feature Learning and Exploration in Low-Rank MDPs

In the previous chapters, we have explored how an underlying linear structure can be used to efficiently explore in a multi-task contextual setting or an arbitrarily complex domain. Similar to Section 4.4 of Chapter 4, in this chapter, we investigate a model selection problem for another linear model setting, specifically, the low-rank MDP setting (also known as linear MDP). In this chapter, we consider a problem setting where the agent is given a representation class Φ which contains the true feature for the underlying linear MDP. We propose the first model-free representation learning algorithms, wherein, we learn a representation from the given class and collect an exploratory dataset. Our main result shows that the learnt representation can be used to compute a near-optimal policy for any reward in a given class using the fitted Q-iteration method described in Section 2.2.4. As such, in this chapter, we propose a provably efficient representation learning and exploration algorithm for the reward-free setting. In addition, we discuss and analyze variants of the main procedure with varying computational and statistical tradeoffs. The resulting algorithms are provably sample efficient and can accommodate general function approximation to scale to complex environments.

5.1 Introduction

A key driver of recent empirical successes in machine learning is the use of rich function classes for discovering transformations of complex data, a sub-task referred to as *representation learning*. For example, when working with images or text, it is standard to train extremely large neural networks in a self-supervised fashion on large datasets, and then fine-tune the network on supervised tasks of interest. The representation learned in the first stage is essential for sample-efficient generalization on the supervised tasks. Can we endow Reinforcement Learning (RL) agents with a similar capability to discover representations that provably enable sample efficient learning in downstream tasks?

In the empirical RL literature, representation learning often occurs implicitly simply through the use of deep neural networks, for example in DQN (Mnih et al., 2015). Recent work has also considered more explicit representation learning via auxiliary losses like inverse dynamics (Pathak et al., 2017), the use of explicit latent state space models (Hafner et al., 2019; Sekar et al., 2020), and via bisimulation metrics (Gelada et al., 2019; Zhang et al., 2020). Crucially, these explicit representations are again often trained in a way that they can be reused across a variety of related tasks, such as domains sharing the same (latent state) dynamics but differing in reward functions.

While these works demonstrate the value of representation learning in RL, theoretical understanding of such approaches is limited. Indeed obtaining sample complexity guarantees is quite subtle as recent lower bounds demonstrate that various representations are not useful or not learnable (Modi et al., 2020; Du et al., 2019b; Van Roy and Dong, 2019; Lattimore and Szepesvari, 2020; Hao et al., 2021). Despite these lower bounds, some prior theoretical works do provide sample complexity guarantees for non-linear function approximation (Jiang et al., 2017; Sun et al., 2019; Osband and Van Roy, 2014; Wang et al., 2020b; Yang et al., 2020), but these approaches do not obviously enable generalization to related tasks.

More direct representation learning approaches were recently studied in Du et al. (2019a); Misra et al. (2020); Agarwal et al. (2020b), who develop algorithms that provably enable sample efficient learning in any downstream task that shares the same dynamics.

In this chapter, our work builds on the most general of the direct representation learning approaches, namely the FLAMBE algorithm of Agarwal et al. (2020b), that finds features under which the transition dynamics are nearly linear. The main limitation of FLAMBE is the assumption that the dynamics can be described in a parametric fashion. In contrast, we take a model-free approach to this problem, thereby accommodating much richer dynamics.

Concretely, we study the low-rank MDP setting (also called linear MDP, factored linear MDP, etc.), which we described in Section 2.1.4. For model-free representation learning, we assume access to a function class Φ containing the underlying feature map ϕ^*. This is a much weaker inductive bias than prior work in the "known features" setting where ϕ^* is known in advance (Jin et al., 2020; Yang and Wang, 2020; Agarwal et al., 2020a) and the model-based setting (Agarwal et al., 2020b) that assumes realizability for both μ^* and ϕ^*.

While our model-free setting captures richer MDP models, addressing the intertwined goals of representation learning and exploration is much more challenging. In particular, the forward and inverse dynamics prediction problems used in prior works are no longer admissible under our weak assumptions. Instead, we address these challenges with a new representation learning procedure based on the following insight: for any function $f : \mathcal{X} \to \mathbb{R}$, the Bellman backup of f is a linear function in the feature map ϕ^*. This leads to a natural minimax objective, where we search for a representation $\hat{\phi}$ that can linearly approximate the Bellman backup of all functions in some

"discriminator" class $\mathcal{F}$. Importantly, the discriminator class $\mathcal{F}$ is induced directly by the class Φ, so that no additional realizability assumptions are required. We also provide an incremental approach for expanding the discriminator set, which leads to a more computationally practical variant of our algorithm.

The two algorithms reduce to minimax optimization problems over non-linear function classes. While such problems can be solved empirically with modern deep learning libraries, they do not come with rigorous computational guarantees. To this end, we investigate the special case of finite feature classes ($|\Phi| < \infty$). We show that when Φ is efficiently enumerable, our optimization problems can be reduced to eigenvector computations, which leads to provable computational efficiency. The results in this chapter represent the first statistically and computationally efficient model-free algorithms for representation learning in RL.

5.2 Problem Setup

In this chapter, we consider an episodic MDP setting under the low rank structure defined in Definition 2.2. In this chapter, we assume that the size of the action space A is finite and derive sample complexity bounds which depend polynomially on A. We use π_h denotes an h-step policy that chooses actions $a_0, \ldots, a_h$. We also use $\mathbb{E}_\pi[\cdot]$ and $\mathbb{P}_\pi[\cdot]$ to denote the expectations over states and actions and probability of an event respectively, when using policy π in $\mathcal{M}$. Below, we redefine the low-rank MDP with additional details as required in our analysis for this chapter:

Definition 5.1. *An operator $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ admits a low-rank decomposition of dimension d if there exists functions $\phi^* : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $\mu^* : \mathcal{S} \to \mathbb{R}^d$ such that: $\forall s, s' \in \mathcal{S}, a \in \mathcal{A}$: $P(s'|s,a) = \langle \phi^*(s,a), \mu^*(s') \rangle$, and additionally $\|\phi^*(s,a)\|_2 \leq 1$ and for all $g : \mathcal{S} \to [0,1]$, $\| \int g(s)\mu^*(s)ds \|_2 \leq \sqrt{d}$. We assume that $\mathcal{M}$ is low-rank with embedding dimension d, i.e., for each $h \in [H]$, the transition operator P_h admits a rank-d decomposition.*

We denote the embedding for P_h by ϕ_h^* and μ_h^*. In addition to the low-rank representation, we also consider a latent variable representation of $\mathcal{M}$, as defined in Agarwal et al. (2020b), as follows:

Definition 5.2. *The latent variable representation of a transition operator $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a latent space $\mathcal{Z}$ along with functions $\psi : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{Z})$ and $\nu : \mathcal{Z} \to \Delta(\mathcal{S})$, such that $P(\cdot|s,a) = \int \nu(\cdot|z)\psi(z|s,a)dz$. The latent variable dimension of P, denoted d_{LV} is the cardinality of smallest latent space $\mathcal{Z}$ for which P admits a latent variable representation. In other words, this representation gives a non-negative factorization of the transition operator P.*

When state space $\mathcal{S}$ is finite, all transition operators $P_h(\cdot|s,a)$ admit a trivial latent variable representation. More generally, the latent variable representation enables us to augment the trajectory

τ as: $\tau = \{s_0, a_0, z_1, s_1, \ldots, z_{H-1}, s_{H-1}, a_{H-1}, z_H, s_H\}$, where $z_{h+1} \sim \psi_h(\cdot|s_h, a_h)$ and $s_{h+1} \sim \nu_h(\cdot|z_{h+1})$. In general we neither assume access to nor do we learn this representation, and it is solely used to reason about the following reachability assumption:

Assumption 5.1 (Reachability)**.** *There exists a constant $\eta_{\min} > 0$, such that $\forall h \in [H], z \in \mathcal{Z}_{h+1}$:* $\max_\pi \mathbb{P}_\pi [z_{h+1} = z] \geq \eta_{\min}$.

Assumption 5.1 posits that in MDP $\mathcal{M}$, for each factor (latent variable) at any level h, there exists a policy which reaches it with a non-trivial probability. This generalizes the reachability of latent states assumption from prior block MDP results (Du et al., 2019a; Misra et al., 2020). Note that, exploring all latent states is still non-trivial, as a policy which chooses actions uniformly at random may hit these latent states with an exponentially small probability.

Representation learning in low-rank MDPs We consider MDPs where the state space $\mathcal{S}$ is large and the agent must employ function approximation to conduct effective learning. Given the low-rank MDP assumption, we grant the agent access to a class of representation functions mapping a state-action pair (s, a) to a d-dimensional embedding. Specifically, the feature class is $\Phi = \{\Phi_h : h \in [H]\}$, where each mapping $\phi_h \in \Phi_h$ is a function $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$. The feature class can now be used to learn ϕ^*[1] and exploit the low-rank decomposition for efficient learning. We assume that our feature class Φ is rich enough:

Assumption 5.2 (Realizability)**.** *For each $h \in [H]$, we have $\phi_h^* \in \Phi_h$. Further, we assume that $\forall \phi_h \in \Phi_h, \forall (s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\phi_h(s, a)\|_2 \leq 1$.*

Learning goal We focus on the problem of representation learning (Agarwal et al., 2020b) in low-rank MDPs where the agent tries to learn good enough features that enable offline optimization of any given reward in downstream tasks instead of optimizing a fixed and explicit reward signal. We consider a model free setting and we provide this *reward-free* learning guarantee for any reward function R in a bounded reward class $\mathcal{R}$. Specifically, for such a bounded reward function $R : \mathcal{S} \times \mathcal{A} \to [0, 1]$, the learned features $\{\bar{\phi}_h\}_{h \in [H]}$ and the collected data should allow the agent to compute a near-optimal policy π_R, such that $v_R^{\pi_R} \geq v_R^* - \varepsilon$.[2] We desire (w.p. $\geq 1 - \delta$) sample complexity bounds which are $\text{poly}(d, H, A, 1/\eta_{\min}, 1/\varepsilon, \log|\Phi|, \log|\mathcal{R}|, \log(1/\delta))$.

[1] Sometimes we drop h in the subscript for brevity.

[2] Here, $v_R^\pi = \mathbb{E}_\pi \left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right]$ is the expected value of policy π and v_R^* is the optimal value.

5.3 MoFFLe: Main Algorithm

In this section, we describe the overall algorithmic framework that we propose for representation learning for low-rank MDPs. A key component in this general framework, which specifies how to recover a good representation once exploratory data has been acquired, is left unspecified in this section and instantiated with two different choices in the subsequent sections. We also present sample complexity guarantees for each choice in the corresponding sections.

At the core of our model-free approach is the following well-known property of a low-rank MDP due to Jin et al. (2020). We provide a proof in Appendix B.2.1 for completeness.

Lemma 5.1 (Jin et al. (2020))**.** *For a low-rank MDP $\mathcal{M}$ with embedding dimension d, for any $f : \mathcal{S} \to [0, 1]$, we have: $\mathbb{E}\left[f(s_{h+1})|s_h, a_h\right] = \langle \phi^*_h(s_h, a_h), \theta^*_f \rangle$, where $\theta^*_f \in \mathbb{R}^d$ and $\|\theta^*_f\|_2 \leq \sqrt{d}$.*

We turn this property into an algorithm by finding a feature map in our class Φ which can verify this condition for a rich enough class of functions $\mathcal{F}$. The key insight in our algorithm is that this property depends solely on ϕ^*, so we do not require additional modeling assumptions.

Before turning to the algorithm description, we clarify useful notation for h step policies. An h-step policy ρ_h chooses actions $a_0, \ldots, a_h$, consequently inducing a distribution over (s_h, a_h, s_{h+1}). We routinely append several random actions to such a policy, and we use ρ_h^{+i} to denote the policy that chooses $a_{0:h}$ according to ρ_h and then takes actions uniformly at random for i steps, inducing a distribution over $(s_{h+i}, a_{h+i}, s_{h+i+1})$. As an edge case, for $i \geq j \geq 0$, ρ_{-j}^{+i} takes actions $a_0, \ldots, a_{i-j}$ uniformly. The mnemonic is that the last action taken by ρ_j^{+i} is a_{i+j}.

Algorithm 5.1 MoFFLe $(\mathcal{R}, \Phi, \eta_{\min}, \varepsilon, \delta)$: Model-Free Feature Learning and Exploration

Set $\mathcal{D} \leftarrow \emptyset$, $\varepsilon_{\text{apx}} \leftarrow \varepsilon^2/(16H^4\kappa A)$.
Compute the exploratory policy cover: $\left\{\rho_{h-3}^{+3}\right\}_{h\in[H]} \leftarrow$ Explore $\left(\Phi, \eta_{\min}, \frac{\delta}{2}\right)$.
for $h \in [H]$ **do**
  Collect dataset $\mathcal{D}_h^{\text{rep}}$ of size n_{rep} using ρ_{h-3}^{+3}.
  Learn representation $\bar{\phi}_h$ by solving (5.8) (or calling Algorithm 5.4) with class $\mathcal{V} = \mathcal{G}_{h+1}$, dataset $\mathcal{D}_h^{\text{rep}}$ and tolerance ε_{apx}.
  Collect dataset $\mathcal{D}_h$ of size n_{fqi} using ρ_{h-3}^{+3}.
  Set $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathcal{D}_h\}$.
return $\mathcal{D}, \bar{\phi}_{0:H-1}$.

Algorithm 5.2 Explore $(\Phi, \eta_{\min}, \delta_e)$

Set $\mathcal{D}_h^{\text{ell}} \leftarrow \emptyset$ for each $h \in [H]$.
for $h = 0, \ldots, H-1$ **do**
  Set exploratory policy for step h to ρ_{h-3}^{+3}.
  Collect dataset $\mathcal{D}_h^{\text{exp}}$ of size n_{exp} using ρ_{h-3}^{+3}.
  Learn representation $\hat{\phi}_h$ for timestep h by solving (5.8) (or calling Algorithm 5.4) with class $\mathcal{F}_{h+1}$, dataset $\mathcal{D}_h^{\text{exp}}$ and tolerance ε_{reg}.
  Collect dataset $\mathcal{D}_h^{\text{ell}}$ of size n_{ell} using ρ_{h-3}^{+3}.
  Call planner (Algorithm 5.3) with features $\hat{\phi}$, dataset $\mathcal{D}_{0:h}^{\text{ell}}$ and β to obtain policy ρ_h.
return Exploratory policies $\left\{\rho_{h-3}^{+3}\right\}_{h\in[H]}$.

Our algorithm, Model Free Feature Learning and Exploration (MOFFLE) shown in Algorithm 5.1, takes as input a feature set Φ, a reward class $\mathcal{R}$, and some parameters related to the final accuracy desired. It outputs a feature map and a dataset such that Fitted Q-Iteration (FQI) (Section 2.2.4), using linear functions of the returned features, can be run with the returned dataset to obtain a near-optimal policy for any reward function in $\mathcal{R}$. The algorithm runs in two stages:

Designing Exploratory Policies In line 2 of MOFFLE, we use the EXPLORE sub-routine (Algorithm 5.2) to compute exploratory policies ρ_{h-3}^{+3} for each timestep $h \in [H]$.

Algorithm 5.2 uses a step-wise forward exploration scheme similar to FLAMBE (Agarwal et al., 2020b). The algorithm proceeds in stages where for each h, we first use an exploratory policy ρ_{h-3}^{+3} (line 5) to learn features $\hat{\phi}_h$ by calling a feature learning sub-routine (discussed in the sequel).

The feature $\hat{\phi}_h$ is computed to approximate the property in Lemma 5.1 for the discriminator function class $\mathcal{F}_{h+1} \subset (\mathcal{S} \to [0,1])$ that contains all functions f of the form of $f(s_{h+1}) = \text{clip}_{[0,1]}(\mathbb{E}_{\text{unif}(\mathcal{A})} \langle \phi_{h+1}(s_{h+1}, a), \theta \rangle)$ where $\phi_{h+1} \in \Phi_{h+1}$ and $\|\theta\|_2 \leq B$ for some $B \geq \sqrt{d}$. Using the policy ρ_{h-3}^{+3}, we also collect the exploratory dataset $\mathcal{D}_h^{\text{ell}}$ for step h (line 6). With the collected datasets[3] $\mathcal{D}_{0:h}^{\text{ell}}$ and features $\hat{\phi}_h$ we call an offline "elliptical" planning subroutine (Algorithm 5.3) to compute the policy ρ_h. This planning algorithm is inspired by techniques used in reward-free exploration (Wang et al., 2020a; Zanette et al., 2020) and is analyzed in Section 5.10.6.

Algorithm 5.3 FQI based Elliptical Planner

1: **input:** Exploratory dataset $\mathcal{D} := \mathcal{D}_{0:\tilde{H}}$ and $\beta > 0$.
2: Initialize $\Gamma_0 = I_{d \times d}$.
3: **for** $t = 1, 2, \ldots,$ **do**
4: Using Algorithm B.1, compute

$$\pi_t = \text{FQI}\left(\mathcal{D}, R_{\tilde{H}} = \left\|\hat{\phi}_{\tilde{H}}(s,a)\right\|^2_{\Gamma_{t-1}^{-1}}\right).$$

5: If the estimated objective is at most $\frac{3\beta}{4}$, halt and output $\rho := \text{unif}(\{\pi_\tau\}_{\tau<t})$.
6: Estimate feature covariance matrix $\widehat{\Sigma}_{\pi_t}$ using n_{plan} rollouts from policy π_t as:

$$\frac{1}{n_{\text{plan}}} \sum_{i=1}^{n_{\text{plan}}} \hat{\phi}_{\tilde{H}}\left(s_{\tilde{H}}^{(i)}, a_{\tilde{H}}^{(i)}\right) \hat{\phi}_{\tilde{H}}\left(s_{\tilde{H}}^{(i)}, a_{\tilde{H}}^{(i)}\right)^\top.$$

7: Update $\Gamma_t \leftarrow \Gamma_{t-1} + \widehat{\Sigma}_{\pi_t}$.

[3]For a dataset $\mathcal{D}_h$, subscript h denotes that it is a collection of tuples (s_h, a_h, s_{h+1}).

Representation learning We subsequently learn a feature $\bar{\phi}_h$ for each level—again by invoking the representation learning subroutine—that allows us to use FQI to plan for any reward $R \in \mathcal{R}$ afterwards. Here, we use a discriminator function class $\mathcal{G}_{h+1} \subset (\mathcal{S} \to [0, H])$ which is the set of functions $g(s') = \text{clip}_{[0,H]}\left(\max_a \left(R(s', a) + \langle \phi_{h+1}(s', a), \theta \rangle\right)\right)$ with $R \in \mathcal{R}, \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq B$ and $B \geq H\sqrt{d}$. Note that the class $\mathcal{G}_{h+1}$, while still derived from Φ is quite different from the class $\mathcal{F}_{h+1}$ used to learn features inside the exploration module.

Finally, MOFFLE returns the computed features $\bar{\phi}_{0:H-1}$ and the exploratory dataset $\mathcal{D}_{0:H-1}$.

Planning in downstream tasks For downstream planning with any reward $R \in \mathcal{R}$, we use FQI (Chen and Jiang, 2019) with the following Q function class defined using the features $\bar{\phi}_{0:H-1}$:

$$\mathcal{Q}_h(\bar{\phi}, R) = \left\{\text{clip}_{[0,H]}\left(R_h(s, a) + \langle \bar{\phi}_h(s, a), w \rangle\right) : \|w\|_2 \leq B\right\}. \tag{5.1}$$

Note that the features $\bar{\phi}_h$ are computed to approximate the backup of candidate functions of this form (class $\mathcal{G}_{h+1}$), and thus, satisfy the conditions stated in Chen and Jiang (2019) for using FQI as discussed in Section 2.2.4.

5.3.1 Understanding the design choices in MOFFLE

In this section, we provide some intuition behind the design choices in MOFFLE and give a sketch of the proof of the main results. Similar to the algorithm description, we divide the proof sketch in two stages: establishing the exploratory nature of the policies ρ_{h-3}^{+3} and approximation power of features $\bar{\phi}_h$ for all levels $h \in [H]$.

Computing exploratory policies To understand the intuition behind Algorithm 5.2, it is helpful to consider how we can discover a policy cover over the latent state space $\mathcal{Z}_{h+1}$. If we knew the mapping to latent states, we could create the reward functions $\mathbf{1}\{z_{h+1} = z\}$ for all $z \in \mathcal{Z}_{h+1}$ and compute policies to optimize such rewards, but we do not have access to this mapping. Additionally, we do not have access to the features ϕ^* to enable tractable planning even for the known rewards. Algorithm 5.2 tackles both of these challenges. For the first challenge, we note that by Definition 5.2 of the latent variables and Lemma 5.1, there always exists f such that

$$\mathbb{E}[\mathbf{1}\{z_{h+1} = z\} |\, s_{h-1}, a_{h-1}] = \mathbb{E}[f(s_h, a_h) |\, s_{h-1}, a_{h-1}] = \langle \phi_{h-1}^*(s_{h-1}, a_{h-1}), \theta_f^* \rangle.$$

The feature learning step in Algorithm 5.2 learns $\hat{\phi}_{h-2}$ so that for all appropriately bounded θ, there is a w such that

$$\mathbb{E}[\langle \phi^*_{h-1}(s_{h-1}, a_{h-1}), \theta \rangle |\, s_{h-2}, a_{h-2}] \approx \langle \hat{\phi}_{h-2}(s_{h-2}, a_{h-2}), w \rangle. \tag{5.2}$$

Given this property, we prove that coverage over $\mathcal{Z}_{h+1}$ can be ensured using coverage of all the reachable directions under $\hat{\phi}_{h-2}$ in the MDP followed by two uniform actions. Thus, we use a cover over $\mathcal{Z}_h$ to learn features $\hat{\phi}_h$ satisfying (5.2), and then plan in the previously learned features $\hat{\phi}_{h-2}$ to obtain a cover over $\mathcal{Z}_{h+1}$. This way, planning trails feature learning like FLAMBE, but with an additional step of lag due to differences between model-free and model-based reasoning.

For feature learning, we choose the class $\mathcal{F}_{h-1}$ inspired by our desideratum in (5.2) and learn features $\hat{\phi}_{h-2}$ (line 5) such that, for some fixed scalar B,

$$\max_{f \in \mathcal{F}_{h-1}} \text{b\_err}\left(\rho^{+3}_{h-3}, \hat{\phi}_{h-2}, f; B\right) \leq \varepsilon_{\text{reg}}. \tag{5.3}$$

For learning a policy ρ_{h-2} that effectively covers all directions spanned by $\hat{\phi}_{h-2}$, we employ the "elliptical planner" technique for reward-free exploration (Wang et al., 2020a; Zanette et al., 2020) and optimize reward functions that are quadratic in the learnt features $\hat{\phi}_{h-2}$. To do so, we repeatedly invoke an FQI subroutine with a function class comprising of all linear functions of $\phi \in \Phi$. We provide a complete description and analysis for the elliptic planner in Section 5.10.6. The reason for planning at $h-2$ is precisely based on our earlier intuition, formalizing which, we show that the policy $\rho^{+2}_{h-2} = \rho_{h-2} \circ \text{unif}(\mathcal{A}) \circ \text{unif}(\mathcal{A})$ is exploratory and hits all latent states in $z \in \mathcal{Z}_{h+1}$:

$$\max_{\pi} \mathbb{P}_\pi\left[z_{h+1} = z\right] \leq \kappa \mathbb{P}_{\rho^{+2}_{h-2}}\left[z_{h+1} = z\right], \tag{5.4}$$

where $\kappa > 0$ is a constant we specify later. Taking the action a_{h+1} uniformly at random inductively returns an exploratory policy ρ^{+3}_{h-2} for state-action pairs (s_{h+1}, a_{h+1}). Note that for the first step $h = 0$, the null policy ρ^{+2}_{-3} satisfies the exploration guarantee in (5.4).

We provide the following result for the policies returned by Algorithm 5.2 with details in Section 5.10.1.

Theorem 5.2. *Fix $\delta \in (0,1)$ and consider the setup in Theorem 5.4. If the features $\hat{\phi}_h$ learned in line 5 in Algorithm 5.2 satisfy the condition in (5.3) for $B \geq \sqrt{d}$, and $\varepsilon_{\text{reg}} = \widetilde{\Theta}\left(\frac{\eta^3_{\min}}{d^2 A^9 \log^2(1+8/\beta)}\right)$, then with probability at least $1 - \delta_e$, the sub-routine* EXPLORE *collects an exploratory mixture policy ρ^{+3}_{h-3} for each level h such that:*

$$\forall \pi, \forall f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+, \text{ we have } \mathbb{E}_\pi[f(s_h, a_h)] \leq \kappa A \mathbb{E}_{\rho^{+3}_{h-3}}[f(s_h, a_h)], \tag{5.5}$$

where $\kappa = \frac{64dA^4 \log(1+8/\beta)}{\eta_{\min}}$. *The total number of episodes used in line 7 by Algorithm 5.2 is:*

$$\widetilde{O}\left(\frac{H^5 d^6 A^{13} B^4 \log(|\Phi|/\delta_e)}{\eta_{\min}^5} + \frac{H d^6 A^{12} B^6 \log(|\Phi|/\delta_e)}{\eta_{\min}^6}\right),$$

with β *chosen to satisfy* $\beta \log(1 + 8/\beta) \le \eta_{\min}^2/128dA^4B^2$.

The precise dependence on parameters d, H, A, and $\eta_{\min}$ is likely improvable. The exponent on A arises from multiple importance sampling steps over the uniform action choice and can be improved when the features $\phi(s, a) \in \Delta(d)$ for all s, a, and $\phi \in \Phi$ (Section 5.10.1.2). Improving these dependencies further is an interesting avenue for future progress.

Representation learning for downstream tasks For showing planning guarantees using FQI, we need to ensure that the following requirements stated in Chen and Jiang (2019) are satisfied: (1) (*concentrability*) we have adequate coverage over the state space, (2) (*realizability*) we can express Q^* with our function class, and (3) (*completeness*) our class is closed under Bellman backups. The first condition is implied by Theorem 5.2. For (2) and (3), we learn features $\bar{\phi}_{0:H-1} \in \Phi$ such that $\bar{\phi}_h$ satisfies

$$\max_{g \in \mathcal{G}_{h+1}} \text{b\_err}\left(\rho_{h-3}^{+3}, \bar{\phi}_h, g; B\right) \le \varepsilon_{\text{apx}}, \tag{5.6}$$

where $\mathcal{G}_{h+1}$ is the discriminator class containing all candidate Q-value functions of form (5.1) (which in turn includes the true Q^* value function). The discriminator class $\mathcal{G}_{h+1}$ is similar to $\mathcal{F}_{h+1}$ in (5.3), where we clip f in $[0, 1]$, set $R(x', a) = 0$ and take expectation with respect to $a \sim \text{unif}(\mathcal{A})$ instead of a maximum. The main conceptual difference over (5.3) is that the discriminator class $\mathcal{G}_{h+1}$ now incorporates reward information, which enables downstream planning. Using (5.6) and the low-rank MDP properties, we show that this function class satisfies approximate realizability and approximate completeness, so we can invoke results for FQI and obtain the following representation learning guarantee:

Theorem 5.3. *Fix* $\delta \in (0, 1)$ *and consider the setup in Theorem 5.4. If the features* $\bar{\phi}_{0:H-1}$ *learnt by* MOFFLE *satisfy the condition in* (5.6) *for all* h *with* $\varepsilon_{\text{apx}} = \widetilde{O}\left(\frac{\varepsilon^2 \eta_{\min}}{dH^4A^5}\right)$, *then for any reward function* $R \in \mathcal{R}$, *running* FQI *with the value function class* $\mathcal{Q}(\bar{\phi}, R)$ *in* (5.1) *and an exploratory dataset* $\mathcal{D}$, *returns a policy* $\hat{\pi}$, *which satisfies* $v_R^{\hat{\pi}} \ge v_R^* - \varepsilon$ *with probability at least* $1 - \delta$. *The total number of episodes collected by* MOFFLE *in line 6 is:*

$$\widetilde{O}\left(\frac{H^7 d^2 A^5 \log(|\Phi||\mathcal{R}|B/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

5.4 Min-max-min representation learning

In this section, we describe our novel representation learning objective used to learn $\hat{\phi}_h$ and $\bar{\phi}_h$. The key insight is that the low-rank property of the MDP $\mathcal{M}$ can be used to learn a feature map $\hat{\phi}$ which can approximate the Bellman backup of all linear functions under feature maps $\phi \in \Phi$, and that approximating the backups of these functions enables subsequent near-optimal planning.

We present the algorithm with an abstract discriminator class $\mathcal{V} \subset (\mathcal{S} \to [0, L])$ that is instantiated either with $\mathcal{F}_{h+1}$ (with $L = 1$) or $\mathcal{G}_{h+1}$ (with $L = H$) defined previously. In order to describe our objective, it is helpful to introduce the shorthand

$$\text{b\_err}\left(\pi_h, \phi_h, v; B\right) = \min_{\|w\|_2 \le B} \mathbb{E}_{\pi_h}\left[\left(\langle \phi_h(s_h, a_h), w\rangle - \mathbb{E}\left[v(s_{h+1})|s_h, a_h\right]\right)^2\right] \tag{5.7}$$

for any policy π_h, feature ϕ_h, function v, and constant B, which we set so that $B \ge L\sqrt{d}$. This is the error in approximating the conditional expectation of $v(s_{h+1})$ using linear functions in the features $\phi_h(s_h, a_h)$. For approximating backups of all functions $v \in \mathcal{V}$, we seek feature $\hat{\phi}_h$ which minimizes $\max_{v\in\mathcal{V}} \text{b\_err}\left(\rho^{+3}_{h-3}, \hat{\phi}_h, v; B\right)$ up to an error of ε_{tol}.

Unfortunately, the quantity $\text{b\_err}\left(\cdot\right)$ contains a conditional expectation inside the square loss, so we cannot estimate it from samples $(s_h, a_h, s_{h+1}) \sim \rho^{+3}_{h-3}$. This is an instance of the well-known double sampling issue (Baird, 1995; Antos et al., 2008). Instead, we introduce the loss function:

$$\mathcal{L}_{\rho^{+3}_{h-3}}(\phi_h, w, v) = \mathbb{E}_{\rho^{+3}_{h-3}}\left[\left(\langle \phi_h(s_h, a_h), w\rangle - v(s_{h+1})\right)^2\right],$$

which is amenable to estimation from samples. However this loss function contains an undesirable conditional variance term, since via the bias-variance decomposition: $\mathcal{L}_{\rho^{+3}_{h-3}}(\phi_h, w, v) = \text{b\_err}\left(\rho^{+3}_{h-3}, \phi_h, v; B\right) + \mathbb{E}_{\rho^{+3}_{h-3}}\left[\mathbb{V}\left[v(s_{h+1}) \mid s_h, a_h\right]\right]$. This excess variance term can lead the agent to erroneously select a bad feature $\hat{\phi}_h$, since discriminators $v \in \mathcal{V}$ with low conditional variance will be ignored. However, via Lemma 5.1, we can rewrite the conditional variance as $\mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_v, v)$ for some $\|\theta^*_v\|_2 \le L\sqrt{d}$. Therefore, with $\mathcal{L}_{\mathcal{D}}$ as an empirical estimate of $\mathcal{L}_{\rho^{+3}_{h-3}}$ using dataset $\mathcal{D}$, we optimize the following objective which includes a variance correction term:

$$\operatorname*{argmin}_{\phi\in\Phi_h} \max_{v\in\mathcal{V}} \left\{ \min_{\|w\|_2\le B} \mathcal{L}_{\mathcal{D}}(\phi, w, v) - \min_{\tilde{\phi}\in\Phi_h, \|\tilde{w}\|_2 \le L\sqrt{d}} \mathcal{L}_{\mathcal{D}}(\tilde{\phi}, \tilde{w}, v)\right\}. \tag{5.8}$$

where we set the constant $B \ge L\sqrt{d}$.

We now state our first result which is an information-theoretic result and assumes that an oracle solution to the objective in (5.8) is available while running MOFFLE. The overall sample complexity of MOFFLE with oracle access is as follows:

Theorem 5.4. *Fix $\delta \in (0,1)$ and consider an MDP $\mathcal{M}$ that satisfies Definition 5.1 and Assumption 5.1, Assumption 5.2 hold. If an oracle solution to* (5.8) *is available, then with probability at least $1-\delta$ and appropriate values for each constant,* MOFFLE *returns an exploratory dataset $\mathcal{D}$ s.t. for any $R \in \mathcal{R}$, running* FQI *with value function class $\mathcal{Q}(\bar{\phi}, R)$ returns an ε-optimal policy for MDP $\mathcal{M}$. The total number of episodes used by the algorithm is:*

$$\widetilde{O}\left(\frac{H^6 d^8 A^{13} \log(|\Phi|/\delta)}{\eta_{\min}^5} + \frac{H d^9 A^{12} \log(|\Phi|/\delta_e)}{\eta_{\min}^6} + \frac{H^7 d^3 A^5 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

We provide a complete proof of the result in Section 5.10.3.

5.5 Iterative greedy representation learning

The min-max-min objective in the previous section in (5.8) is not provably computationally tractable for non-enumerable and non-linear function classes. However, recent empirical work (Lin et al., 2020) has considered a heuristic approach for solving similar min-max-min objectives by alternating between updating the outer min and inner max-min components. In this section, we show that a similar iterative approach that alternates between a squared loss minimization problem and a max-min objective in each iteration can be used to provably solve our representation learning problem.

This iterative procedure is displayed in Algorithm 5.4. Given the discriminator class $\mathcal{V}$, the algorithm grows finite subsets $\mathcal{V}^1, \mathcal{V}^2, \ldots \subset \mathcal{V}$ in an incremental and greedy fashion with $\mathcal{V}^1 = \{v_1\}$ initialized arbitrarily. In the t^{th} iteration, we have discriminator class $\mathcal{V}^t$ and we estimate a feature $\hat{\phi}_{t,h}$ which has low total squared loss with respect to all functions in $\mathcal{V}^t$ (line 6). Importantly the total square loss (sum) avoids the double sampling issue that arises with the worst case loss over class $\mathcal{V}^t$ (max), so no correction term is required. Next, we try to certify that $\hat{\phi}_{t,h}$ is a good representation by searching for a *witness* function $v_{t+1} \in \mathcal{V}$ for which $\hat{\phi}_{t,h}$ has large excess square loss (line 7). The optimization problem in (5.9) does require a correction term to address double sampling, but since $\hat{\phi}_{t,h}$ is fixed, it can be written as a simpler max-min program, when compared with the oracle approach. If the objective value here is smaller than some threshold, then our certification successfully verifies that $\hat{\phi}_{t,h}$ can approximate the Bellman backup of all functions in $\mathcal{V}$, so we terminate and output $\hat{\phi}_{t,h}$. On the other hand, if the objective is large, we add the witness v_{t+1} to our growing discriminator class and advance to the next iteration.

One technical point worth noting is that in (5.9) we relax the norm constraint on w to allow it to grow with $\sqrt{t}$. This is required by our iteration complexity analysis which shows that the procedure provably terminates in finitely many iterations with a polynomial sample complexity. We state the iteration complexity result below:

Lemma 5.5. *Fix $\delta_1 \in (0,1)$. If the greedy feature selection algorithm (Algorithm 5.4) is run with a sample $\mathcal{D}$ of size $n = \widetilde{O}\left(\frac{L^6 d^7 \log(|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{\varepsilon_{\mathrm{tol}}^3}\right)$, then with $B = \sqrt{\frac{13L^4d^3}{\varepsilon_{\mathrm{tol}}}}$, it terminates after $T = \frac{52L^2d^2}{\varepsilon_{\mathrm{tol}}}$ iterations and returns a feature $\hat{\phi}_h$ such that for $\mathcal{V} \subset (\mathcal{S} \to [0,L]) := \{v(s_{h+1}) = \mathrm{clip}_{[0,L]}(\mathbb{E}_{a_{h+1}\sim\pi_{h+1}(s_{h+1})}[R(s_{h+1},a_{h+1}) + \langle \phi_{h+1}(s_{h+1},a_{h+1}),\theta\rangle]) : \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \le L\sqrt{d}, R \in \mathcal{R}\}$, we have:*

$$\max_{v\in\mathcal{V}} \text{b\_err}\left(\rho_{h-3}^{+3}, \hat{\phi}_h, v; B\right) \le \varepsilon_{\mathrm{tol}}.$$

Algorithm 5.4 Feature Selection via Greedy Improvement

input: Feature class Φ_h, discriminator class $\mathcal{V}$, dataset $\mathcal{D}$ and tolerance $\varepsilon_{\mathrm{tol}}$.
Set $\mathcal{V}^0 \leftarrow \emptyset$ and choose $v_1 \in \mathcal{V}$ arbitrarily.
Set $\varepsilon_0 \leftarrow \varepsilon_{\mathrm{tol}}/52d^2$, $t \leftarrow 1$ and $l \leftarrow \infty$.
repeat
  Set $\mathcal{V}^t \leftarrow \mathcal{V}^{t-1} \cup \{v_t\}$.
  (Fit feature) Compute $\hat{\phi}_{t,h}$ as: $\hat{\phi}_{t,h}, W_t = \operatorname{argmin}_{\substack{\phi\in\Phi_h, W\in\mathbb{R}^{d\times t} \\ \|W\|_{2,\infty}\le L\sqrt{d}}} \sum_{i=1}^{t} \mathcal{L}_{\mathcal{D}}(\phi, W^i, v_i)$.
  (Find witness) Find test witness function:

$$v_{t+1} = \operatorname*{argmax}_{v\in\mathcal{V}} \max_{\tilde{\phi}\in\Phi_h, \|\tilde{w}\|_2\le L\sqrt{d}} \left(\min_{\|w\|_2 \le \frac{L\sqrt{dt}}{2}} \mathcal{L}_{\mathcal{D}}(\hat{\phi}_{t,h}, w, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}, \tilde{w}, v) \right). \tag{5.9}$$

  Set test loss l to the objective value in (5.9).
until $l < 24d^2\varepsilon_0 + \varepsilon_0^2$.
return Feature $\hat{\phi}_{T,h}$ from last iteration T.

We now state a polynomial sample complexity result for MOFFLE when Algorithm 5.4 is used as the feature learning sub-routine:

Theorem 5.6. *Fix $\delta \in (0,1)$ and consider the setup in Theorem 5.4. If (5.8) is solved via Algorithm 5.4, then with probability at least $1-\delta$ and appropriate values for each constant,* MOFFLE *returns an exploratory dataset $\mathcal{D}$ s.t. for any $R \in \mathcal{R}$, running* FQI *with value function class $\mathcal{Q}(\bar{\phi}, R)$ returns an ε-optimal policy for MDP $\mathcal{M}$. The total number of episodes used by the algorithm is:*

$$\widetilde{O}\left(\frac{H^6 d^{16} A^{31} \log(|\Phi|/\delta)}{\eta_{\min}^{11}} + \frac{H d^{17} A^{40} \log(|\Phi|/\delta)}{\eta_{\min}^{15}} + \frac{H^{19} d^{10} A^{15} \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^6 \eta_{\min}^{3}}\right).$$

While using Algorithm 5.4 does degrade the overall sample complexity when compared to the oracle approach, it leads to a more computationally viable algorithm. We provide a detailed analysis for the result in Section 5.10.4.

5.6 Enumerable feature class: A computionally tractable instance

In this section, we show that when Φ is a finite class and is efficiently enumerable, we can use Algorithm 5.2 to compute an exploratory policy cover in a computationally tractable manner. Note that the finite/enumerable feature class still leads to a value function class of infinite size (all linear functions of $\phi \in \Phi$). We show that for finite Φ, in Algorithm 5.2, we can learn $\hat{\phi}$ using a slightly different min-max-min objective:

$$\min_{\phi \in \Phi_h} \max_{f \in \mathcal{F}_{h+1}, \tilde{\phi} \in \Phi_h, \|\tilde{w}\|_2 \leq B} \left\{ \min_{\|w\|_2 \leq B} \mathcal{L}_{\mathcal{D}}(\phi, w, f) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}, \tilde{w}, f) \right\}, \tag{5.10}$$

where $\mathcal{F}_{h+1}$ is now the discriminator class that contains all *unclipped* functions f in form of $f(s_{h+1}) = \mathbb{E}_{\text{unif}(\mathcal{A})}\left[\langle \phi_{h+1}(s_{h+1}, a), \theta \rangle\right]$ where $\phi_{h+1} \in \Phi_{h+1}$ and $\|\theta\|_2 \leq \sqrt{d}$.

We derive a ridge regression based reduction of the min-max-min objective to eigenvector computation problems. To that end, consider the min-max-min objective where we fix $\phi, \tilde{\phi} \in \Phi_h$. Rewriting the objective for a sample of size n, we get the following updated objective:

$$\max_{f \in \mathcal{F}_{h+1}} \min_{\|w\|_2 \leq \sqrt{d}} \|Xw - f(\mathcal{D})\|_2^2 - \min_{\|\tilde{w}\|_2 \leq \sqrt{d}} \|\tilde{X}\tilde{w} - f(\mathcal{D})\|_2^2$$

where $X, \tilde{X} \in \mathbb{R}^{n \times d}$ are the covariate matrices for features ϕ and $\tilde{\phi}$ respectively. We overload the notation and use $f(\mathcal{D}) \in \mathbb{R}^n$ to denotes the value of any $f \in \mathcal{F}$ on the n samples. Now, instead of solving the constrained least squares problem, we use a ridge regression solution with regularization parameter λ. Thus, for any target f in the min-max objective, for feature ϕ, we get:

$$w_f = \left(\tfrac{1}{n} X^\top X + \lambda I_{d \times d}\right)^{-1} \left(\tfrac{1}{n} X^\top f(\mathcal{D})\right)$$
$$\|Xw - f(\mathcal{D})\|_2^2 = \|X \left(\tfrac{1}{n} X^\top X + \lambda I_{d \times d}\right)^{-1} \left(\tfrac{1}{n} X^\top f(\mathcal{D})\right) - f(\mathcal{D})\|_2^2 = \|A(\phi) f(\mathcal{D})\|_2^2$$

where $Y(\phi) = I_{n \times n} - X \left(\tfrac{1}{n} X^\top X + \lambda I_{d \times d}\right)^{-1} \left(\tfrac{1}{n} X^\top\right)$. In addition, any regression target f can be rewritten as $f = X'\theta$ for a feature $\phi' \in \Phi_{h+1}$ and $\|\theta\|_2 \leq \sqrt{d}$. Thus, for a fixed ϕ', ϕ and $\tilde{\phi}$, the maximization problem for $\mathcal{F}_{h+1}$ is the same as:

$$\max_{\|\theta\|_2 \leq \sqrt{d}} \theta^\top X'^\top \left(Y(\phi)^\top Y(\phi) - Y(\tilde{\phi})^\top Y(\tilde{\phi})\right) X'\theta. \tag{5.11}$$

where $X' \in \mathbb{R}^{n \times d}$ is again the sample matrix defined using $\phi' \in \Phi_{h+1}$. For each tuple of $(\phi, \tilde{\phi}, \phi')$, the maximization problem reduces to an eigenvector computation. As a result, we can efficiently solve the min-max-min objective in (5.10) by enumerating over each candidate feature in $(\phi, \tilde{\phi}, \phi')$

to solve

$$\min_{\phi\in\Phi_h} \max_{\tilde{\phi}\in\Phi_h,\phi'\in\Phi_{h+1},\|\theta\|_2\le\sqrt{d}} \theta^\top X'^\top \left(Y(\phi)^\top Y(\phi) - Y(\tilde{\phi})^\top Y(\tilde{\phi})\right) X'\theta. \tag{5.12}$$

While the analysis is more technical, (5.12) still allows us to plan using $\hat{\phi}_h$ in Algorithm 5.2 to guarantee that the policies ρ_{h-3}^{+3} are exploratory. We summarize the overall result in the following theorem:

Theorem 5.7. *Fix $\delta \in (0,1)$ and consider the setup in Theorem 5.4. In Algorithm 5.2, if $\hat{\phi}_h$ is learned using the eigenvector formulation* (5.12)*, then* MOFFLE *returns an exploratory dataset $\mathcal{D}$ such that for any $R \in \mathcal{R}$, running* FQI *with the collected dataset and the value function class in* (5.1) *returns an ε-optimal policy with probability at least $1-\delta$. The total number of episodes used by the algorithm is* $\mathrm{poly}(d, H, A, 1/\eta_{\min}, 1/\varepsilon, \log|\Phi|, \log|\mathcal{R}|, \log(1/\delta))$.

A detailed and formal theorem statement with its proof can be found in Section 5.10.5.

5.7 Related Work and Comparisons

Much recent attention has been devoted to linear function approximation (c.f., Jin et al., 2020; Yang and Wang, 2020). These results provide important building blocks for our work. In particular, the low-rank MDP model we study is from Jin et al. (2020) who assume that the feature map ϕ^* is known in advance. However, as we are focused on nonlinear function approximation, it is more apt to compare to related nonlinear approaches, which can be categorized in terms of their dependence on the size of the function class:

Polynomial in $|\Phi|$ approaches Many approaches, while not designed explicitly for our setting, can yield sample complexity scaling polynomially with $|\Phi|$ in our setup. Note, however, that polynomial-in-$|\Phi|$ scaling can be straightforwardly obtained by concatenating all of candidate feature maps and running the algorithm of Jin et al. (2020). This is the only obvious way to apply Eluder dimension results here (Osband and Van Roy, 2014; Wang et al., 2020b; Ayoub et al., 2020), and it also pertains to work on model selection (Pacchiano et al., 2020a; Lee et al., 2021). Indeed, the key observation that enables a logarithmic-in-$|\Phi|$ sample complexity is that all value function are in fact represented as *sparse* linear functions of this concatenated feature map.

However, exploiting sparsity in RL (and in contextual bandits) is quite subtle. In both settings, it is not possible to obtain results scaling logarithmically in both the ambient dimension *and* the number of actions (Lattimore and Szepesvári, 2020; Hao et al., 2021). That said, it is possible to obtain results scaling polynomially with the number of actions and logarithmically with the ambient dimension, as we do here.

Logarithmic in $|\Phi|$ approaches For logarithmic-in-$|\Phi|$ approaches, the assumptions and results vary considerably. Several results focus on the block MDP setting (Du et al., 2019a; Misra et al., 2020; Foster et al., 2020), where the dynamics are governed by a discrete latent state space, which is decodable from the observations. This setting is a special case of our low-rank MDP setting. Additionally, these works make stronger function approximation assumptions than we do. As such our work can be seen as generalizing and relaxing assumptions, when compared with existing block MDP results.

Most closely related to our work are the OLIVE and FLAMBE algorithms (Jiang et al., 2017; Agarwal et al., 2020b). OLIVE is a model-free RL algorithm that can be instantiated to produce a logarithmic-in-$|\Phi|$ sample complexity guarantee in precisely our setting (it also applies more generally). However, it is not computationally efficient even in tabular settings (Dann et al., 2018). In contrast, our algorithm involves a more natural minimax optimization problem that we show is computationally tractable in the "abstraction selection" case, which is even more general than the "known feature" setting (Section 5.6).

FLAMBE is computationally efficient, but it is model-based, so the function approximation assumptions are stronger than ours. Thus the key advancement is our weaker model-free function approximation assumption which does not require modeling μ^* whatsoever. On the other hand, FLAMBE does not require reachability assumptions as we do.

Related algorithmic approaches Central to our approach is the idea of embedding plausible futures into a "discriminator" class and using this class to guide the learning process. Bellemare et al. (2019) also propose a min-max representation learning objective using a class of *adversarial value functions*, but their work only empirically demonstrates its usefulness as an auxiliary task during learning and does not study exploration. Similar ideas of using a discriminator class have been deployed in model-based RL in Chapter 4 and in Farahmand et al. (2017); Sun et al. (2019); Ayoub et al. (2020), but the application to model-free representation learning and exploration is novel to our knowledge.

5.8 Discussion

Representation learning is a critical component of recent machine learning approaches and is also the backbone of many deep RL algorithms. Representation learning has been addressed in practice for RL via various methods as we discussed in the introduction: via auxiliary losses like inverse dynamics (Pathak et al., 2017), the use of explicit latent state space models (Hafner et al., 2019; Sekar et al., 2020), via bisimulation metrics (Gelada et al., 2019; Zhang et al., 2020), and contrastive learning (Laskin et al., 2020). All these methods use different underlying principles for learning representations but the questions of what representations are *sufficient* for planning has not been

answered precisely in previous literature. Here, we have addressed the question for the specific setting of linear MDPs using the following ideas.

Value-aware representation learning As highlighted in the previous chapter, the ability to estimate Bellman backups of value functions is often sufficient for computing near-optimal policies. In our case, we use the low-rank structure in the MDP to ensure this property for all possible value functions which are linear in the true feature representation. This idea, and thus the representation learning objective, can be extended to general value function classes to learn a representation under the min-max-min objective. Thus, the idea of value-function aware model learning in Farahmand et al. (2017) can also be used to derive novel representation learning objective for different settings (we use FQI in our framework).

Using feature based exploration In addition to representation learning, our work also addresses exploration by carefully combining the two algorithmic components. As such, we conjecture that our ideas can be used to improve and extend existing algorithms for exploration. For instance, in Pathak et al. (2017), the authors use prediction errors for the next state as a bonus for exploration. More similar to our work, Burda et al. (2018) uses prediction errors in random functions as bonuses for encouraging the agent to visit unexplored regions in the state space. Our work suggests that an adversarial choice of such functions from a sufficiently rich discriminator class allows the agent to learn provably useful representations. We leave the empirical study of such algorithms to future work.

Further, in our exploration scheme, we use an elliptic planner which explores all directions in the learnt feature. To our knowledge, such a bonus based algorithm has not been studied in the literature, as most methods simply add the prediction error as a reward bonus. Our proposed approach directly optimizes visitation in the feature space and can allow more efficient exploration. We leave the empirical study for this heuristic to future work as well.

5.9 Summary

In this chapter, we present, MOFFLE, a new model-free algorithm for representation learning and exploration in low-rank MDPs. We develop several representation learning schemes that vary in their computational and statistical properties, each yielding a different instantiation of the overall algorithm. Importantly MOFFLE can leverage a general function class Φ for representation learning, which provides it with the expressiveness and flexibility to scale to rich observation environments in a provably sample-efficient manner.

5.10 Proofs of Main Results

In this section, we give a detailed proof for the main results. We start by proving the results about the two basic components of MOFFLE and then show specific corollaries for each specific representation learning algorithm. In all our proofs, we will frequently use results about FQI planning, which we state for completeness in Appendix B.1.

5.10.1 Exploration and sample complexity results for Algorithm 5.2

5.10.1.1 Proof of Theorem 5.2

Theorem (Restatement of Theorem 5.2)**.** *Fix $\delta \in (0,1)$. Consider an MDP $\mathcal{M}$ that admits a low-rank factorization in Definition 5.1 and Assumption 5.1, Assumption 5.2 hold. If the features $\hat{\phi}_h$ learnt in line 5 of Algorithm 5.2 satisfy the condition in (5.3) for $B \geq \sqrt{d}$ and $\varepsilon_{\text{reg}} = \widetilde{\Theta}\left(\frac{\eta_{\min}^3}{d^2 A^9 \log^2(1+8/\beta)}\right)$, then with probability at least $1-\delta_e$, the sub-routine EXPLORE collects an exploratory mixture policy ρ_{h-3}^{+3} for each level h such that:*

$$\forall \pi : \mathbb{E}_{\pi}[f(s_h, a_h)] \leq \kappa A \mathbb{E}_{\rho_{h-3}^{+3}}[f(s_h, a_h)] \tag{5.13}$$

for any $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ and $\kappa = \frac{64 d A^4 \log(1+8/\beta)}{\eta_{\min}}$. The total number of episodes used in line 7 by Algorithm 5.2 is:

$$\widetilde{O}\left(\frac{H^5 d^6 A^{13} B^4 \log(|\Phi|/\delta_e)}{\eta_{\min}^5} + \frac{H d^6 A^{12} B^6 \log(|\Phi|/\delta_e)}{\eta_{\min}^6}\right).$$

The value of β is chosen such that $\beta \log(1+8/\beta) \leq \frac{\eta_{\min}^2}{128 d A^4 B^2}$.

Proof We will now prove the result assuming that the following condition from (5.3) is satisfied by $\hat{\phi}_h$ for all $h \in [H]$ with probability at least $1 - \delta_e/2$:

$$\max_{f \in \mathcal{F}_{h+1}} \min_{\|w\|_2 \leq B} \mathbb{E}_{\rho_{h-3}^{+3}}\left[\left(\left\langle \hat{\phi}_h(s_h, a_h), w \right\rangle - \mathbb{E}\left[f(s_{h+1}) | s_h, a_h\right]\right)^2\right] \leq \varepsilon_{\text{reg}}. \tag{5.14}$$

Now, let us turn to the inductive argument to show that the constructed policies ρ_{h-3}^{+3} are exploratory for every h. For our analysis, we will assume Lemma 5.15 stated in Section 5.10.6. We will establish the following inductive statement for each timestep h:

$$\forall z \in \mathcal{Z}_{h+1} : \max_{\pi} \mathbb{P}_{\pi}\left[z_{h+1} = z\right] \leq \kappa \mathbb{P}_{\rho_{h-2}^{+2}}\left[z_{h+1} = z\right]. \tag{5.15}$$

Assume that the exploration statement is true for all timesteps $h' \leq h$. We first show an error guarantee similar to (5.13) under distribution shift:

Lemma 5.8. *If the inductive assumption in* (5.15) *is true for all $h' \leq h$, then for all $v : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ we have:*

$$\forall \pi : \mathbb{E}_\pi[v(s_h, a_h)] \leq \kappa A \mathbb{E}_{\rho^{+3}_{h-3}}[v(s_h, a_h)] \tag{5.16}$$

Proof. Consider any timestep h and non-negative function v. Using the inductive assumption, we have:

$$\begin{aligned}
\mathbb{E}_\pi\left[v(s_h, a_h)\right] &= \sum_{z \in \mathcal{Z}_h} \mathbb{P}_\pi[z_h = z] \cdot \int \mathbb{E}_{\pi_h}[v(s_h, a_h)] \nu^*(s_h|z) d(s_h) \\
&\leq \kappa \sum_{z \in \mathcal{Z}_h} \mathbb{P}_{\rho^{+2}_{h-3}}[z_h = z] \cdot \int \mathbb{E}_{\pi_h}[v(s_h, a_h)] \nu^*(s_h|z) d(s_h) \\
&= \kappa \mathbb{E}_{\rho^{+2}_{h-3}}[\mathbb{E}_{\pi_h}[v(s_h, a_h)]] \\
&\leq \kappa A \mathbb{E}_{\rho^{+3}_{h-3}}[v(s_h, a_h)]
\end{aligned}$$

Therefore, the result holds for any policy π, timestep $h' \leq h$, and non-negative function v. □

As a result of Lemma 5.8, we have the same guarantee for the following squared-loss term for pair (s_h, a_h) for some $f \in \mathcal{F}_{h+1}$:

$$v(s_h, a_h) = \mathbb{E}_{\pi_h}\left[\left(\left\langle \hat{\phi}_h(s_h, a_h), w \right\rangle - \mathbb{E}\left[f(s_{h+1})|s_h, a_h\right]\right)^2\right].$$

Thus, using the feature learning guarantee in (5.14) along with (5.16), we have:

$$\forall f \in \mathcal{F}_{h+1} : \min_{\|w\|_2 \leq B} \mathbb{E}_\pi\left[\left(\left\langle \hat{\phi}_h(s_h, a_h), w \right\rangle - \mathbb{E}\left[f(s_{h+1})|s_h, a_h\right]\right)^2\right] \leq \kappa A \varepsilon_{\mathrm{reg}}.$$

We now outline our key argument to establish exploration: Fix a latent variable $z \in \mathcal{Z}_{h+1}$ and let $\pi := \pi_h$ be the policy which maximizes $\mathbb{P}_\pi[z_{h+1} = z]$. Thus, with $f(s_h, a_h) = \mathbb{P}[z_{h+1} = z|s_h, a_h]$ we have:

$$\begin{aligned}
\mathbb{E}_\pi\left[f(s_h, a_h)\right] &\leq A^2 \mathbb{E}_{\pi_{h-2} \circ \mathrm{unif}(\mathcal{A}) \circ \mathrm{unif}(\mathcal{A})}\left[f(s_h, a_h)\right] \\
&= A^2 \mathbb{E}_{\pi_{h-2}}\left[\mathbb{E}_{\mathrm{unif}(\mathcal{A})}\left[g(x_{h-1}, a_{h-1})|x_{h-2}, a_{h-2}\right]\right] \\
&\leq A^2 \mathbb{E}_{\pi_{h-2}}\left[\left|\langle \hat{\phi}_{h-2}(x_{h-2}, a_{h-2}), w_g \rangle\right|\right] + \sqrt{\kappa A^5 \varepsilon_{\mathrm{reg}}}.
\end{aligned} \tag{5.17}$$

The first inequality follows by using importance weighting on timesteps $h-1$ and h, where

we choose actions uniformly at random among $\mathcal{A}$. In the next step, we define $g(x_{h-1}, a_{h-1}) = \mathbb{E}_{\text{unif}(\mathcal{A})}[f(s_h, a_h)|x_{h-1}, a_{h-1}] = \left\langle \phi^*_{h-1}(x_{h-1}, a_{h-1}), \theta^*_f \right\rangle$ with $\|\theta^*_f\|_2 \le \sqrt{d}$. For (5.17), we use the result from Lemma 5.8 that $\hat{\phi}_{h-2}$ has small squared loss for the regression target specified by $g(\cdot)$ for a vector w_g with $\|w_g\|_2 \le B$. We further use the weighted RMS-AM inequality in the same step to bound the mean absolute error using the squared error bound.

Lemma 5.9. *If the FQI planner (Algorithm 5.3) is called with a sample of size* $n_{\text{ell}} := \widetilde{O}\left(\frac{H^4 d^3 \kappa \log(|\Phi|H/\delta_e)}{\beta^2}\right)$ *and total rollouts* $T \cdot n_{\text{plan}} := \widetilde{O}\left(\frac{d^3 \log \frac{1}{\delta_e}}{\beta^3} + \frac{H^4 d^3 \kappa A \log(|\Phi|/\delta_e)}{\beta^2}\right)$*, then for all* $h \in [H]$*, we have:*

$$\mathbb{E}_{\pi_{h-2}}\left[\left|\langle \hat{\phi}_{h-2}(x_{h-2}, a_{h-2}), w_g\rangle\right|\right] \le \frac{\alpha}{2}\mathbb{E}_{\rho_{h-2}}\left[\left(\langle \hat{\phi}_{h-2}(x_{h-2}, a_{h-2}), w_g\rangle\right)^2\right] + \frac{T\beta}{2\alpha} + \frac{\alpha\|w_g\|_2^2}{2T} + \frac{\alpha\beta\|w_g\|_2^2}{2}.$$

Proof. Applying Cauchy-Schwarz inequality followed by AM-GM, for any matrix $\widehat{\Sigma}$, we have:

$$\begin{aligned}\mathbb{E}_{\pi_{h-2}}\left[\left|\langle \hat{\phi}_{h-2}(x_{h-2}, a_{h-2}), w_g\rangle\right|\right] &\le \mathbb{E}_{\pi_{h-2}}\left[\left\|\hat{\phi}_{h-2}(x_{h-2}, a_{h-2})\right\|_{\widehat{\Sigma}^{-1}} \cdot \|w_g\|_{\widehat{\Sigma}}\right] \\ &\le \frac{1}{2\alpha}\mathbb{E}_{\pi_{h-2}}\left[\left\|\hat{\phi}_{h-2}(x_{h-2}, a_{h-2})\right\|^2_{\widehat{\Sigma}^{-1}}\right] + \frac{\alpha}{2}\|w_g\|^2_{\widehat{\Sigma}}.\end{aligned}$$

Here, we choose $\widehat{\Sigma}$ to be the (normalized) matrix returned by the elliptic planner in Algorithm 5.3. As can be seen in the algorithm pseudocode, $\widehat{\Sigma}$ is obtained by summing up a (normalized) identity matrix and the empirical estimates of the population covariance matrix $\Sigma_{\pi_\tau} = \mathbb{E}_{\pi_\tau}\hat{\phi}_{h-2}(x_{h-2}, a_{h-2})\hat{\phi}_{h-2}(x_{h-2}, a_{h-2})^\top$ where $\{\pi_\tau\}_{\tau \le T}$ are the T policies computed by the planner. Noting that ρ_{h-2} is a mixture of these T policies, we consider the following empirical and population quantities:

$$\Sigma_{\rho_{h-2}} = \frac{1}{T}\sum_{t=1}^{T}\Sigma_{\pi_t}, \qquad \Sigma = \Sigma_{\rho_{h-2}} + \frac{1}{T}I_{d\times d}, \qquad \widehat{\Sigma} = \frac{1}{T}\Gamma_T = \frac{1}{T}\sum_{i=1}^{T}\widehat{\Sigma}_{\pi_i} + \frac{1}{T}I_{d\times d}.$$

Now, we use the termination conditions satisfied by the elliptic planner (shown in Lemma 5.15) in

the following steps:

$$
\begin{aligned}
&\mathbb{E}_{\pi_{h-2}}\left[\left|\langle\hat{\phi}_{h-2}(x_{h-2},a_{h-2}),w_g\rangle\right|\right] \\
&\leq \frac{1}{2\alpha}\mathbb{E}_{\pi_{h-2}}\left[\left\|\hat{\phi}_{h-2}(x_{h-2},a_{h-2})\right\|^2_{\widehat{\Sigma}^{-1}}\right]+\frac{\alpha}{2}\left\|w_g\right\|^2_{\widehat{\Sigma}} && (5.18)\\
&\leq \frac{T\beta}{2\alpha}+\frac{\alpha}{2}\left\|w_g\right\|^2_{\widehat{\Sigma}} \\
&\leq \frac{T\beta}{2\alpha}+\frac{\alpha}{2}\left\|w_g\right\|^2_{\Sigma}+\frac{\alpha}{2}\beta\|w_g\|_2^2 && (5.19)\\
&= \frac{T\beta}{2\alpha}+\frac{\alpha}{2}\mathbb{E}_{\rho_{h-2}}\left[\left(\langle\hat{\phi}_{h-2}(x_{h-2},a_{h-2}),w_g\rangle\right)^2\right]+\frac{\alpha\|w_g\|_2^2}{2T}+\frac{\alpha\beta\|w_g\|_2^2}{2}.
\end{aligned}
$$

For the second inequality, note that $\frac{1}{T}\left\|\hat{\phi}_{h-2}(x_{h-2},a_{h-2})\right\|^2_{\widehat{\Sigma}^{-1}}$ is the reward function optimized by the FQI planner in the last iteration. Let $V_T(\pi)$ denote the expected value of a policy π for this reward function and MDP $\mathcal{M}$. From the termination condition and the results for the FQI planner in Lemma 5.15, we get:

$$
\max_{\pi} V_T(\pi) \leq V_T(\pi_T)+\beta/8 \leq \hat{V}_T(\pi_T)+\beta/4 \leq \beta.
$$

Therefore, the first term on the rhs in (5.18) can be bounded by $T\beta/2\alpha$. In step (5.19), we use the estimation guarantee for $\Sigma=\Gamma_T/T$ for the FQI planner shown in Lemma 5.15. Then, in the last equality step, we expand the norm of w_g using the definition of Σ to arrive at the desired result.

The sample size requirements directly follow from the result in Lemma 5.15. □

Using Lemma 5.9 in (5.17), we get:

$$
\begin{aligned}
\mathbb{E}_{\pi}\left[f(s_h,a_h)\right] &\leq \frac{\alpha A^2}{2}\mathbb{E}_{\rho^{+2}_{h-2}}\left[\left(\langle\hat{\phi}_{h-2}(x_{h-2},a_{h-2}),w_g\rangle\right)^2\right]+\frac{\beta A^2 T}{2\alpha}+\frac{\alpha A^2\|w_g\|_2^2}{2T} \\
&\quad+\frac{\alpha\beta A^2\|w_g\|_2^2}{2}+\sqrt{\kappa A^5\varepsilon_{\text{reg}}} && (5.20)\\
&\leq \alpha A^2\mathbb{E}_{\rho^{+2}_{h-2}}\left[\left(\langle\phi^*_{h-2}(x_{h-2},a_{h-2}),\theta^*_g\rangle\right)^2\right]+\alpha\kappa A^3\varepsilon_{\text{reg}}+\frac{A^2T\beta}{2\alpha}+\frac{\alpha A^2\|w_g\|_2^2}{2T} \\
&\quad+\frac{\alpha\beta A^2\|w_g\|_2^2}{2}+\sqrt{\kappa A^5\varepsilon_{\text{reg}}}. && (5.21)
\end{aligned}
$$

The next inequality in (5.21) again uses the approximation guarantee for features $\hat{\phi}$ in (5.14) along with the inequality $(a+b)^2 \leq 2a^2+2b^2$. Finally, we note that the inner product inside the expectation is always bounded between $[0,1]$ which allows use to use the fact that $f(s)^2 \leq f(s)$ for

$f : \mathcal{S} \to [0, 1]$. Substituting the upper bound for $\|w_g\|_2$, we get:

$$\begin{aligned}
&\mathbb{E}_\pi \left[f(s_h, a_h) \right] \\
&\leq \alpha A^2 \mathbb{E}_{\rho_{h-2}^{+2}} \left[\left\langle \phi^*_{h-2}(x_{h-2}, a_{h-2}), \theta^*_g \right\rangle \right] + \alpha\kappa A^3 \varepsilon_{\text{reg}} + \sqrt{\kappa A^5 \varepsilon_{\text{reg}}} \\
&\quad + \frac{\beta A^2 T}{2\alpha} + \frac{\alpha\beta A^2 B^2}{2} + \frac{\alpha A^2 B^2}{2T} \\
&= \alpha A^2 \mathbb{P}_{\rho_{h-2}^{+2}} \left[z_{h+1} = z \right] + \alpha\kappa A^3 \varepsilon_{\text{reg}} + \sqrt{\kappa A^5 \varepsilon_{\text{reg}}} + \frac{\beta A^2 T}{2\alpha} + \frac{\alpha\beta A^2 B^2}{2} + \frac{\alpha A^2 B^2}{2T}. \qquad (5.22)
\end{aligned}$$

The equality step (5.22) follows by the definition of the function $g(\cdot)$.

We now set $\kappa \geq 2\alpha A^2$ in (5.22). Therefore, if we set the parameters $\alpha, \beta, \varepsilon_{\text{reg}}$ such that

$$\max \left\{ \alpha\kappa A^3 \varepsilon_{\text{reg}} + \sqrt{\kappa A^5 \varepsilon_{\text{reg}}}, \frac{\beta A^2 T}{2\alpha}, \frac{\alpha\beta A^2 B^2}{2}, \frac{\alpha A^2 B^2}{2T} \right\} \leq \eta_{\min}/8, \qquad (5.23)$$

(5.22) can be re-written as:

$$\max_\pi \mathbb{P}_\pi \left[z_{h+1} = z \right] \leq \frac{\kappa}{2} \mathbb{P}_{\rho_{h-2}^{+2}} \left[z_{h+1} = z \right] + \frac{\eta_{\min}}{2} \leq \kappa \mathbb{P}_{\rho_{h-2}^{+2}} \left[z_{h+1} = z \right]$$

where in the last step, we use Assumption 5.1. Hence, we prove the exploration guarantee in Theorem 5.2 by induction.

To find the feasible values for the constants, we first note that $T \leq 8d \log(1 + 8/\beta)/\beta$ (Lemma 5.15). We start by setting $\frac{\beta A^2 T}{2\alpha} = \eta_{\min}/8$ which gives $\alpha/T = \frac{4\beta A^2}{\eta_{\min}}$. Using the upper bound on T, we get $\alpha \leq \frac{32dA^2 \log(1+8/\beta)}{\eta_{\min}}$. Next, we set the term $\alpha\kappa A^3 \varepsilon_{\text{reg}} + \sqrt{\kappa A^5 \varepsilon_{\text{reg}}} \leq \eta_{\min}/8$. Using the value of $\kappa = 2\alpha A^2$ we get:

$$2\alpha^2 A^5 \varepsilon_{\text{reg}} + \sqrt{2\alpha A^7 \varepsilon_{\text{reg}}} \leq \eta_{\min}/8,$$

which is satisfied by $\varepsilon_{\text{reg}} = \Theta\left(\frac{\eta_{\min}^3}{d^2 A^9 \log^2(1+8/\beta)} \right)$.

Lastly, we will consider the term $\frac{\alpha\beta A^2 B^2}{2}$ and by setting it less than $\eta_{\min}/8$, we get:

$$\beta \log(1 + 8/\beta) \leq \frac{\eta_{\min}^2}{128 d B^2 A^4}.$$

One can verify that under this condition we also have $\frac{\alpha A^2 B^2}{2T} \leq \eta_{\min}/8$, and setting $\beta = \widetilde{O}\left(\frac{\eta_{\min}^2}{dB^2 A^4} \right)$ satisfies the feasibility constraint for β. Here, we assume that B only has a $\text{poly}\log$ dependence on β and show later that this is true for all our feature selection methods. Notably, the only cases when B depends on β in our results is when $B = O\left(\frac{1}{\varepsilon_{\text{reg}}^c} \right)$ for a constant $c = \{1/2, 1\}$ which has a $\log^2(1 + 8/\beta)$ term.

Substituting the value of κ and β in Lemma 5.9 with an additional factor of H to account for all h gives us the final sample complexity bound in Theorem 5.2. Finally, the change of measure guarantee follows from the result in Lemma 5.8.

5.10.1.2 Improved sample complexity bound for simplex features

We can obtain more refined results when the agent instead has access to a latent variable feature class $\{\Psi_h\}_{h\in[H]}$ with $\psi_h : \mathcal{S} \times \mathcal{A} \to \Delta(d_{\mathrm{LV}})$. We call this the *simplex features* setting and show the improved results in this section. In order to achieve this improved result, we make two modifications to EXPLORE: 1) We use a smaller discriminator function class $\mathcal{F}_{h+1} := \{f(s_{h+1}, a_{h+1}) = \phi_{h+1}(s_{h+1}, a_{h+1})[i] : \phi_{h+1} \in \Phi_{h+1}, i \in [d_{\mathrm{LV}}]\}$ and 2) in EXPLORE, instead of calling the planner with learnt features $\hat{\phi}_{h-2}$ and taking three uniform actions, we plan for the features $\hat{\phi}_{h-1}$ and add two uniform actions to collect data for feature learning in timestep h. The key idea here is that instead of estimating the expectation of any bounded function f, we only need to focus on the expectation of coordinates of ϕ^* as included in class $\mathcal{F}_{h+1}$. Further, since $\phi^*_{h+1}[i]$ is already a linear function of the feature ϕ^*_{h+1}, we take only one action at random at timestep h.

Theorem 5.10 (Exploration with simplex features)**.** *Fix $\delta \in (0, 1)$. Consider an MDP $\mathcal{M}$ which admits a low-rank factorization with dimension d in Definition 5.1 and satisfies Assumption 5.1. If Assumption 5.2 holds, the features $\hat{\phi}_h$ learnt in line 5 in Algorithm 5.2 satisfy the condition in* (5.3) *for $B \geq \sqrt{d}$, and $\varepsilon_{\mathrm{reg}} = \widetilde{\Theta}\left(\frac{\eta^3_{\min}}{d^2A^5\log^2(1+8/\beta)}\right)$, then with probability at least $1-\delta_e$, the sub-routine* EXPLORE *collects an exploratory mixture policy ρ^{+3}_{h-3} for each level h such that:*

$$\forall \pi : \mathbb{E}_\pi[f(s_h, a_h)] \leq \kappa A \mathbb{E}_{\rho^{+3}_{h-3}}[f(s_h, a_h)] \tag{5.24}$$

for any $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ and $\kappa = \frac{64dA^2\log(1+8/\beta)}{\eta_{\min}}$. The total number of episodes used in line 7 by Algorithm 5.2 is:

$$\widetilde{O}\left(\frac{H^5d^6A^7B^4\log(|\Phi|/\delta_e)}{\eta^5_{\min}} + \frac{Hd^6A^6B^6\log(|\Phi|/\delta_e)}{\eta^6_{\min}}\right).$$

The value of β is chosen such that $\beta\log(1+8/\beta) \leq \frac{\eta^2_{\min}}{128dA^2B^2}$.

Proof. For simplex features, the key observation is that for any latent state $z \in \mathcal{Z}_{h+1}$, the function $f(s_h) = \mathbb{E}_{\mathrm{unif}(\mathcal{A})}[\Pr[z_{h+1} = z|s_h, a_h]]$ is already a member of the discriminator function class $\mathcal{F}_h := \{f(s_h) = \mathbb{E}_{\mathrm{unif}(\mathcal{A})}[\phi_h(s_h, a_h)[i]] : \phi_h \in \Phi_h, i \in [d_{\mathrm{LV}}]\}$. Thus, when we rewrite the term $\mathbb{E}_\pi[f(s_h, a_h)]$ as a linear function, we only need to backtrack one timestep to use the feature selection

guarantee:

$$\begin{aligned}\mathbb{E}_{\pi}\left[f(s_h,a_h)\right] &\leq A\mathbb{E}_{\pi_{h-1}\circ\,\mathrm{unif}(\mathcal{A})}\left[f(s_h,a_h)\right]\\ &= A\mathbb{E}_{\pi_{h-1}}\left[g(x_{h-1},a_{h-1})\right]\\ &\leq A\mathbb{E}_{\pi_{h-1}}\left[\left|\langle\hat{\phi}_{h-1}(s,a),w_g\rangle\right|\right]+\sqrt{\kappa A^3\varepsilon_{\mathrm{reg}}},\end{aligned} \tag{5.25}$$

where we define $g(x_{h-1},a_{h-1}) = \mathbb{E}_{\mathrm{unif}(\mathcal{A})}[f(s_h,a_h)|x_{h-1},a_{h-1}]$. Therefore, the new value of κ becomes $2\alpha A$ and by shaving off this A factor in the chain of inequalities, we get the following constraint set for the parameters:

$$\max\left\{\alpha\kappa A^2\varepsilon_{\mathrm{reg}}+\sqrt{\kappa A^3\varepsilon_{\mathrm{reg}}},\frac{\beta AT}{2\alpha},\frac{\alpha\beta AB^2}{2},\frac{\alpha AB^2}{2T}\right\}\leq\eta_{\min}/8, \tag{5.26}$$

Thus, the values of these parameters for the simplex features case are as follows:

$$\frac{\alpha}{T}=\frac{4\beta A}{\eta_{\min}},\quad \alpha\leq\frac{32dA\log(1+8/\beta)}{\eta_{\min}},\quad \varepsilon_{\mathrm{reg}}=\widetilde{\Theta}\left(\frac{\eta_{\min}^3}{d^2A^5\log^2(1+8/\beta)}\right).$$

Hence, the updated constraint for β is:

$$\beta\log\left(1+8/\beta\right)\leq\frac{\eta_{\min}^2}{64dB^2A^2}.$$

Other than the values for these parameters, the algorithm remains the same. Therefore, substituting the new values of κ and β in Lemma 5.9 as before, we get the improved sample complexity result. □

5.10.2 Proof of downstream lanning guarantee

We show that after obtaining the exploratory policies ρ_{h-3}^{+3} for all $h\in[H]$ using MOFFLE, we can collect a dataset $\mathcal{D}$ to learn a feature $\bar{\phi}_h\in\Phi_h$ for all levels and use FQI to plan for any reward function $R\in\mathcal{R}$. Specifically, we use the sub-routine LEARNREP to compute a feature $\bar{\phi}_h\in\Phi_h$ such that:

$$\max_{g\in\mathcal{G}_{h+1}}\min_{\|w\|_2\leq B}\mathbb{E}_{\rho_{h-3}^{+3}}\left[\left(\left\langle\bar{\phi}_h(s_h,a_h),w\right\rangle-\mathbb{E}\left[g(s_{h+1})|s_h,a_h\right]\right)^2\right]\leq\varepsilon_{\mathrm{apx}}, \tag{5.27}$$

where $\mathcal{G}_{h+1}\subset(\mathcal{S}\to[0,H])$ is the set of functions

$$g(s')=\mathrm{clip}_{[0,H]}\left(\max_a\left(R(s',a)+\langle\phi_{h+1}(s',a),\theta\rangle\right)\right),$$

with $R \in \mathcal{R}, \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq B$ and $B \geq H\sqrt{d}$. Using $\bar{\phi}_{0:H-1}$, we define the value function class $\mathcal{Q} = \{\mathcal{Q}_h(\bar{\phi}, R)\}_{h=0}^{H-1}$ for any given reward $R \in \mathcal{R}$. For a level $h \in [H]$, $\mathcal{Q}_h(\cdot)$ is defined as the set of following functions:

$$Q_h(s,a) = \text{clip}_{[0,H]}\left(R_h(s,a) + \langle \bar{\phi}_h(s,a), w\rangle\right)$$

The learnt feature serves two purposes as discussed in the main text:

- (*realizability*) The optimal value function for any timestep $h+1$ and reward $R_{h+1} \in \mathcal{R}$, is defined as $V^*_{h+1}(s') = \max_a\left(R_{h+1}(s',a) + \mathbb{E}[Q^*_{h+2}(\cdot)|s',a]\right) = \max_a\left(R_{h+1}(s',a) + \langle \phi^*_{h+1}, \theta^*_{h+1}\rangle\right)$. Thus, we have realizability as $V^*_{h+1} \in \mathcal{G}_{h+1}$, which in turn implies that $\exists f \in \mathcal{Q}_h$, s.t. $f \approx R_h + \mathbb{E}[V^*_{h+1}(\cdot)]$.

- (*completeness*) For completeness, note that $\mathcal{G}_{h+1}$ contains the Bellman backup of all possible $Q_{h+1}(\cdot)$ value functions we may encounter while running FQI with $\mathcal{Q}$ as defined above. Therefore, for any such Q_{h+1}, we have that $\exists f \in \mathcal{Q}_h$, s.t. $f \approx \mathcal{T}Q_{h+1}$.

Thus, the final sample complexity result for offline planning using FQI with value function class $\mathcal{Q}(\bar{\phi}, R)$ is as follows:

Theorem (Restatement of Theorem 5.3)**.** *Fix $\delta \in (0,1)$. Consider an MDP $\mathcal{M}$ that admits a low-rank factorization in Definition 5.1 and Assumption 5.1, Assumption 5.2 hold. If the features $\bar{\phi}_h$ learnt by* MOFFLE *satisfy the condition in* (5.6) *for all h with $\varepsilon_{\text{apx}} = \widetilde{O}\left(\frac{\varepsilon^2\eta_{\min}}{dH^4A^5}\right)$, then for any reward function $R \in \mathcal{R}$, running* FQI *with the value function class $\mathcal{Q}(\bar{\phi}, R)$ and an exploratory dataset $\mathcal{D}$, returns a policy $\hat{\pi}$ which satisfies $v^{\hat{\pi}}_R \geq v^*_R - \varepsilon$ with probability at least $1-\delta$. The total number of episodes collected by* MOFFLE *in line 6 is:*

$$\widetilde{O}\left(\frac{H^7d^2A^5\log(|\Phi||\mathcal{R}|B/\delta)}{\varepsilon^2\eta_{\min}}\right).$$

Proof. We run FQI with the learnt representation $\bar{\phi}_h$ using the value function class $\mathcal{Q}_h(\bar{\phi}_h, R_h)$ defined for each $h \in [H]$. Lemma B.6 shows that when (5.27) is satisfied with an error ε_{apx}, running FQI using a total of $n_h = \widetilde{O}\left(\frac{H^6d\kappa A\log(|\mathcal{R}|B/\delta')}{\beta^2}\right)$ episodes collected from each exploratory policy $\{\rho^{+3}_{h-3}\}$ returns a policy $\hat{\pi}$ which satisfies:

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=0}^{H-1} R_h(s_h,a_h)\right] \geq \max_\pi \mathbb{E}_\pi\left[\sum_{h=0}^{H-1} R_h(s_h,a_h)\right] - \beta - 2H^2\sqrt{\kappa A\varepsilon_{\text{apx}}}$$

with probability at least $1-\delta'$. Then union bounding over all possible $\bar{\phi}$, and setting $\delta = \delta'/|\Phi|$, $\beta = \varepsilon/2$, $\varepsilon_{\text{apx}} = \frac{\varepsilon^2}{16H^4\kappa A}$, we get the final planning result with a value error of ε and probability at

least $1-\delta$. Substituting $\kappa = \widetilde{O}\left(\frac{32dA^4}{\eta_{\min}}\right)$, we get $n_h = \widetilde{O}\left(\frac{H^6 d^2 A^5 \log(|\Phi||\mathcal{R}|B/\delta)}{\varepsilon^2 \eta_{\min}}\right)$. The final sample complexity is thus $n = \widetilde{O}\left(\frac{H^7 d^2 A^5 \log(|\Phi||\mathcal{R}|B/\delta)}{\varepsilon^2 \eta_{\min}}\right)$ where we sum up the collected episodes across all levels. □

5.10.3 Proofs for oracle representation learning

In this section, we present the sample complexity result for MOFFLE when a computational oracle FLO is available. Since we need to set $B \geq L\sqrt{d}$ in the min-max-min objective ((5.8)), we assume FLO solves (5.8) with $B = L\sqrt{d}$. The computational oracle is defined as follows:

Definition 5.3 (Minimax optimization oracle)**.** *Given a feature class Φ_h and an abstract discriminator class $\mathcal{V} \subset (\mathcal{S} \to [0, L])$, we define the minimax Feature Learning Oracle (*FLO*) as a subroutine that takes a dataset $\mathcal{D}$ of tuples (s_h, a_h, s_{h+1}) and returns a solution to the following objective:*

$$\hat{\phi}_h = \operatorname*{argmin}_{\phi \in \Phi_h} \max_{v \in \mathcal{V}} \left\{ \min_{\|w\|_2 \leq L\sqrt{d}} \mathcal{L}_{\mathcal{D}}(\phi, w, v) - \min_{\tilde{\phi} \in \Phi_h, \|\tilde{w}\|_2 \leq L\sqrt{d}} \mathcal{L}_{\mathcal{D}}(\tilde{\phi}, \tilde{w}, v) \right\}. \tag{5.28}$$

After defining FLO, we start by showing a sample complexity result for the min-max-min objective against a general discriminator function class $\mathcal{V}$ consisting of the set of functions

$$v(s_{h+1}) = \mathrm{clip}_{[0,L]}(\mathbb{E}_{a_{h+1} \sim \pi_{h+1}(s_{h+1})}[R(s_{h+1}, a_{h+1}) + \langle \phi_{h+1}(s_{h+1}, a_{h+1}), \theta \rangle])$$

where $\phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq L\sqrt{d}, R \in \mathcal{R}$ for any policy π_{h+1} over s_{h+1}. Note that, $\mathcal{F}_h$ in the main text uses a singleton reward class $R(s_{h+1}, a_{h+1}) = 0$ with $L = 1$ and $\pi_{h+1} = \mathrm{unif}(\mathcal{A})$. Similarly, $\mathcal{G}_h$ uses $L = H$ with π_{h+1} as the greedy arg-max policy.

Lemma 5.11 (Deviation bound for FLO)**.** *If the min-max feature learning objective is solved by the* FLO *for a sample of size n, then for $\mathcal{V} \subset (\mathcal{S} \to [0, L]) \coloneqq \{v(s_{h+1}) = \mathrm{clip}_{[0,L]}(\mathbb{E}_{a_{h+1} \sim \pi_{h+1}(s_{h+1})}[R(s_{h+1}, a_{h+1}) + \langle \phi_{h+1}(s_{h+1}, a_{h+1}), \theta \rangle]) : \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq L\sqrt{d}, R \in \mathcal{R}\}$, with probability at least $1-\delta$, we have:*

$$\max_{v \in \mathcal{V}} \mathrm{b\_err}\left(\rho^{+3}_{h-3}, \hat{\phi}_h, v; L\sqrt{d}\right) \leq \frac{512 L^2 d^2 \log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}.$$

Proof. Firstly, note the term $\mathrm{b\_err}\left(\rho^{+3}_{h-3}, \hat{\phi}_h, v; L\sqrt{d}\right)$ is a shorthand for

$$\min_{\|w\|_2 \leq L\sqrt{d}} \mathcal{L}_{\rho^{+3}_{h-3}}(\hat{\phi}_h, w, v) - \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_v, v)$$

where $\theta^*_v = \operatorname{argmin}_{\|\theta\|_2 \leq L\sqrt{d}} \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta, v)$.

Now, using the result in Lemma B.8 from Appendix B.2 and denoting $\mathcal{L}_{\rho_{h-3}^{+3}}(\cdot)$ as $\mathcal{L}(\cdot)$, with probability at least $1-\delta$, we have:

$$\begin{aligned}
&|\mathcal{L}(\phi, w, v) - \mathcal{L}(\phi^*, \theta_v^*, v) - (\mathcal{L}_{\mathcal{D}}(\phi, w, v) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_v^*, v))| \\
&\leq \frac{1}{2}\left(\mathcal{L}(\phi, w, v) - \mathcal{L}(\phi^*, \theta_v^*, v)\right) + \frac{128L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}
\end{aligned}$$

for all $\|w\|_2 \leq L\sqrt{d}$ ($B = L\sqrt{d}$), $\phi \in \Phi_h$ and $v \in \mathcal{V}$.

By the definition of FLO, for any $v \in \mathcal{V}$, we know that there exists θ_v^* such that (ϕ^*, θ_v^*) is the population minimizer $\operatorname{argmin}_{\tilde{\phi}\in\Phi_h, \|\tilde{w}\|_2 \leq L\sqrt{d}} \mathcal{L}(\tilde{\phi}, \tilde{w}, v)$. Using this and the concentration result, for all $v \in \mathcal{V}$, with $(\tilde{\phi}_v, \tilde{w}_v)$ as the solution of the innermost min in (5.28), we have:

$$\begin{aligned}
&\mathcal{L}_{\mathcal{D}}(\phi^*, \theta_v^*, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v) \\
&\leq \frac{3}{2}\left(\mathcal{L}(\phi^*, \theta_v^*, v) - \mathcal{L}(\tilde{\phi}_v, \tilde{w}_v, v)\right) + \frac{128L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n} \\
&\leq \frac{128L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}.
\end{aligned}$$

Now, for the oracle solution $\hat{\phi}$, for all $v \in \mathcal{V}$, we have:

$$\begin{aligned}
&\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v) \\
&= \mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_v^*, v) + \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_v^*, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v) \\
&\geq \frac{1}{2}\left(\mathcal{L}(\hat{\phi}, \hat{w}_v, v) - \mathcal{L}(\phi^*, \theta_v^*, v)\right) - \frac{128L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}.
\end{aligned}$$

Combining the two chains of inequalities, we get:

$$\begin{aligned}
&\mathcal{L}(\hat{\phi}, \hat{w}_v, v) - \mathcal{L}(\phi^*, \theta_v^*, v) \\
&\leq 2\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v)\right) + \frac{256L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n} \\
&\leq 2\max_{g\in\mathcal{V}}\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_g, g) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_g, \tilde{w}_g, g)\right) + \frac{256L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n} \\
&\leq 2\max_{g\in\mathcal{V}}\left(\mathcal{L}_{\mathcal{D}}(\phi^*, \theta_g^*, g) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_g, \tilde{w}_g, g)\right) + \frac{256L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n} \\
&\leq \frac{512L^2d^2\log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}.
\end{aligned}$$

Hence, we have proved the desired result. □

Here, we explicitly give a result for the discriminator function classes used by MOFFLE. However, a similar result can be easily derived for a general discriminator class $\mathcal{V}$ with the dependence $\log N$

where N is either the cardinality or an appropriate complexity measure of $\mathcal{V}$.

5.10.3.1 Proof of Theorem 5.4

Theorem (Restatement of Theorem 5.4)**.** *Fix $\delta \in (0,1)$. Consider an MDP $\mathcal{M}$ that admits a low-rank factorization in Definition 5.1 and Assumption 5.1, Assumption 5.2 hold. If the* LEARNREP *sub-routine to learn features $\hat{\phi}_h$ in Algorithm 5.2 and $\bar{\phi}_h$ in Algorithm 5.1 is implemented using the oracle* FLO*,* MOFFLE *returns an exploratory dataset $\mathcal{D}$ such that for any $R \in \mathcal{R}$, running* FQI *with value function class $\mathcal{Q}(\bar{\phi}, R)$ returns an ε-optimal policy with probability at least $1-\delta$. The total number of episodes used by the algorithm is:*

$$\widetilde{O}\left(\frac{H^6 d^8 A^{13} \log(|\Phi|/\delta)}{\eta_{\min}^5} + \frac{H d^9 A^{12} \log(|\Phi|/\delta_e)}{\eta_{\min}^6} + \frac{H^7 d^3 A^5 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

Proof. In MOFFLE, we call the sub-routine LEARNREP twice for discriminator classes $\mathcal{F}$ and $\mathcal{G}$ with $L=1$ and $L=H$ respectively. Similarly, the error threshold given as input to LEARNREP are ε_{reg} and ε_{apx}.

Firstly, we consider learning $\hat{\phi}_h$ that satisfies (5.3) so that we can collect a exploratory dataset. Let $\mathcal{R} = \{\mathbf{0}\}$ and $L=1$ (i,e. consider $\mathcal{V} = \mathcal{F}$), for any level h, applying Lemma 5.11, we know that if

$$n \geq \frac{512 d^2 \log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta/(3H))}{\varepsilon_{\text{reg}}},$$

then condition (5.3) holds with probability at least $1-\delta/(3H)$. Setting $\varepsilon_{\text{reg}} = \widetilde{\Theta}\left(\frac{\eta_{\min}^3}{d^2 A^9 \log^2(1+8/\beta)}\right)$ we get:

$$n_{\text{exp}} = \widetilde{O}\left(\frac{d^2 \log(|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{\varepsilon_{\text{reg}}}\right) = \widetilde{O}\left(\frac{d^4 A^9 \log(|\Phi|/\delta)}{\eta_{\min}^3}\right).$$

Substituting the value $B = \sqrt{d}$ in Theorem 5.2, we get the sample complexity for the elliptic planner as:

$$\begin{aligned} n_{\text{ell}} + T \cdot n_{\text{plan}} &= \widetilde{O}\left(\frac{H^5 d^6 A^{13} B^4 \log(|\Phi|/\delta)}{\eta_{\min}^5} + \frac{H d^6 A^{12} B^6 \log(|\Phi|/\delta_e)}{\eta_{\min}^6}\right) \\ &= \widetilde{O}\left(\frac{H^5 d^8 A^{13} \log(|\Phi|/\delta)}{\eta_{\min}^5} + \frac{H d^9 A^{12} \log(|\Phi|/\delta_e)}{\eta_{\min}^6}\right). \end{aligned}$$

Then we consider learning $\bar{\phi}_h$ that satisfies (5.6). Let $L=H$ and consider original $\mathcal{R}$ (i.e. consider $\mathcal{V} = \mathcal{G}$). Similarly, for any level h, setting $\varepsilon_{\text{apx}} = \widetilde{O}\left(\frac{\varepsilon^2 \eta_{\min}}{d H^4 A^5}\right)$ and applying Lemma 5.11, we know

that if

$$n_{\text{rep}} = \widetilde{O}\left(\frac{H^6 d^3 A^5 \log\left(\frac{|\Phi||\mathcal{R}|}{\delta/(3H)}\right)}{\varepsilon^2 \eta_{\min}}\right) = \widetilde{O}\left(\frac{H^6 d^3 A^5 \log\left(\frac{|\Phi||\mathcal{R}|}{\delta}\right)}{\varepsilon^2 \eta_{\min}}\right),$$

then condition (5.6) is satisfied with prob ability at least $1 - \delta/3H$.

Notice that (5.6) holds and we collect an exploratory dataset by applying Theorem 5.2. Then Theorem 5.3 implies the required sample complexity for offline FQI planning with $\bar{\phi}_{0:H-1}$ to guarantee ε error with probability at least $1 - \delta/(3H)$ is:

$$n_{\text{fqi}} = \widetilde{O}\left(\frac{H^6 d^2 A^5 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

Finally, union bounding over $h \in [H]$, the final sample complexity is $H(n_{\text{exp}} + n_{\text{ell}} + n_{\text{rep}} + n_{\text{fqi}})$. Reorganizing these terms completes the proof. □

5.10.4 Proofs for greedy representation learning method

We again start by showing a sample complexity result for the feature learning (line 6) and witness computation (line 7) steps in Algorithm 5.4 for a general discriminator class. We will later use the result to show a feature selection guarantee for Algorithm 5.4 in Lemma 5.5.

Lemma 5.12 (Deviation bounds for Algorithm 5.4)**.** *Let* $\tilde{\varepsilon} = \frac{64d(B+L\sqrt{d})^2 \log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}$. *If Algorithm 5.4 is called with a dataset* $\mathcal{D}$ *of size* n *and termination loss cutoff* $3\varepsilon_1/2 + \tilde{\varepsilon}$*, then with probability at least* $1 - \delta$*, for all* $v \in \mathcal{V} \subset (\mathcal{S} \to [0, L]) := \{v(s_{h+1}) = \text{clip}_{[0,L]}(\mathbb{E}_{a_{h+1}\sim\pi_{h+1}(s_{h+1})}[R(s_{h+1}, a_{h+1}) + \langle\phi_{h+1}(s_{h+1}, a_{h+1}), \theta\rangle]) : \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq L\sqrt{d}, R \in \mathcal{R}\}$ *and* $t \leq T$*, we have:*

$$\sum_{i\leq t} \mathbb{E}_{\rho^{+3}_{h-3}}\left[\left(\hat{\phi}_{t,h}(s_h, a_h)^\top w_{t,i} - \phi^*_h(s_h, a_h)^\top \theta^*_i\right)^2\right] \leq t\tilde{\varepsilon}$$

$$\mathbb{E}_{\rho^{+3}_{h-3}}\left[\left(\hat{\phi}_{t,h}(s_h, a_h)^\top w - \phi^*_h(s_h, a_h)^\top \theta^*_{t+1}\right)^2\right] \geq \varepsilon_1$$

for all $\|w\|_2 \leq B_t$ *where* $w_{t,i} = \text{argmin}_{\|\tilde{w}\|\leq B_t} \mathcal{L}_{\mathcal{D}}(\hat{\phi}_{t,h}, \tilde{w}, v_i)$ *and* $\theta^*_i = \text{argmin}_{\|\tilde{w}\|_2\leq L\sqrt{d}} \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{w}, v_i)$ *where* $B_t = \frac{L\sqrt{dt}}{2}$ *in Algorithm 5.4.*

Further, at termination, the learnt feature $\hat{\phi}_{T,h}$ *satisfies:*

$$\max_{v\in\mathcal{V}} \text{b\_err}\left(\rho^{+3}_{h-3}, \hat{\phi}_{T,h}, v; B\right) \leq 3\varepsilon_1 + 4\tilde{\varepsilon}.$$

Proof. We again denote $\mathcal{L}_{\rho_{h-3}^{+3}}(\cdot)$ as $\mathcal{L}(\cdot)$ and set $\tilde{\varepsilon} = \frac{64d(B+L\sqrt{d})^2 \log(2n|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{n}$. Further, we remove the subscript h for simplicity unless not clear by context. We begin by using the result in Lemma B.8 such that, with probability at least $1-\delta$, for all $\|w\|_2 \leq B$ ($B \geq L\sqrt{d}$), $\phi \in \Phi_h$ and $v \in \mathcal{V}$, we have

$$|\mathcal{L}(\phi, w, v) - \mathcal{L}(\phi^*, \theta_v^*, v) - (\mathcal{L}_{\mathcal{D}}(\phi, w, v) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_v^*, v))| \leq \frac{1}{2}\left(\mathcal{L}(\phi, w, v) - \mathcal{L}(\phi^*, \theta_v^*, v)\right) + \frac{\tilde{\varepsilon}}{2}.$$

Thus, for the feature fitting step in line 6 of Algorithm 5.4 in iteration t, with probability at least $1-\delta$ we have:

$$\begin{aligned}
\sum_{v_i \in \mathcal{V}^t} \mathbb{E}\left[\left(\hat{\phi}_t^\top w_{t,i} - \phi^{*\top}\theta_i^*\right)^2\right] &= \sum_{v_i \in \mathcal{V}^t}\left(\mathcal{L}(\hat{\phi}_t, w_{t,i}, v_i) - \mathcal{L}(\phi^*, \theta_i^*, v_i)\right) \\
&\leq \sum_{v_i \in \mathcal{V}^t} 2\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}_t, w_{t,i}, v_i) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_i^*, v_i)\right) + |\mathcal{V}^t|\tilde{\varepsilon} \\
&\leq t\tilde{\varepsilon},
\end{aligned}$$

which means the first inequality in the lemma statement holds.

For the adversarial test function at iteration t with $B_t \leq B$, let $\bar{w} \coloneqq \operatorname{argmin}_{\|w\|_2 \leq B_t} \mathcal{L}_{\mathcal{D}}(\hat{\phi}_t, w, v_{t+1})$. Using the same sample size for the adversarial test function at each non-terminal iteration with loss cutoff c, for any vector $w \in \mathbb{R}^d$ with $\|w\|_2 \leq B_t$ we get:

$$\begin{aligned}
\mathbb{E}\left[\left(\hat{\phi}_t^\top w - \phi^{*\top}\theta_{t+1}^*\right)^2\right] &= \mathcal{L}(\hat{\phi}_t, w, v_{t+1}) - \mathcal{L}(\phi^*, \theta_{t+1}^*, v_{t+1}) \\
&\geq \frac{2}{3}\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}_t, w, v_{t+1}) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_{t+1}^*, v_{t+1})\right) - \frac{\tilde{\varepsilon}}{3} \\
&\geq \frac{2}{3}\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}_t, \bar{w}, v_{t+1}) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_{t+1}^*, v_{t+1})\right) - \frac{\tilde{\varepsilon}}{3} \\
&\geq \frac{2c}{3} + \frac{2}{3}\left(\min_{\tilde{\phi} \in \Phi_h, \|\tilde{w}\|_2 \leq L\sqrt{d}} \mathcal{L}_{\mathcal{D}}(\tilde{\phi}, \tilde{w}, v_{t+1}) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_{t+1}^*, v_{t+1})\right) - \frac{\tilde{\varepsilon}}{3} \\
&\geq \frac{2c}{3} + \frac{1}{3}\left(\mathcal{L}(\tilde{\phi}_{t+1}, \tilde{w}_{t+1}, v_{t+1}) - \mathcal{L}(\phi^*, \theta_{t+1}^*, v_{t+1})\right) - \frac{2\tilde{\varepsilon}}{3} \\
&\geq \frac{2c}{3} - \frac{2\tilde{\varepsilon}}{3}.
\end{aligned}$$

In the first inequality, we invoke Lemma B.8 to move to empirical losses. In the third inequality, we add and subtract the ERM loss over (ϕ, w) pairs along with the fact that the termination condition is not satisfied for v_{t+1}. In the next step, we again use Lemma B.8 for the ERM pair $(\tilde{\phi}_{t+1}, \tilde{w}_{t+1})$ for v_{t+1}.

Thus, if we set the cutoff c for test loss to $3\varepsilon_1/2 + \tilde{\varepsilon}$, for a non-terminal iteration t, for any

$w \in \mathbb{R}^d$ with $\|w\|_2 \leq B_t$, we have:

$$\mathbb{E}\left[\left(\hat{\phi}_t^\top w - \phi^{*\top}\theta^*_{t+1}\right)^2\right] \geq \varepsilon_1, \tag{5.29}$$

which implies the second inequality in the lemma statement holds.

At the same time, for the last iteration, for all $v \in \mathcal{V}$, the feature $\hat{\phi}_T$ satisfies:

$$\begin{aligned}
&\min_{\|w\|_2 \leq B} \mathbb{E}\left[\left(\hat{\phi}_T^\top w - \phi^{*\top}\theta^*_v\right)^2\right] \\
&\leq 2\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}_T, \hat{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta^*_v, v)\right) + \tilde{\varepsilon} \\
&\leq 2\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}_T, \hat{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v) + \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta^*_v, v)\right) + \tilde{\varepsilon} \\
&\leq 2\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}_T, \hat{w}_v, v) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_v, \tilde{w}_v, v)\right) + \tilde{\varepsilon} \\
&\leq 3\varepsilon_1 + 4\tilde{\varepsilon}.
\end{aligned}$$

This gives us the third inequality in the lemma, thus completes the proof. □

5.10.4.1 Proof of Lemma 5.5

Lemma (Restatement of Lemma 5.5)**.** *Fix $\delta_1 \in (0,1)$. If the greedy feature selection algorithm (Algorithm 5.4) is run with a sample $\mathcal{D}$ of size $n = \widetilde{O}\left(\frac{L^6 d^7 \log(|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{\varepsilon_{\text{tol}}^3}\right)$, then with $B = \sqrt{\frac{13L^4d^3}{\varepsilon_{\text{tol}}}}$, it terminates after $T = \frac{52L^2d^2}{\varepsilon_{\text{tol}}}$ iterations and returns a feature $\hat{\phi}_h$ such that for $\mathcal{V} \subset (\mathcal{S} \to [0, L]) \coloneqq \{v(s_{h+1}) = \text{clip}_{[0,L]}(\mathbb{E}_{a_{h+1}\sim\pi_{h+1}(s_{h+1})}[R(s_{h+1}, a_{h+1}) + \langle\phi_{h+1}(s_{h+1}, a_{h+1}), \theta\rangle]) : \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq L\sqrt{d}, R \in \mathcal{R}\}$, we have:*

$$\max_{v\in\mathcal{V}} \text{b\_err}\left(\rho^{+3}_{h-3}, \hat{\phi}_h, v; B\right) \leq \varepsilon_{\text{tol}}.$$

Proof. For ease of notation, we will not use the subscript ρ^{+3}_{h-3} in the expectations below ($\mathcal{L}(\cdot) \coloneqq \mathcal{L}_{\rho^{+3}_{h-3}}(\cdot)$). Similarly, we will use ϕ_t to denote feature $\phi_{t,h}(s_h, a_h)$ of iteration t and (s', a') for (s_{h+1}, a_{h+1}) unless required by context. Further, for any iteration t, let $W_t = [w_{t,1} \mid w_{t,2} \mid \ldots \mid w_{t,t}] \in \mathbb{R}^{d\times t}$ be the matrix with columns W_t^i as the linear parameter $w_{t,i} = \text{argmin}_{\|w\|_2\leq L\sqrt{d}} \mathcal{L}_{\mathcal{D}}(\hat{\phi}_{t,h}, w, v_i)$. Similarly, let $A_t = [\theta^*_1 \mid \theta^*_2 \mid \ldots \mid \theta^*_t]$.

In the proof, we assume that the total number of iterations T does not exceed $\frac{52L^2d^2}{\varepsilon_{\text{tol}}}$ and set parameters accordingly. We later verify that this assumption holds. Further, let $\tilde{\varepsilon} = \frac{\varepsilon^2_{\text{tol}}}{2704L^2d^3}$ and $\varepsilon_0 = T_{\max} \cdot \tilde{\varepsilon} = \frac{\varepsilon_{\text{tol}}}{52d}$.

To begin, based on the deviation bound in Lemma 5.12, we note that if the sample $\mathcal{D}$ in Algorithm 5.4 is of size $n = \widetilde{O}\left(\frac{L^6 d^7 \log(|\Phi_h||\Phi_{h+1}||\mathcal{R}|/\delta)}{\varepsilon_{\text{tol}}^3}\right)$ and the termination loss cutoff set to

$3\varepsilon_1/2 + \tilde{\varepsilon}$ such that, with probability at least $1 - \delta_1$, for all non-terminal iterations t we have:

$$\sum_{v_i \in \mathcal{V}^t} \mathbb{E}\left[\left(\hat{\phi}_t^\top W_t^i - \phi^{*\top} A_t^i\right)^2\right] \leq t\tilde{\varepsilon} \leq \varepsilon_0, \tag{5.30}$$

$$\mathbb{E}\left[\left(\hat{\phi}_t^\top w - \phi^{*\top} \theta_{t+1}^*\right)^2\right] \geq \varepsilon_1 \tag{5.31}$$

where $\tilde{\varepsilon}$ is an error term dependent on the size of $\mathcal{D}$ and w is any vector with $\|w\|_2 \leq B_t \leq B$. Further, when the algorithm does terminate, we get the loss upper bound to be $3\varepsilon_1 + 4\tilde{\varepsilon}$.

Using (5.30) and (5.31), we will now show that the maximum iterations in Algorithm 5.4 is bounded.

At round t, for functions $v_1, \ldots, v_t \in \mathcal{V}$ in Algorithm 5.4, let $\theta_i^* = \theta_{v_i}^*$ as before and further let $\Sigma_t = A_t A_t^\top + \lambda I_{d\times d}$. Using the linear parameter θ_{t+1}^* of the adversarial test function v_{t+1}, define $\hat{w}_t = W_t A_t^\top \Sigma_t^{-1} \theta_{t+1}^*$. For this $\hat{w}_t$, we can bound its norm as:

$$\|W_t A_t^\top \Sigma_t^{-1} \theta_{t+1}^*\|_2 \leq \|W_t\|_2 \|A_t^\top \Sigma_t^{-1}\|_2 \|\theta_{t+1}^*\|_2 \leq L^2 d\sqrt{\frac{t}{4\lambda}}.$$

Here $\|W_t\|_2 \leq L\sqrt{dt}$ and $\|\theta_{t+1}^*\|_2 \leq L\sqrt{d}$. Applying SVD decomposition and the property of matrix norm, $\|A_t^\top \Sigma_t^{-1}\|_2$ can be upper bounded by $\max_{i\leq d} \frac{\sqrt{\lambda_i}}{\lambda_i + \lambda} \leq \frac{1}{\sqrt{4\lambda}}$, where λ_i are the eigenvalues of $A_t A_t^\top$. Then noticing AM-GM inequality, we get $\|A_t^\top \Sigma_t^{-1}\|_2 \leq \sqrt{1/4\lambda}$.

Setting $B_t = L^2 d\sqrt{\frac{t}{4\lambda}}$, from (5.31), we have

$$\begin{aligned}
\varepsilon_1 \leq \mathbb{E}\left[\left(\hat{\phi}_t^\top \hat{w}_t - \phi^{*\top} \theta_{t+1}^*\right)^2\right] &= \mathbb{E}\left[\left(\hat{\phi}_t^\top W_t A_t^\top \Sigma_t^{-1} \theta_{t+1}^* - \phi^{*\top} \Sigma_t \Sigma_t^{-1} \theta_{t+1}^*\right)^2\right] \\
&\leq \|\Sigma_t^{-1} \theta_{t+1}^*\|_2^2 \cdot \mathbb{E}\left[\|\hat{\phi}_t^\top W_t A_t^\top - \phi^{*\top} \Sigma_t\|_2^2\right] \\
&\leq 2\|\Sigma_t^{-1} \theta_{t+1}^*\|_2^2 \cdot \mathbb{E}\left[\|\hat{\phi}_t^\top W_t A_t^\top - \phi_t^\top A_t A_t^\top\|_2^2 + \lambda^2 \|\phi^{*\top}\|_2^2\right] \\
&\leq 2\|\Sigma_t^{-1} \theta_{t+1}^*\|_2^2 \cdot \left(\sigma_1^2(A_t) \mathbb{E}\left[\|\hat{\phi}_t^\top W_t - \phi^{*\top} A_t\|_2^2\right] + \lambda^2\right) \\
&\leq 2\|\Sigma_t^{-1} \theta_{t+1}^*\|_2^2 \cdot \left(L^2 dt\varepsilon_0 + \lambda^2\right).
\end{aligned}$$

The second inequality uses Cauchy-Schwarz. The last inequality applies the upper bound $\sigma_1(A_t) \leq L\sqrt{dt}$ and the guarantee from (5.30). Using the fact that $t \leq T$, this implies that

$$\|\Sigma_t^{-1} \theta_{t+1}^*\|_2 \geq \sqrt{\frac{\varepsilon_1}{2(L^2 dT\varepsilon_0 + \lambda^2)}}.$$

We now use the generalized elliptic potential lemma from Carpentier et al. (2020) to upper bound the total value of $\|\Sigma_t^{-1} \theta_{t+1}^*\|_2$. From Lemma B.9 in Appendix B.2.3, if $\lambda \geq L^2 d$ and we do not

terminate in T rounds, then

$$T\sqrt{\frac{\varepsilon_1}{2(L^2dT\varepsilon_0+\lambda^2)}} \le \sum_{t=1}^{T} \|\Sigma_t^{-1}\theta^*_{t+1}\|_2 \le 2\sqrt{\frac{Td}{\lambda}}.$$

From this chain of inequalities, we can deduce

$$T\varepsilon_1 \le 8(d/\lambda)\left(L^2dT\varepsilon_0+\lambda^2\right),$$

therefore

$$T \le \frac{8d\lambda}{\varepsilon_1 - 8L^2d^2\varepsilon_0/\lambda}. \tag{5.32}$$

Now, if we set $\varepsilon_1 = 16L^2d^2\varepsilon_0/\lambda$ in the above inequality, we can deduce that

$$T \le \frac{\lambda^2}{L^2d\varepsilon_0}.$$

Putting everything together, for input parameter ε_{tol}, the termination threshold for the loss l is set such that $\frac{48L^2d^2\varepsilon_0}{\lambda} + \frac{4L^2d\varepsilon_0^2}{\lambda^2} \le \varepsilon_{\text{tol}}$ which is satisfied for $\varepsilon_0 = \frac{\lambda\varepsilon_{\text{tol}}}{52L^2d^2}$. In addition, with $\lambda = L^2d$, we set the constants for Algorithm 5.4 as follows:

$$T \le \frac{52L^2d^2}{\varepsilon_{\text{tol}}}, \qquad \varepsilon_0 = \frac{\varepsilon_{\text{tol}}}{52d}, \qquad B_t \coloneqq \sqrt{\frac{L^2dt}{4}}, \qquad B \coloneqq \sqrt{\frac{13L^4d^3}{\varepsilon_{\text{tol}}}}.$$

Further, for Lemma 5.12, we set $\tilde{\varepsilon}$ to $\varepsilon_0/T = O\left(\frac{\varepsilon_{\text{tol}}^2}{L^2d^3}\right)$. □

5.10.4.2 Proof of Theorem 5.6

Theorem (Restatement of Theorem 5.6)**.** *Fix $\delta \in (0,1)$. Consider an MDP $\mathcal{M}$ that admits a low-rank factorization in Definition 5.1 and Assumption 5.1, Assumption 5.2 hold. If the* LEARNREP *sub-routine to learn features $\hat{\phi}_h$ in Algorithm 5.2 and $\bar{\phi}_h$ in Algorithm 5.1 is implemented using Algorithm 5.4,* MOFFLE *returns an exploratory dataset $\mathcal{D}$ such that for any $R \in \mathcal{R}$, running* FQI *with value function class $\mathcal{Q}(\bar{\phi}, R)$ returns an ε-optimal policy with probability at least $1-\delta$. The total number of episodes used by the algorithm is:*

$$\widetilde{O}\left(\frac{H^6d^{16}A^{31}\log(|\Phi|/\delta)}{\eta_{\min}^{11}} + \frac{Hd^{17}A^{40}\log(|\Phi|/\delta)}{\eta_{\min}^{15}} + \frac{H^{19}d^{10}A^{15}\log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^6\eta_{\min}^{3}}\right).$$

Proof. In MOFFLE, we call the sub-routine LEARNREP twice for discriminator classes $\mathcal{F}$ and $\mathcal{G}$

with $L = 1$ and $L = H$ respectively. Similarly, the error threshold given as input to LEARNREP are ε_{reg} and ε_{apx}. Using the result in Lemma 5.5, we know that for an approximation error of ε_{tol}, we need to set the sample size in Algorithm 5.4 to $n = \widetilde{O}\left(\frac{L^6 d^7 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon_{\text{tol}}^3}\right)$.

Setting the values of the parameter $\varepsilon_{\text{tol}} = \varepsilon_{\text{reg}} = \widetilde{\Theta}\left(\frac{\eta_{\min}^3}{d^2 A^9 \log^2(1+8/\beta)}\right)$ from Theorem 5.2 with $|\mathcal{R}| = 1$ and $L = 1$, we get the number of episodes per $h \in [H]$ for learning $\hat{\phi}_h$ that satisfies (5.3) as:

$$n_{\text{exp}} = \widetilde{O}\left(\frac{L^6 d^7 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon_{\text{reg}}^3}\right) = \widetilde{O}\left(\frac{d^{13} A^{27} \log(|\Phi|/\delta)}{\eta_{\min}^9}\right).$$

Substituting the value $B = \sqrt{\frac{13d^3}{\varepsilon_{\text{reg}}}}$ in Theorem 5.2, we get the sample complexity for the elliptic planner as

$$\begin{aligned} n_{\text{ell}} + T \cdot n_{\text{plan}} &= \widetilde{O}\left(\frac{H^5 d^6 A^{13} B^4 \log(|\Phi|/\delta)}{\eta_{\min}^5} + \frac{H d^6 A^{12} B^6 \log(|\Phi|/\delta_e)}{\eta_{\min}^6}\right) \\ &= \widetilde{O}\left(\frac{H^5 d^{16} A^{31} \log(|\Phi|/\delta)}{\eta_{\min}^{11}} + \frac{H d^{17} A^{40} \log(|\Phi|/\delta)}{\eta_{\min}^{15}}\right). \end{aligned}$$

Similarly, for learning $\bar{\phi}_h$ that satisfies (5.6), we set $\varepsilon_{\text{tol}} = \varepsilon_{\text{apx}} = \widetilde{O}\left(\frac{\varepsilon^2 \eta_{\min}}{dH^4 A^5}\right)$. Then, applying Lemma 5.12 with $L = H$, we get the number of episodes per $h \in [H]$ for learning $\bar{\phi}_h$ as:

$$n_{\text{rep}} = \widetilde{O}\left(\frac{L^6 d^7 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon_{\text{apx}}^3}\right) = \widetilde{O}\left(\frac{H^{18} d^{10} A^{15} \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^6 \eta_{\min}^3}\right).$$

Notice that (5.6) holds and we collect an exploratory dataset by applying Theorem 5.2. Then Theorem 5.3 implies the required sample complexity for offline FQI planning with $\bar{\phi}_{0:H-1}$ is

$$n_{\text{fqi}} = \widetilde{O}\left(\frac{H^6 d^2 A^5 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

The final result in the theorem statement is obtained by setting the bound to $H(n_{\text{exp}} + n_{\text{ell}} + n_{\text{rep}} + n_{\text{fqi}})$. □

5.10.5 Results for enumerable representation class

We first show that using the ridge estimator for an enumerable feature class as described in Section 5.6, discriminator class $\mathcal{F}_{h+1}$ and an appropriately set value of λ still allows us to establish a feature approximation result similar to FLO and greedy feature selection:

Lemma 5.13. *For the features selected via the ridge estimator, with $\lambda = \widetilde{\Theta}\left(\frac{1}{n^{1/3}}\right)$ for any function*

$f \in \mathcal{F}_{h+1}$*, with probability at least* $1-\delta$*, we have:*

$$\max_{f \in \mathcal{F}_{h+1}} \mathcal{L}_{\rho_{h-3}^{+3}}(\hat{\phi}_h, \hat{w}_f, f) - \mathcal{L}_{\rho_{h-3}^{+3}}(\phi_h^*, \theta_f^*, f) \leq \widetilde{O}\left(\frac{d^2 \log(|\Phi_h||\Phi_{h+1}|/\delta)}{n^{1/3}}\right)$$

where the discriminator function class $\mathcal{F}_{h+1} := \{f(s_{h+1}) = \mathbb{E}_{\text{unif}(\mathcal{A})}\left[\langle \phi_{h+1}(s_{h+1}, a), \theta\rangle\right] : \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \leq \sqrt{d}\}$.

Proof. We again denote $\mathcal{L}_{\rho_{h-3}^{+3}}(\cdot)$ as $\mathcal{L}(\cdot)$, $\mathcal{F}_{h+1}$ as $\mathcal{F}$. Firstly, as the discriminator function class $\mathcal{F}$ is defined without clipping, we now have: $\mathbb{E}[f(s')|s,a] = \langle \phi^*(s,a), \theta_f^*\rangle$ with $\|\theta_f^*\|_2 \leq d$. Also, the scale of the ridge estimator $w_f = \left(\frac{1}{n}X^\top X + \lambda I_{d\times d}\right)^{-1}\left(\frac{1}{n}X^\top f\right)$ now scales as $\frac{1}{\lambda}$. Now, similar to Lemma B.8, for all $\phi \in \Phi_h$, $\|w\|_2 \leq 1/\lambda$ and $f \in \mathcal{F}$, we have:

$$\begin{aligned}
&\left|\mathcal{L}(\phi, w, f) - \mathcal{L}(\phi^*, \theta_f^*, f) - \left(\mathcal{L}_{\mathcal{D}}(\phi, w, f) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_f^*, f)\right)\right| \\
\leq &\frac{1}{2}\left(\mathcal{L}(\phi, w, f) - \mathcal{L}(\phi^*, \theta_f^*, f)\right) + \frac{32d(1/\lambda + d)^2 \log(2n|\Phi_h||\Phi_{h+1}|/\delta)}{n}.
\end{aligned}$$

Now, let w_f^* denote the population ridge regression estimator for target $f \in \mathcal{F}$ for features ϕ^*. Assume $\lambda \leq 1/d$, which upper bounds the second term in the rhs above as $\gamma := \frac{128d\log(2n|\Phi_h||\Phi_{h+1}|/\delta)}{\lambda^2 n}$. For the selected feature $\hat{\phi}$, we have:

$$\begin{aligned}
&\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_f, \tilde{w}_f, f) \\
= &\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_f^*, f) + \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_f^*, f) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_f, \tilde{w}_f, f) \\
\geq &\frac{1}{2}\left(\mathcal{L}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}(\phi^*, \theta_f^*, f)\right) + \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_f^*, f) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_f, \tilde{w}_f, f) - \gamma \\
\geq &\frac{1}{2}\left(\mathcal{L}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}(\phi^*, \theta_f^*, f)\right) + \mathcal{L}_{\mathcal{D}}(\phi^*, \theta_f^*, f) - \mathcal{L}_{\mathcal{D}}(\phi^*, w_f^*, f) - \gamma.
\end{aligned}$$

Thus, with the feature selection output, we have:

$$
\begin{aligned}
&\mathcal{L}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}(\phi^*, \theta^*_f, f) \\
&\leq 2\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_f, \tilde{w}_f, f)\right) + 2\left(\mathcal{L}_{\mathcal{D}}(\phi^*, w^*_f, f) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta^*_f, f)\right) + 2\gamma \\
&\leq 2\max_{g\in\mathcal{F}}\left(\mathcal{L}_{\mathcal{D}}(\hat{\phi}, \hat{w}_g, g) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_g, \tilde{w}_g, g)\right) + 3\left(\mathcal{L}(\phi^*, w^*_f, f) - \mathcal{L}(\phi^*, \theta^*_f, f)\right) + 4\gamma \\
&\leq 2\max_{g\in\mathcal{F}}\left(\mathcal{L}_{\mathcal{D}}(\phi^*, w^*_g, g) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_g, \tilde{w}_g, g)\right) + 3\left(\mathcal{L}(\phi^*, w^*_f, f) - \mathcal{L}(\phi^*, \theta^*_f, f)\right) + 4\gamma \\
&\leq 2\max_{g\in\mathcal{F}}\left(\mathcal{L}_{\mathcal{D}}(\phi^*, w^*_g, g) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta^*_g, g)\right) + 2\max_{g\in\mathcal{F}}\left(\mathcal{L}_{\mathcal{D}}(\phi^*, \theta^*_g, g) - \mathcal{L}_{\mathcal{D}}(\tilde{\phi}_g, \tilde{w}_g, g)\right) \\
&\quad + 3\left(\mathcal{L}(\phi^*, w^*_f, f) - \mathcal{L}(\phi^*, \theta^*_f, f)\right) + 4\gamma \\
&\leq 2\max_{g\in\mathcal{F}}\left(\mathcal{L}_{\mathcal{D}}(\phi^*, w^*_g, g) - \mathcal{L}_{\mathcal{D}}(\phi^*, \theta^*_g, g)\right) + 3\left(\mathcal{L}(\phi^*, w^*_f, f) - \mathcal{L}(\phi^*, \theta^*_f, f)\right) + 6\gamma \\
&\leq 6\max_{g\in\mathcal{F}}\left(\mathcal{L}(\phi^*, w^*_g, g) - \mathcal{L}(\phi^*, \theta^*_g, g)\right) + 8\gamma.
\end{aligned}
$$

The third inequality uses the fact that $\hat{\phi}$ is the solution of the ridge-regression based feature selection objective. Further, in all steps, we repeatedly apply the deviation bound from Lemma B.8 to move from $\mathcal{L}_{\mathcal{D}}(\cdot)$ to $\mathcal{L}(\cdot)$.

Now, for ridge regression estimate w^*_g, we can bound the bias term on the rhs as follows:

$$
\begin{aligned}
\mathcal{L}(\phi^*, w^*_g, g) - \mathcal{L}(\phi^*, \theta^*_g, g) &= \mathbb{E}\left[\left(\left\langle \phi^*, w^*_g\right\rangle - \left\langle \phi^*, \theta^*_g\right\rangle\right)^2\right] \\
&= \|w^*_g - \theta^*_g\|^2_{\Sigma^*} = \sum_{i=1}^{d} \lambda_i \left\langle v_i, w^*_g - \theta^*_g\right\rangle^2 \\
&= \sum_{i=1}^{d} \lambda_i \left(\frac{\lambda_i}{\lambda + \lambda_i}\left\langle v_i, \theta^*_g\right\rangle - \left\langle v_i, \theta^*_g\right\rangle\right)^2 \\
&= \sum_{i=1}^{d} \frac{\lambda_i \lambda^2 \left\langle v_i, \theta^*_g\right\rangle^2}{(\lambda_i + \lambda)^2} \leq \frac{\lambda}{4}\|\theta^*_g\|_2^2 \leq \frac{\lambda d^2}{4},
\end{aligned}
$$

where (λ_i, v_i) denote the i-th eigenvalue-eigenvector pair of the population covariance matrix Σ^* for feature ϕ^*. In the derivation above, we use the fact that $w^*_g = (\Sigma + \lambda I)^{-1}\mathbb{E}[\phi^* g] = \frac{\lambda_i}{\lambda+\lambda_i}\left\langle v_i, \theta^*_g\right\rangle$. Therefore, the final deviation bound for $\hat{\phi}$ is:

$$
\mathcal{L}(\hat{\phi}, \hat{w}_f, f) - \mathcal{L}(\phi^*, \theta^*_f, f) \leq \frac{3\lambda d^2}{2} + \frac{1024 d \log(2n|\Phi_h||\Phi_{h+1}|/\delta)}{\lambda^2 n}.
$$

Thus, setting $\lambda = \widetilde{O}\left(\frac{1}{n^{1/3}}\right)$ gives the final result. □

5.10.5.1 Sample complexity of MOFFLE for enumerable feature class

Now that we have a feature selection guarantee assumed by the analysis of MOFFLE in Section 5.10.1, we show an overall sample complexity result for this version of MOFFLE as follows:

Theorem 5.14 (Restatement of Theorem 5.7)**.** *Fix $\delta \in (0,1)$. Consider an MDP $\mathcal{M}$ that admits a low-rank factorization in Definition 5.1 and Assumption 5.1, Assumption 5.2 hold. In Algorithm 5.2, if $\hat{\phi}_h$ is learned using the eigenvector formulation* (5.12)*, then* MOFFLE *returns an exploratory dataset $\mathcal{D}$ such that for any $R \in \mathcal{R}$, running* FQI *using the full representation class Φ returns an ε-optimal policy with probability at least $1-\delta$. The total number of episodes used by the algorithm is:*

$$\widetilde{O}\left(\frac{H^6 d^{22} A^{49} \log^5(|\Phi|/\delta)}{\eta_{\min}^{17}} + \frac{H d^{30} A^{67} \log^7(|\Phi|/\delta)}{\eta_{\min}^{24}} + \frac{H^7 d^3 A^3 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

Proof. In MOFFLE, we now use the eigenvector formulation for discriminator class $\mathcal{F}$ with the error threshold ε_{reg}. Setting the values of the parameter $\varepsilon_{\text{reg}} = \widetilde{\Theta}\left(\frac{\eta_{\min}^3}{d^2 A^9 \log^2(1+8/\beta)}\right)$ from Theorem 5.2 and the deviation bound in Lemma 5.13, we get the number of episodes per $h \in [H]$ for learning $\hat{\phi}_h$ as:

$$n_{\text{exp}} = \widetilde{O}\left(\frac{d^6 \log^3(|\Phi_h||\Phi_{h+1}|/\delta)}{\varepsilon_{\text{reg}}^3}\right) = \widetilde{O}\left(\frac{d^{12} A^{27} \log^3(|\Phi|/\delta)}{\eta_{\min}^9}\right).$$

Now, substituting the value $B = 1/\lambda = \widetilde{\Theta}\left(n_{\text{exp}}^{1/3}\right)$ in Theorem 5.2, we get the sample complexity for the elliptic planner as:

$$\begin{aligned} n_{\text{ell}} + T \cdot n_{\text{plan}} &= \widetilde{O}\left(\frac{H^5 d^6 A^{13} B^4 \log(|\Phi|/\delta)}{\eta_{\min}^5} + \frac{H d^6 A^{12} B^6 \log(|\Phi|/\delta_e)}{\eta_{\min}^6}\right) \\ &= \widetilde{O}\left(\frac{H^5 d^{22} A^{49} \log^5(|\Phi|/\delta)}{\eta_{\min}^{17}} + \frac{H d^{30} A^{67} \log^7(|\Phi|/\delta)}{\eta_{\min}^{24}}\right). \end{aligned}$$

Finally, using Corollary B.2 from Appendix B.1.2, the number of episodes collected for running FQI with the full representation class Φ can be bounded by:

$$n_{\text{fqi}} = \widetilde{O}\left(\frac{H^6 d^2 \kappa A \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2}\right) = \widetilde{O}\left(\frac{H^6 d^3 A^3 \log(|\Phi||\mathcal{R}|/\delta)}{\varepsilon^2 \eta_{\min}}\right).$$

The final result in the theorem statement is obtained by setting the bound to $H(n_{\text{exp}} + n_{\text{ell}}) + n_{\text{fqi}}$. □

5.10.6 The analysis of FQI based elliptical planner

In this section, we show the iteration complexity and the estimation guarantee for FQI based elliptical planner (Algorithm 5.3). The analysis follows a similar approach as Agarwal et al. (2020b), while the major difference here is that we apply FQI for the policy optimization step.

Lemma 5.15 (Estimation and iteration guarantees for Algorithm 5.3)**.** *If Algorithm 5.3 is run with a dataset of size* $n_{\text{ell}} = \widetilde{O}\left(\frac{H^4 d^3 \kappa A \log(|\Phi|/\delta)}{\beta^2}\right)$ *for a fix* $\beta > 0, \delta \in (0,1)$*, then upon termination, it outputs a matrix* Γ_T *and a policy* ρ *that with probability at least* $1-\delta$*:*

$$\forall \pi : \mathbb{E}_\pi \left[\hat{\phi}_{\tilde{H}-1}(s_{\tilde{H}-1}, a_{\tilde{H}-1})^\top (\Gamma_T)^{-1} \hat{\phi}_{\tilde{H}-1}(s_{\tilde{H}-1}, a_{\tilde{H}-1})\right] \le O(\beta),$$
$$\left\| \frac{\Gamma_T}{T} - \left(\mathbb{E}_\rho \left[\hat{\phi}_{\tilde{H}-1}(s_{\tilde{H}-1}, a_{\tilde{H}-1}) \hat{\phi}_{\tilde{H}-1}(s_{\tilde{H}-1}, a_{\tilde{H}-1})^\top\right] + \frac{I_{d\times d}}{T}\right)\right\|_{\text{op}} \le O(\beta/d).$$

Further, the iteration complexity is also bounded $T \le \frac{8d}{\beta} \log\left(1 + \frac{8}{\beta}\right)$*. The total number of rollouts used by the algorithm is then* $T \cdot n_{\text{plan}} := \widetilde{O}\left(\frac{d^3 \log \frac{1}{\delta}}{\beta^3}\right)$.

Proof. Since the policy optimization step (Algorithm B.1) is performed via an application of Lemma B.4, we can find an $\beta/8$-suboptimal policy π_t for the reward function induced by Γ_{t-1}. Then we use the sampling subroutine to estimate the value of this policy, which we denote $\hat{V}_t(\pi_t)$. As before, we terminate if $\hat{V}_t(\pi_t) \le 3\beta/4$. If we terminate in round t, we output $\rho = \text{unif}(\{\pi_i\}_{i=1}^{t-1})$ and we also output Γ_t. As notation, we use $V_t(\pi)$ to denote the value for policy π on the reward function used in iteration t, which is induced by Γ_{t-1}. We also denote the (element-wise) expectation of matrix $\hat{\Sigma}_{\pi_t}$ as Σ_{π_t}.

With $O\left(\text{poly}\left(d, H, A, T, 1/\eta_{\min}, \log|\Phi|, 1/\beta, \log(1/\delta)\right)\right)$ sample complexity, we can show that with probability at least $1-\delta$:

$$\max_{t\in[T]} \max \left\{ d \cdot \left\|\hat{\Sigma}_{\pi_t} - \Sigma_{\pi_t}\right\|_{\text{op}}, \left|\hat{V}_t(\pi_t) - V_t(\pi_t)\right|, \max_\pi V_t(\pi) - V_t(\pi_t) \right\} \le \beta/8. \tag{5.33}$$

For the first two terms in (5.33), we need to re-sample data in each iteration of the elliptic planner. As such, for each iteration, we can use standard concentration bounds to derive the per-iteration sample complexity to ensure the require estimation guarantees. Since the first term dominates the second term, we will only focus on that. Using standard concentration argument we can show the following guarantee:

Corollary 5.16 (Cor. 6.20 in Wainwright (2019))**.** *If the number of trajectories* n_{plan} *collected in*

each iteration by the elliptic planner in Algorithm 5.3 satisfies the following:

$$n_{\text{plan}} := \widetilde{O}\left(\frac{d^2 \log \frac{T}{\delta}}{\beta^2}\right),$$

then with probability at least $1 - \delta/2$*, for all* $t \le T$*, the estimates satisfy:*

$$\max_{t\in[T]} \max\left\{d \cdot \left\|\hat{\Sigma}_{\pi_t} - \Sigma_{\pi_t}\right\|_{\text{op}}, \left|\hat{V}_t(\pi_t) - V_t(\pi_t)\right|\right\} \le \beta/8. \tag{5.34}$$

The result follows directly from the concentration result, the fact that $\|\phi(s,a)\| \le 1$ and taking a union bound over all T iterations.

We now compute the number of samples used during FQI planning for the required error tolerance. Lemma B.4 states that for a sample of size n, the computed policy is sub-optimal by a value difference of order upto $\widetilde{O}\left(\sqrt{\frac{H^4 d^3 \kappa A \log(|\Phi|/\delta)}{n}}\right)$. Setting the failure probability of FQI planning to be $\delta/2HT$ for each level $h \in [H]$ and T iterations, and setting the planning error to $\beta/8$, we conclude that the total number of episodes n_{ell} used by Algorithm 5.3 for each timestep h is $\widetilde{O}\left(\frac{H^4 d^3 \kappa A \log(T|\Phi|/\delta)}{\beta^2}\right)$.

The accuracy guarantee for the covariance matrix Γ_T is straightforward, since each $\hat{\Sigma}_{\pi_t}$ is $\widetilde{O}(\beta/d)$ accurate and $\frac{\Gamma_T}{T} - \frac{I_{d\times d}}{T}$ is the average of such matrices.

Then, we turn to the iteration complexity. Now, if we terminate in iteration t, we know that $\hat{V}_t(\pi_t) \le 3\beta/4$. This implies

$$\max_{\pi} V_t(\pi) \le V_t(\pi_t) + \beta/8 \le \hat{V}_t(\pi_t) + \beta/4 \le \beta.$$

Similarly to the above inequality, we have

$$\begin{aligned} T\left(3\beta/4 - \beta/4\right) &\le \sum_{t=1}^{T} \hat{V}_t(\pi_t) - \beta/4 \le \sum_{t=1}^{T} V_t(\pi_t) - \beta/8 \\ &= \sum_{t=1}^{T} \mathbb{E}\left[\hat{\phi}_{\tilde{H}}^{\top} \Gamma_{t-1}^{-1} \hat{\phi}_{\tilde{H}} \mid \pi_t\right] - \beta/8 = \sum_{t=1}^{T} \operatorname{tr}(\Sigma_{\pi_t} \Gamma_{t-1}^{-1}) - \beta/8 \\ &\le \sum_{t=1}^{T} \operatorname{tr}(\hat{\Sigma}_{\pi_t} \Gamma_{t-1}^{-1}) \le 2d \log\left(1 + \frac{T}{d}\right). \end{aligned}$$

In the last step, we apply elliptical potential lemma (e.g. Lemma 26 of Agarwal et al. (2020b)).

Reorganizing the equation yields $T \le \frac{4d}{\beta} \log\left(1 + \frac{T}{d}\right)$. Further, if $T \le \frac{8d}{\beta} \log\left(1 + \frac{8}{\beta}\right)$, then we

have

$$T \leq \frac{4d}{\beta} \log\left(1 + \frac{T}{d}\right) \leq \frac{4d}{\beta} \log\left(1 + \frac{8 \log\left(1 + \frac{8}{\beta}\right)}{\beta}\right)$$
$$\leq \frac{4d}{\beta} \log\left(1 + \left(\frac{8}{\beta}\right)^2\right) \leq \frac{8d}{\beta} \log\left(1 + \frac{8}{\beta}\right).$$

Therefore, we obtain an upper bound on T by this set and guess approach.

Thus, the total sample complexity of the elliptic planner is: $T \cdot n_{\text{plan}} + n_{\text{ell}} = \widetilde{O}\left(\frac{d^3 \log \frac{1}{\delta}}{\beta^3} + \frac{H^4 d^3 \kappa A \log(|\Phi|/\delta)}{\beta^2}\right)$. □

CHAPTER 6

Provably Efficient Multi-Task Learning for Linear Quadratic Regulators

In the previous chapters of this thesis, we have considered different learning settings for fixed-horizon episodic MDPs. A common theme in the previous chapters was the use of feature representations of state and actions and an underlying linear structure in the underlying dynamics model. Both these components are naturally present in the field of optimal control. Optimal control has been influential in a range of applications, with a prominent example being robotics, where the underlying task has dynamics which are naturally determined by a (locally) linear dynamics function. Therefore, concurrent to the progress in learning in MDPs, we have also seen a flurry of non-asymptotic results for linear-quadratic control which is one of the fundamental problems in optimal control. In this chapter, we study a problem of concurrent and online adaptive control in multiple linear quadratic regulator systems. Following the main theme of this thesis, we again consider a structural assumption, where each LQR system is obtained as a linear function of a shared *basis* of dynamics matrices. Under this structural assumption, we first propose a joint estimator for the transition matrices of a linear time-invariant dynamical system and then build upon that result to propose and analyze a certainty equivalence based multi-task control algorithm for multiple LQR systems.

6.1 Introduction

The problem of adaptively controlling a given linear dynamical system represents a canonical problem in optimal control. Consequently, there has been a lot of work focused on this learning instance, ranging from the earlier work on stability and convergence (Ioannou and Sun, 2012; Krstic et al., 1995) to the recent works on finite time regret guarantees in online stochastic LQR control (Abbasi-Yadkori and Szepesvári, 2011; Faradonbeh et al., 2020b; Dean et al., 2018; Simchowitz and Foster, 2020). The problem of identifying the underlying transition matrices of the controlled

linear dynamical system is a key component in most adaptive control methods and has also been studied extensively in the literature. Recent works establish finite-time rates for accurately learning the dynamics in different online and offline settings (Faradonbeh et al., 2018a; Simchowitz et al., 2018; Sarkar and Rakhlin, 2019). The existing results are established assuming that the goal is to identify the transition matrix of *a single* dynamical system.

However, in many areas where LTIDS models (as in (6.1) below) are used, such as macroeconomics (Stock and Watson, 2016), functional genomics (Fujita et al., 2007), and neuroimaging (Seth et al., 2015), one observes multiple dynamical systems and needs to estimate the transition matrices for all of them jointly. Further, the underlying dynamical systems share commonalities, but also exhibit heterogeneity. For example, (Skripnikov and Michailidis, 2019a) analyze economic indicators of US states whose local economies share a strong manufacturing base. Moreover, in time course genetics experiments, one is interested in understanding the dynamics and drivers of gene expressions across related animal or cell line populations (Basu et al., 2015), while in neuroimaging, one has access to data from multiple subjects that suffer from the same disease (Skripnikov and Michailidis, 2019b).

In all these settings, there are remarkable similarities in the dynamics of the systems, but some degree of heterogeneity is also present. Hence, it becomes natural to pursue a joint learning and control strategy, by pooling the data of the underlying systems. This strategy should be particularly beneficial in settings, wherein the available data are limited (e.g., state trajectories are short), or the dimension of the systems is relatively large. The problem of joint learning (also referred to as multi-task learning) aims to study system identification methods that leverage similarities across systems. Similarly, pooling the data together across different systems can provide an improvement in the statistical efficiency of adaptive control.

In this chapter, we will consider an instance of such a multi-task estimation and control problem where the system matrices for each individual LQR system shares a common basis (see Assumption 6.1). This is analogous to the commonly used assumption in joint learning problems that all tasks share a common representation in the form of linear combinations of unknown parameters. The theoretical analysis presented in this chapter contributes to the understanding of joint learning for dynamical systems and provides novel results for a joint estimator for the system matrices, and finite time regret bounds for an adaptive controller which uses the joint estimator. In addition to these results, our work highlights the impact of the spectral properties of transition matrices on the estimation error for joint learning, and demonstrates a fundamental difference between settings with independent observations vs sequentially generated data (as applicable to linear dynamical systems).

Chapter outline. The rest of the chapter is organized as follows: In Section 6.2, we formally set up the problem and state the learning goal and assumptions. In the first part of this chapter, starting

in Section 6.3, we study the joint estimation problem for linear time-invariant dynamical systems and study the per-system estimation error, and provide the roles of various key quantities. Further, in Section 6.3.2, we investigate the impact of violations of the shared structure and then provide numerical illustrations in Section 6.3.3. In the second part, in Section 6.4, we study the adaptive control problem for our multi-task LQR setting and provide a certainty equivalence based adaptive controller with formal regret guarantees. Finally, we conclude with discussions and related work in Section 6.5. The detailed proofs for the formal results are provided in Section 6.8 (regret) and Section 6.9 (estimation).

6.2 Problem Setup: Shared Linear Basis

In this chapter, we study the problem of jointly controlling multipl linear quadratic regulator (LQR) systems. Specifically, in our multi-task setting, we consider a set of M LQR systems with states $x_m(t) \in \mathbb{R}^{d_x}$, such that each system $m \in [M]_+$ evolves in the following manner:

$$x_m(t+1) = A_m x_m(t) + B_m u_m(t) + \eta_m(t+1) \tag{6.1}$$

where $A_m \in \mathbb{R}^{d_x \times d_x}$ is the open loop system matrix and $B_m \in \mathbb{R}^{d_x \times d_u}$ is the gain matrix for control inputs $u_m(t) \in \mathbb{R}^{d_u}$. The sequence $\eta_m(t)$ is a zero-mean noise process. Our main goal here is to study how a shared structure between these M systems can allow efficient online control across these M systems. To this end, we make the following assumption on how the system matrices $[A_m|B_m]$ relate to each other:

Assumption 6.1 (Shared Basis for Control)**.** *For each system $m \in [M]_+$, the system matrices A_m and B_m can be expressed as a linear combination of a shared basis as follows:*

$$\Theta_m = \begin{bmatrix} A_m & B_m \end{bmatrix} = \sum_{i=1}^{k} \beta_m[i] W_i \tag{6.2}$$

where $\{W_i\}_{i=1}^k$ form a shared basis of matrices in $\mathbb{R}^{d_x \times d}$ and $\beta_m \in \mathbb{R}^k$ is the task-specific mixture coefficient[1].

Intuitively, we can see that if the size of the shared basis k is small enough, the effective dimensionality of the multi-task system is $O(kM + (d_x + d_u)d_x k)$ which can be significantly smaller than $(d_x + d_u)d_x M$. Therefore, an efficient joint learning and control procedure will exploit this low-dimensional representation and the shared structure, such that, for each system, the effective sample size increases by a factor of M for the shared components. In this chapter, we will propose

[1]Recall that we use $d = d_x + d_u$.

a certainty equivalence based online control procedure for the multi-task setting under the following assumption:

Assumption 6.2. *For each system, a stabilizing controller $K_{m,0}$ is given as an input to the agent. The controller $K_{m,0}$ can be arbitrarily sub-optimal compared to $K(\Theta_m)$.*

This requirement of having access to a stabilizing controller is not restrictive, and is common in the statistical results on online optimal control (Dean et al., 2018; Simchowitz and Foster, 2020). Further, offline strategies for obtaining such stabilizing controllers have been provided in Faradonbeh et al. (2018b).

The main goal for online multi-task learning in LQR will again be minimizing the *average* regret incurred across the M systems as compared to the optimal linear controller (see Section 2.3):

$$\text{Regret}\left(\{\Theta_m\}_{m=1}^{M}, T\right) = \frac{1}{M}\sum_{m=1}^{M} \text{Regret}\left(\Theta_m, T\right) \tag{6.3}$$

where $\text{Regret}(\Theta_m, T)$ is defined in (2.17). Recall that regret for an individual system is defined as follows:

$$\text{Regret}(\Theta, T) = \left[\sum_{t=0}^{T-1} c(x(t), u(t))\right] - \mathcal{J}^*(A, B)$$

where $c(x, u) = x^\top R_x x + u^\top R_u u$. We make the following assumptions for our regret analysis:

Assumption 6.3. *The noise process $\eta_m(t) \in \mathbb{R}^{d_x}$ is drawn iid as $\eta_m(t) \sim \mathcal{N}(0, 1)$ for all systems $m \in [M]_+$ and $t > 0$. We assume that each system (A_m, B_m) is* stabilizable *as defined in Definition 2.6. Further, without loss of generality, we assume $R_u = I$ and $R_x \succeq I$.*

For simplicity of notation, we will use the shorthand $P_m := P_\infty(A_m, B_m)$ and $K_m := K_\infty(A_m, B_m)$ and use $\mathcal{J}_m := \mathcal{J}^*(A_m, B_m)$ for the optimal cost. In Section 6.4, we will present our main guarantee as a multi-task regret bound which is based on our novel system identification analysis (see Section 6.3), combined with the certainty equiqualence based template from Simchowitz and Foster (2020). In the following sections, we formally study the different components of our algorithm and state the final guarantee in Section 6.4.

6.3 Joint Learning for Linear Time-Invariant Dynamical Systems

In this section, we will study the rates of jointly learning multiple linear time-invariant dynamical systems (LTIDS). Later on, we will show that similar results can be established for a multi-task

linear system in presence of control inputs. For simplicity, we will look at a closed loop system with states $x_m(t) \in \mathbb{R}^d$.

Specifically, here, the data consists of state trajectories of length T from M different systems. Let $x_m(t) \in \mathbb{R}^d$, $m \in [M]_+, t = 0, 1, \ldots, T$, denote the state trajectory of the m-th system, $m \in [M]_+$, that evolves according to the Vector Auto-regressive (VAR) process:

$$x_m(t+1) = A_m x_m(t) + \eta_m(t+1). \tag{6.4}$$

Above, $A_m \in \mathbb{R}^{d \times d}$ denotes the transition matrix of the m-th system and $\eta_m(t+1)$ is a mean zero noise process. Moreover, the transition matrices A_m are *related* as specified in Assumption 6.6.

The theoretical analysis presented in Section 6.3.1 shows that the specific issues pertaining to the Jordan forms of the transition matrices A_m are consequential for joint estimation of transition matrices in LTIDS and lead to major differences from multi-task learning with independent data.

6.3.1 Joint learning of LTIDS

We consider the problem of joint learning of M LTIDS, whose dynamic evolution is according to (6.4).

Assumption 6.4. *For all $m \in [M]_+$, the linear system A_m is non-explosive. That is, we have $\lambda_1(A_m) \leq 1 + \rho/T$, where $\rho > 0$ is a fixed constant independent of T.*

For succinctness, we use Θ^* to denote the set of M true transition matrices $\{A_m\}_{m=1}^M$.

To proceed, let $\mathcal{F}_t \coloneqq \sigma(\eta_1, \eta_2, \ldots, \eta_t, x_0, x_1, \ldots x_t)$ denote the filtration generated by the noise vectors and states variables until time t. Based on this, we impose the following assumption on the noise sequences[2].

Assumption 6.5. *For every $m \in [M]_+$, the noise sequence $\{\eta_m(t)\}_{t=1}^\infty$ is a martingale difference sequence; η_t is $\mathcal{F}_{t-}$-measurable, $\mathbb{E}[\eta_t | \mathcal{F}_{t-1}] = \mathbf{0}$, and $\mathbb{E}\left[\eta_m(t)\eta_m(t)^\top | \mathcal{F}_{t-1}\right] = C$. Further, $\eta_m(t)$ is sub-Gaussian; $\mathbb{E}[\exp(\langle \lambda, \eta_m(t) \rangle) | \mathcal{F}_{t-1}] \leq \exp(\|\lambda\|^2 \sigma^2/2)$, for all $\lambda \in \mathbb{R}^d$.*

We denote by $c^2 = \max(\sigma^2, \lambda_{\max}(C))$. The above assumption is widely-used in the finite-sample analysis of statistical learning methods for time evolving systems (Abbasi-Yadkori et al., 2011; Faradonbeh et al., 2020c). It includes normally distributed martingale difference sequences such that Assumption 6.5 is satisfied with $\sigma^2 = \lambda_{\max}(\Sigma)/2$. Moreover, whenever the components $\eta_m(t)[i]$ are mutually independent (given the filtration $\mathcal{F}_t$) and have sub-Gaussian distributions with constant σ_i, it suffices to let $\sigma^2 = \sum_{i=1}^d \sigma_i^2$.

[2]We establish the system identification results under a more general assumption of martingale noise and later simplify it to independent Gaussian noise for the control setting. The control results can be obtained for general martingale noise as well, which we don't write here for simplicity.

For a single LTIDS, its underlying transition matrices A_m can be *individually* identified from its own state trajectory data by using the least squares estimator (Faradonbeh et al., 2018a; Sarkar and Rakhlin, 2019). As mentioned in Section 6.2, we are interested in jointly learning the transition matrices of M systems, under the assumption that they share some common structure. For LTIDS, the analogous assumption can be states as follows:

Assumption 6.6 (Shared Basis)**.** *Each transition matrix A_m can be expressed as*

$$A_m = \sum_{i=1}^{k} \beta_m^*[i] W_i^*, \tag{6.5}$$

where $\{W_i^*\}_{i=1}^k$ are common matrices in $\mathbb{R}^{d\times d}$ and $\beta_m^* \in \mathbb{R}^k$ contains the idiosyncratic coefficients for the m-th system.

Later on in Section 6.3.2, we consider violations of the structure in (6.5) by allowing idiosyncratic additive factors for each system $m \in [M]_+$.

Based on the parameterization in (6.5), we solve for $\mathbf{W} = \{W_i\}_{i=1}^k$ and $B = [\beta_1|\beta_2|\cdots\beta_M] \in \mathbb{R}^{k\times M}$, as follows:

$$\widehat{\mathbf{W}}, \widehat{B} := \underset{\mathbf{W},B}{\operatorname{argmin}}\, \mathcal{L}(\Theta^*, \mathbf{W}, B),$$

where $\mathcal{L}(\Theta^*, \mathbf{W}, B)$ is the averaged squared loss across all M systems:

$$\frac{1}{MT}\sum_{m=1}^{M}\sum_{t=0}^{T}\left\| x_m(t+1) - \left(\sum_{i=1}^{k}\beta_m[i]W_i\right) x_m(t)\right\|_2^2. \tag{6.6}$$

The objective function in the minimization problem in (6.6) is non-convex and in the analysis, we assume that we have oracle access to the minimizer. Similar explicit parameterizations in least squares loss functions are studied in the non-convex optimization literature. It is shown that gradient descent (Ge et al., 2017) and alternating methods (Jain and Kar, 2017) converge to near-optimal minima. In the sequel, we establish key identification rates for the joint estimator in (6.6) and show that $\sum_{m=1}^{M}\left\|A_m - \widehat{A}_m\right\|_F^2 \to 0$, at a certain rate, with high probability. This generalizes recent results on multi-task learning in settings involving iid data (Du et al., 2020; Hu et al., 2021).

We focus on the estimation error utilizing the structure in Assumption 6.6. For ease of presentation, we rewrite the problem as a univariate regression one. To that end, we introduce some notation to express each transition matrix in vector form and rewrite (6.6), as follows.

First, for each state vector $x_m(t) \in \mathbb{R}^d$, we create d different covariates of size $\mathbb{R}^{d^2}$. So, for $j = 1, \cdots, d$, the vector $\tilde{x}_{m,j}(t) \in \mathbb{R}^{d^2}$ contains $x_m(t)$ in the j-th block of size d and $0's$ elsewhere.

Then, we express the system matrix $A_m \in \mathbb{R}^{d\times d}$ as a vector $\tilde{A}_m \in \mathbb{R}^{d^2}$. Similarly, the concatenation of all vectors $\tilde{A}_m$ can be coalesced into the matrix $\tilde{\Theta} \in \mathbb{R}^{d^2\times M}$. Analogously, $\tilde{\eta}_m(t)$ will denote the concatenated dt dimensional vector of noise vectors for system m. Thus, the structural assumption in (6.16) can be written as:

$$\tilde{A}_m = W^* \beta_m^*, \tag{6.7}$$

where $W^* \in \mathbb{R}^{d^2\times k}$ and $\beta_m^* \in \mathbb{R}^k$. Similarly, the overall parameter set can be factorized as $\tilde{\Theta}^* = W^* B^*$, where the matrix $B^* = [\beta_1^* | \beta_2^* | \cdots \beta_M^*] \in \mathbb{R}^{k\times M}$ contains the true weight vectors β_m^*.

Thus, expressing the system matrices A_m in this manner leads to a low rank structure in (6.7), so that the matrix $\tilde{\Theta}^*$ is of rank k. Using the vectorized parameters, the evolution for the components $j \in [d]_+$ of all state vectors $x_m(t)$ can be written as:

$$x_m(t+1)[j] = \tilde{A}_m \tilde{x}_{m,j}(t) + \eta_m(t+1)[j]. \tag{6.8}$$

For each system $m \in [M]_+$, we therefore have a total of dT samples, where the statistical dependence now follows a block structure: d covariates of $x_m(1)$ are all constructed using $x_m(0)$, next d using $x_m(1)$ and so forth. To estimate the parameters, we solve the following optimization problem:

$$\begin{aligned}\widehat{W}, \{\hat{\beta}_m\}_{m=1}^M &\coloneqq \underset{W,\{\beta_m\}_{m=1}^M}{\operatorname{argmin}} \underbrace{\sum_{m,t}\sum_{j=1}^{d} \left(x_m(t+1)[j] - \langle W\beta_m, \tilde{x}_{m,j}(t)\rangle\right)^2}_{\mathcal{L}(W,\beta)} \\ &= \underset{W,\{\beta_m\}_{m=1}^M}{\operatorname{argmin}} \sum_{m=1}^{M} \left\| y_m - \tilde{X}_m W \beta_m \right\|_2^2, \end{aligned} \tag{6.9}$$

where $y_m \in \mathbb{R}^{Td}$ contains all T state vectors stacked vertically and $\tilde{X}_m \in \mathbb{R}^{Td\times d^2}$ contains the corresponding matrix input. We denote the covariance matrices by $\Sigma_m = \sum_{t=0}^{T-1} X_m(t) X_m(t)^\top$ and $\tilde{\Sigma}_m = \sum_{t=0}^{T-1} \tilde{X}_m(t) \tilde{X}_m(t)^\top$, for the vectorized matrices.

The analysis relies on high probability bounds on the sample covariance matrices Σ_m. Technically, we utilize the Jordan forms of matrices, as described in Section 2.3. In this section, for matrix A_m, its Jordan decomposition is written as $A_m = P_m^{-1} \Lambda_m P_m$, where Λ_m is a block diagonal matrix; $\Lambda_m = \operatorname{diag}(\Lambda_{m,1}, \ldots \Lambda_{m,q_m})$, and for $i = 1, \ldots q_m$, each block $\Lambda_{m,i} \in \mathbb{C}^{l_{m,i}\times l_{m,i}}$ is a Jordan matrix of the eigenvalue $\lambda_{m,i}$. Further, we denote the size of each Jordan block by $l_{m,i}$, for $i = 1, \cdots, q_m$, and the size of the largest Jordan block for system m by l_m^*. Using this notation, we first define the

following quantity:

$$\alpha(A_m) \coloneqq \begin{cases} \left\|P_m^{-1}\right\|_{\infty\to 2} \left\|P_m\right\|_\infty f(\Lambda_m) & \lambda_{m,1} < 1, \\ \left\|P_m^{-1}\right\|_{\infty\to 2} \left\|P_m\right\|_\infty e^{\rho+1} & \lambda_{m,1} \leq 1 + \frac{\rho}{T}, \end{cases} \tag{6.10}$$

where

$$f(\Lambda_m) = e^{1/|\lambda_{m,1}|} \left[\frac{l_m^* - 1}{-\log|\lambda_{m,1}|} + \frac{(l_m^* - 1)!}{(-\log|\lambda_{m,1}|)^{l_{m,1}^*}} \right].$$

For some $\delta_C > 0$ that will be determined later on, for system m define $\bar{b}_m = b_T(\delta_C/3) + \|x_m(0)\|_\infty$, where $b_T(\delta) = \sqrt{2\sigma^2 \log(2dMT\delta^{-1})}$. Then, we establish the following high probability bounds on Σ_m.

Lemma 6.1 (Bounds on sample covariance matrices)**.** *For each system m, let $\underline{\Sigma}_m = \underline{\lambda}_m I$ and $\bar{\Sigma}_m = \bar{\lambda}_m I$, where $\underline{\lambda}_m \coloneqq 4^{-1}\lambda_{\min}(C)T$, and*

$$\bar{\lambda}_m \coloneqq \begin{cases} \alpha(A_m)^2 \bar{b}_m^2 T, & \text{if } \lambda_{m,1} < 1, \\ \alpha(A_m)^2 \bar{b}_m^2 T^{2l_m^*+1}, & \text{if } \lambda_{m,1} \leq 1 + \frac{\rho}{T}. \end{cases}$$

Then, there is T_0, such that for $m \in [M]_+$ and $T \geq T_0$:

$$\Pr\left[0 \prec \underline{\Sigma}_m \preceq \Sigma_m \preceq \bar{\Sigma}_m\right] \geq 1 - \delta_C. \tag{6.11}$$

Henceforth, we use $\mathcal{E}_C$ to refer to the event $\left\{0 \prec \underline{\Sigma}_m \preceq \Sigma_m \preceq \bar{\Sigma}_m\right\}$. Moreover, we use the notation $\bar{\lambda} = \max_m \bar{\lambda}_m$, $\underline{\lambda} = \min_m \underline{\lambda}_m$, $\boldsymbol{\kappa}_m = \bar{\lambda}_m/\underline{\lambda}_m$, $\boldsymbol{\kappa} = \max_m \boldsymbol{\kappa}_m$ and $\boldsymbol{\kappa}_\infty = \bar{\lambda}/\underline{\lambda}$. Note that $\boldsymbol{\kappa}_\infty > \boldsymbol{\kappa}$.

Proving the above high probability bound involves tools from existing work on identification of LTIDS. Specifically, we leverage the truncation-based arguments of Faradonbeh et al. (2018a) to establish upper-bounds using the constant $\alpha(A_m)$ that captures the effect of the spectral properties of the transition matrices on the state trajectory. Further, we use strategies based on self-normalized martingales, similar to the works of Abbasi-Yadkori et al. (2011) and Sarkar and Rakhlin (2019). A complete and detailed proof of the above lemma is provided in Section 6.9.2.

Lemma 6.1 provides a tight characterization of the sample covariance matrix for each system $m \in [M]_+$, in terms of the magnitude of eigenvalues of A_m, as well as the size of the Jordan matrices in the decomposition of A_m. Specifically, the upper bounds demonstrate that $\bar{\lambda}_m$ grows exponentially with the dimension d whenever $l_m^* = \Omega(d)$. Further, if A_m has eigenvalues with magnitudes close to 1, then scaling with the time T can be as large as T^d. Note that, the two expressions for $\bar{\lambda}_m$ are not contradictory as the first bound for $\lambda_{m,1} < 1$ can be tighter, whereas the second one is sharper for the case when $\lambda_{m,1} \to 1$. Our upper bounds in Lemma 6.1 are

more general and can be used to calculate the term $\mathrm{tr}\sum_{t=0}^{T} A_m^t A_m^{\top t}$ which appears in previous analyses (Sarkar and Rakhlin, 2019; Simchowitz et al., 2018).

Next, we state the main estimation error rate result for the the joint estimator in (6.6):

Theorem 6.2. *Under Assumption 6.6, the estimator in* (6.6) *returns* $\widehat{A}_m$ *for each system* $m \in [M]_+$, *such that with probability at least* $1-\delta$, *the following holds:*

$$\frac{1}{M}\sum_{m=1}^{M}\left\|\widehat{A}_m - A_m\right\|_F^2 \lesssim \frac{c^2\left(k\log\boldsymbol{\kappa}_\infty + \frac{d^2k}{M}\log\frac{\boldsymbol{\kappa}dT}{\delta}\right)}{\underline{\lambda}}. \tag{6.12}$$

Hence, by Lemma 6.1, the estimation error per system is

$$\frac{c^2 k\log\boldsymbol{\kappa}_\infty}{\lambda_{\min}(C)T} + \frac{c^2d^2k\log\frac{\boldsymbol{\kappa}dT}{\delta}}{M\lambda_{\min}(C)T}. \tag{6.13}$$

Equation (6.13) demonstrates the effects of pooling data across the systems. The first term $\frac{c^2k\log\boldsymbol{\kappa}_\infty}{\lambda_{\min}(C)T}$ on the RHS can be interpreted as the error in estimating the idiosyncratic components β_m for each system. The convergence rate is $O\left(\frac{k}{T}\right)$, as each β_m is a k-dimensional parameter and for each system, we have a trajectory of length T. More importantly, the second term $\frac{c^2d^2k\log\frac{\boldsymbol{\kappa}dT}{\delta}}{M\lambda_{\min}(C)T}$ indicates that the joint estimator in (6.6) effectively increases the sample size for the shared components $\{W_i\}_{i=1}^k$ by pooling the data of all systems. Therefore, the error decays at the rate $O(\frac{d^2k}{MT})$, indicating that the effective sample size for W^*B^* is MT.

In contrast, for LTI systems with general martingale noise, the individual estimation error rate is

$$\left\|\widehat{A}_m - A_m\right\|_F^2 \lesssim \frac{c^2d^2\log\frac{\alpha(A_m)T}{\delta}}{\lambda_{\min}(C)T}$$

(Faradonbeh et al., 2018a, 2020a; Sarkar and Rakhlin, 2019; Simchowitz et al., 2018). Thus, when the largest block-sizes l_m^* are not too large, the joint estimation error rate significantly improves when

$$k \ll d^2 \text{ and } k \ll M. \tag{6.14}$$

Note that if l_m^* is large, the rates are again similar for both joint and individual learning methods (as the error is $O(\log\alpha(A_m))$). Further, the conditions in (6.14) are as expected, as the ensuing discussion shows. First, when $k \approx d^2$, the idiosyncratic components in Assumption 6.6 do not provide any reduction in the effective dimension of the unknown transition matrices. That is, the absence of any shared structure in case $k \approx d^2$ prevents joint learning from being any different than individual learning. On the other hand, $k \approx M$ indicates that the systems $\{A_m\}_{m=1}^M$ are too

heterogeneous to allow any improvement under joint estimation.

Importantly, when the largest block-size l_m^* varies significantly across the M systems, a higher degree of shared structure is needed to improve the joint estimation error for all systems. Since $\boldsymbol{\kappa}$ and $\boldsymbol{\kappa}_\infty$ depend exponentially on l_m^* (Lemma 6.1) and l_m^* can be as large as d, it holds that $\log \boldsymbol{\kappa}_\infty = \log \boldsymbol{\kappa} = \Omega(d)$. Hence, we incur an additional dimension dependence in the error of the joint estimator. This leads to the following constraints on k for obtaining improved rates: $k \ll d$ and $kd \ll M$. Therefore, our analysis highlights the effects of the largest block-size in the Jordan form of the transition matrices of systems on joint estimation. This is an inherent difference between *independent* observations and sequentially *dependent data* generated by (6.4).

Moreover, an estimation error result can be obtained for the joint stochastic matrix regression setting

$$y_m(t) = A_m x_m(t) + \eta_m(t), \tag{6.15}$$

wherein the regressor $x_m(t)$ for task m is drawn from some distribution $\mathcal{D}_m$, and y_m is the response of task m. In this case, the sample covariance matrix Σ_m for each task is independent of A_m. Hence, the error for the joint estimator is not affected by the block-sizes in the Jordan decomposition of A_m. Therefore, in this setting, joint learning always leads to improved per-task error rates when $k \ll d^2$ and $k \ll M$.

Finally, it is worth noting that one does not need to solve the minimization problem in (6.6) to global optimality and can account for a moderate degree of optimization error. For instance, suppose that the optimization problem is solved up to an error of Δ from the global optimum. It can be shown that an additional term of magnitude $O\left(\Delta/\lambda_{\min}(C)\right)$ arises in the per-system estimation error in Theorem 6.2, due to this optimization error.

6.3.2 Impact of Misspecification on Estimation Error

In Theorem 6.2, we showed that the structure in Assumption 6.6 can be utilized for obtaining an improved estimation error by jointly learning the M systems.

Next, we consider what is the impact on the estimation error if the shared data generation mechanism in Assumption 6.6 is misspecified.

Formally, we consider the case where the dynamics of each system $m \in [M]_+$ follows a misspecified version of Assumption 6.6, as follows:

Assumption 6.7. *For each system $m \in [M]_+$, we have*

$$A_m = \left(\sum_{i=1}^{k} \beta_m^*[i] W_i^* \right) + D_m, \tag{6.16}$$

such that $\|D_m\|_F \leq \zeta_m$. Further, let $\bar{\zeta}^2 = \sum_{m=1}^{M} \zeta_m^2$.

Indeed, $D_m \in \mathbb{R}^{d \times d}$ is the deviation of system m from the shared structure of linear combination and $\bar{\zeta}^2$ captures the total misspecification. Under Assumption 6.7, we show the following result for the joint estimation error.

Theorem 6.3. *Under Assumption 6.7, the estimator in* (6.6) *returns $\widehat{A}_m$ for each system $m \in [M]_+$, such that with probability at least $1 - \delta$, we have:*

$$\frac{1}{M} \sum_{m=1}^{M} \left\| \widehat{A}_m - A_m \right\|_F^2 \lesssim \frac{c^2 \left(k \log \boldsymbol{\kappa}_\infty + \frac{d^2 k}{M} \log \frac{\boldsymbol{\kappa} dT}{\delta} \right)}{\underline{\lambda}} + \frac{(\boldsymbol{\kappa}_\infty + 1)\, \bar{\zeta}^2}{M}. \tag{6.17}$$

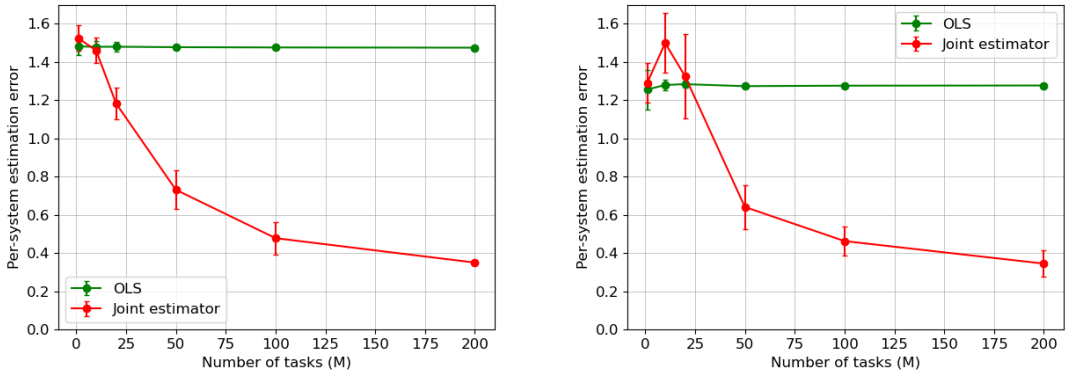
The proof of Theorem 6.3 can be found in Section 6.9.3. In (6.17), we observe that the addition of $\bar{\zeta}^2$ imposes an additional error of $(\boldsymbol{\kappa}_\infty + 1)\bar{\zeta}^2$ for jointly learning all M LTIDS. Hence, to obtain accurate estimates of the transition matrices, we need the total misspecification $\bar{\zeta}^2$ to be small. The discussion following Theorem 2 indicates that in order to have acccurate estimates, the number of the shared bases k must be small as well. Specifically, compared to individual learning, the joint estimation error improves despite model misspecification, if:

$$\frac{\boldsymbol{\kappa}_\infty \bar{\zeta}^2}{M} \ll \frac{d^2}{T}. \tag{6.18}$$

The condition in (6.18) shows that when when the total misspecification is proportional to the number of systems $\bar{\zeta}^* = \Omega(M)$, we pay a constant factor proportional to $\boldsymbol{\kappa}_\infty$ on the per-system estimation error. Note that in case that all systems are stable, according to Lemma 6.1, $\boldsymbol{\kappa}_\infty$ can be exponentially large in d (if $l_m^* = \Omega(d)$), but does not grow with T. On the other hand, when a transition matrix A_m has eigenvalues close to the unit circle, the factor $\boldsymbol{\kappa}_\infty$ can grow polynomially with T. Thus, misspecification nullifies the benefits of joint learning.

More generally, if for the total misspecification we have $\bar{\zeta}^2 = O(M^{1-a})$ for some $a > 0$, joint estimation improves over the individual estimators, as long as $\frac{\boldsymbol{\kappa}_\infty}{M^a} \ll \frac{d^2}{T}$. Hence, when all systems are stable, the joint estimation error rate improves when the number of systems satisfies $M \gg T^{1/a}$.

Further, the impact of $\bar{\zeta}^2$ is amplified when the transition matrices A_m have eigenvalues close to the unit circle. Indeed, letting l^* be the size of the largest block in the Jordan forms of the matrices A_m, the joint estimation error improves over the individual estimator when $d^2 M^a \gg T^{2l^*+2}$. Thus,



(a) System matrices A_m are stable

(b) System matrices A_m have a unit root

Figure 6.1: Per-system estimation error vs. number of systems M. OLS refers to the least squares estimator for learning linear dynamical systems.

when $l^* = \Omega(d)$, the number of systems needs to be as large as $d^2 T^{(2d+2)/a}$.

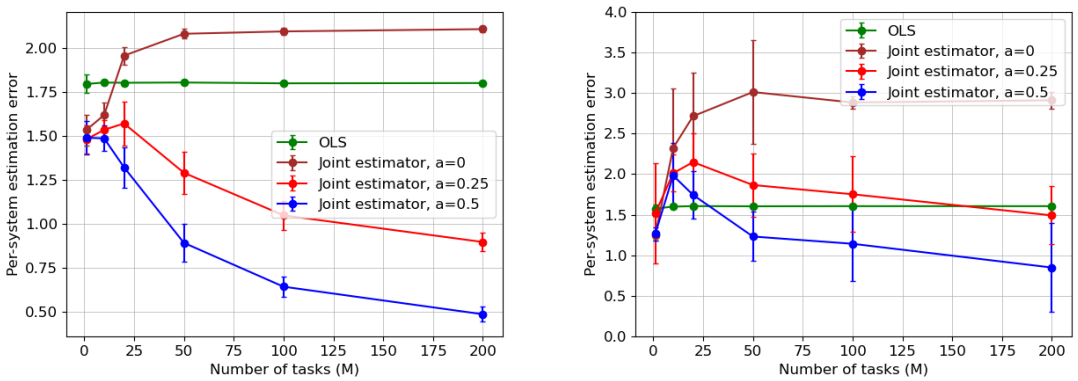
In contrast, the joint estimation error for a stochastic matrix regression problem in (6.15) incurs an additive factor of $O(1/M^a)$ and does not need to scale exponentially in d as in the unit root case for linear systems. Hence, *Theorem 6.3 further highlights the stark difference between joint estimation rates for independent observations and sequentially dependent data.*

6.3.3 Numerical Study

We complement our theoretical results with a set of numerical experiments which demonstrate the benefit of using the joint estimator for learning LTIDS. To that end, we compare the estimation error for the joint estimator in (6.6) against the ordinary least-squares estimates of the transition matrices for each system individually. For solving (6.6), we use a minibatch gradient descent based implementation in PyTorch (Paszke et al., 2019) and choose Adam (Kingma and Ba, 2015) as the optimization algorithm.

For generating the transition matrices, we consider settings with the number of bases $k = 10$, dimension $d = 25$, trajectory length $T = 200$, and the number of systems $M \in \{1, 10, 20, 50, 100, 200\}$. We simulate two cases: (i) the largest magnitude of the eigenvalues of all systems are in the range $[0.7, 0.9]$, and (ii) all systems have at least one eigenvalue of magnitude 1.

The matrices $\{W_i\}_{i=1}^{10}$ are generated randomly, such that each entry of W_i is sampled independently from the standard normal distribution $N(0, 1)$. Using these matrices, we generate M systems by randomly generating the idiosyncratic components β_m from a standard normal distribution. Each matrix is then rescaled to ensure that the magnitude of the largest eigenvalue lies in the range $[0.7, 0.9]$ or is equal to 1. For generating the data of state trajectories, the noise vectors are isotropic



(a) System matrices A_m are stable

(b) System matrices A_m have a unit root

Figure 6.2: Per-system estimation error vs. number of systems M for varying proportions of misspecified systems ($a \in \{0, 0.25, 0.5\}$) averaged across 20 runs.

Gaussian random vectors with variance 4.

We simulate the joint learning problem in both cases with and without model misspecification. For the latter, deviations from the shared structure are simulated by the components D_m, which are added with probability $1/M^a$ for $a \in \{0, 0.25, 0.5\}$. The matrices D_m are generated from a $N(0, 0.01)$ distribution.

In our simulations, for each value of M, we average the errors from 10 random replicates (20 for Figure 6.2) and plot the standard deviation as the error bar. Figure 6.1 depicts the estimation errors for both stable and unit-root transition matrices versus different values of M. It can be seen that the joint estimator exhibits the expected improvement against the individual ordinary least-squares one.

More interestingly, in Figure 6.2(a), we observe that for stable systems the joint estimator is worse than the least squares one, when a violation occurs in all systems (i.e., $a = 0$). Note that it is consistent with Theorem 6.3, since in this case the total misspecification $\bar{\zeta}^2$ *scales linearly* with M. However, if the proportion of systems which violates the structure in Assumption 6.6 decreases, the joint estimation error also improves as expected ($a = 0.25, 0.5$).

Figure 6.2(b) depicts the estimation error for the joint estimator under misspecification for systems which have an eigenvalue on the unit circle in the complex plane. Our theoretical results suggest that the number of systems needs to be much larger in this case to circumvent the cost of misspecification in joint learning. The Figure corroborates this result, wherein we observe that the joint estimation error is worse than the least squares error for $a = 0.25$, in contrast to the improvement in the stable case. Decreasing the total misspecification further to $a = 0.5$ starts showing improvement for such systems as well.

6.4 Certainty Equivalence: From System Identification to Regret Minimization

In this section, we build upon the recent result of Simchowitz and Foster (2020) to show a finite time regret bound using the multi task assumption in (6.1) with the joint estimator shown in Algorithm 6.1. Recall that for the regret results, we will again use the iid Gaussian noise assumption stated in Assumption 6.3. Our analysis will rely on this independence assumption in order to use the Hanson-Wright inequality to bound quadratic functions of the states and control inputs which arise in the regret decomposition (Section 6.8). Similar results can be obtained for the general noise case via the truncation based method used in Section 6.3. Further, since we only consider strictly stable systems, we will not resort to the Jordan forms of the closed loop matrices $A_m + B_m K$ to analyze the formal guarantees, as we did for joint estimation in Section 6.3. Note that, we only consider stable systems here, and for the final bounds, ignore the additional dimension dependencies which might arise due to the sizes of the Jordan forms.

Algorithm 6.1 MT-OLS algorithm for joint system identification

1: **Input:** Data $z_m(\tau_{j-1}), \ldots, z_m(\tau_j - 1), x_m(\tau_j)$ for each system $m \in [M]_+$.

2: **return** $\left[\widehat{A}_{m,j}, \widehat{B}_{m,j}\right] = \sum_{i=1}^{k} \hat{\beta}_m[i] \widehat{W}_i$ and $\Sigma_{m,j}$, where

$$\left\{\widehat{W}_i\right\}_{i=1}^{k}, \left\{\hat{\beta}_m\right\}_{m=1}^{M} := \underset{\mathbf{W}, \beta_m}{\operatorname{argmin}} \sum_{m=1}^{M} \sum_{t=\tau_{j-1}}^{\tau_j - 1} \left\| x_m(t+1) - \left(\sum_{i=1}^{k} \beta_m[i] W_i \right) z_m(t) \right\|_2^2 \tag{6.19}$$

6.4.1 Algorithm: A perturbed certainty equivalent controller

We now formally describe the main multi-task control algorithm, Algorithm 6.2. Later, we will show a regret guarantee for Algorithm 6.2 in Theorem 6.4. The algorithm is a multi-task variant of the certainty equivalence algorithm proposed by Simchowitz and Foster (2020). Similar to their proposed CE setup, Algorithm 6.2 also takes as input a stabilizing controller for each system $m \in [M]_+$. Note that, the controller $K_{m,0}$ can be arbitrarily sub-optimal for each system. The key idea behind the algorithm is the use of the perturbation bounds on the cost of CE controllers as shown by Simchowitz and Foster (2020) (which we state in Theorem 6.6). The algorithm proceeds in epochs $j = 1, 2 \ldots$ of doubling length: $\tau_j - \tau_{j-1} = \tau_{j-1}$, $\tau_1 = 1$. We divide the multi-task algorithm into two stages as follows:

1. **Individual burn-in phase** At the beginning of each epoch, for each system $m \in [M]$, the algorithm uses an ordinary least squares estimate (see Algorithm 2.1 in Section 2.3.2) to form

an estimate $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}\right)$ using the data collected in the previous epoch. In the next step (line 9), the algorithm checks if the estimates for all system dynamics matrices $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}\right)$ are sufficiently close to the (A_m, B_m), for the perturbation bounds in Theorem 6.6 to take effect. If the test fails, the algorithm continues to use the stabilizing controller for each system while adding exploratory noise at a constant scale. If the test is successful, Simchowitz and Foster (2020) showed that a controller computed using any pair $(A, B) \in \mathrm{Conf}^m_{\mathrm{safe}}$ (a confidence set obtained via Algorithm 6.3) stabilizes the system and has low regret. Thereafter, the algorithm uses a certainty equivalent controller based on the joint estimates for the system matrices.

2. **Certainty equivalent controller using joint estimation** When the test in the first phase succeeds for all systems, the algorithm computes the certainty equivalent controller $K_\infty(\widehat{A}_{m,j}, \widehat{B}_{m,j})$ where $(\widehat{A}_{m,j}, \widehat{B}_{m,j})$ are now estimated using the joint estimator[3] in (6.19). The certainty equivalent controller is used for the remainder of the epoch where we add exploratory noise to our control inputs, whose scale is carefully chosen in the following analysis.

In the algorithm and analysis, we use the following defined quantities:

$$C_{\mathrm{safe}}(A, B) \coloneqq 54 \left\|\!\left| P_\infty(A, B) \right|\!\right|^5, \quad C_{\mathrm{est}}(A, B) \coloneqq 142 \left\|\!\left| P_\infty(A, B) \right|\!\right|^8,$$
$$\Psi_{B,m} \coloneqq \max(1, \left\|\!\left| B_m \right|\!\right|), \quad \Psi_m = \max(1, \left\|\!\left| A_m \right|\!\right|, \left\|\!\left| B_m \right|\!\right|),$$

and the set used for projecting into the safe set:

$$\mathcal{B}_{\mathrm{op}}(\varepsilon, A_0, B_0) = \{(A, B) : \left\|\!\left| A_0 - A \right|\!\right| \vee \left\|\!\left| B_0 - B \right|\!\right| \leq \varepsilon\}$$

When the joint estimation procedure in Algorithm 6.1 is used in Algorithm 6.2, we can show the following regret bound:

Theorem 6.4 (Multi-task regret bound for Algorithm 6.2)**.** *If Algorithm 6.2 is invoked with stabilizing controllers $K_{m,0}$ along with the joint estimation procedure (Algorithm 6.1), then with probability at least $1 - M\delta$, the average regret incurred across the M LQR systems over T rounds is bounded*

[3] Recall that we use $z(t) = [x(t), u(t)]$.

Algorithm 6.2 Multi-task Certainty Equivalent Control for LQR

Input: Stabilizing controllers for each system $K_{m.0}$, confidence parameter δ.
Initialize safe $\leftarrow$ False.
For each $m \in [M]_+$, play $u_m(0) \sim N(0, I)$.
for $j = 2, 3, \ldots$ **do**
  Let $\tau_j \leftarrow 2^j$.
  if safe $=$ False **then**
    Set $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}, \Sigma_{m,j}\right) \leftarrow$ OLS(j) for each $m \in [M]_+$.
    $\mathrm{Conf}_{m,j} \leftarrow 6\lambda_{\min}(\Sigma_{m,j})^{-1}\left(d \log 5 + \log\left(4j^2 \det(3\Sigma_{m,j}/\delta)\right)\right)$.
    if $\Sigma_{m,j} \succeq I$ and $1/\mathrm{Conf}_{m,j} \geq 9C_{\mathrm{safe}}\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}\right)$ for all $m \in [M]_+$ **then**
      safe $\leftarrow$ True and $j_{\mathrm{safe}} = j$.
      $\mathcal{B}^m_{\mathrm{safe}}, \sigma_m^2 \leftarrow$ SAFEROUNDINIT $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}, \mathrm{Conf}_{m,j}, \delta\right)$.
    else
      for $t = \tau_j, \ldots, 2\tau_j - 1$ **do**
        For each system, play $u_m(t) = K_{m,0}x_m(t) + g_m(t)$, where $g_m(t) \sim N(0, I)$.
  else
    Set $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}, \Sigma_{m,j}\right) \leftarrow$ MT-OLS(j) for each $m \in [M]_+$.
    Let $(\overline{A}_m, \overline{B}_m)$ be the Euclidean projection of $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}\right)$ onto $\mathcal{B}^m_{\mathrm{safe}}$.
    $\widehat{K}_{m,j} \leftarrow K_\infty(\overline{A}_m, \overline{B}_m)$ for each $m \in [M]_+$.
    for $t = \tau_j, \ldots, 2\tau_j - 1$ **do**
      For each system, play $u_m(t) = \widehat{K}_{m,j}x_m(t) + \sigma_{m,j}g_m(t)$, where $g_m(t) \sim N(0, I)$ and $\sigma^2_{m,j} \coloneqq \min\left(1, \sigma_m^2 \tau_j^{-1/2}\right)$.

as follows:

$$
\begin{aligned}
&\mathrm{Regret}\left(\{\Theta_m\}_{m=1}^M, T\right) \\
&\lesssim \sqrt{T \max_m \|\!| P_m |\!\|^9 \, \Psi_{B,m}^2 \log \frac{\|\!| P_m |\!\|}{\delta}} \left(k \log \boldsymbol{\kappa} + \frac{d d_x k}{M} \log \frac{d\boldsymbol{\kappa}}{\delta}\right) \\
&\quad + \sqrt{T} \frac{1}{M} \sum_{m=1}^M \left(d_u \Psi_{B,m}^3 \|\!| P_m |\!\|^{11/2} + \sqrt{d \log \tfrac{1}{\delta}} \|\!| P_m |\!\|^4\right) \\
&\quad + \frac{1}{M} \sum_{m=1}^M d^2 \Psi_{B,m}^2 \mathcal{P}_{m,0} \log \frac{1}{\delta} \left(\max_{m \in [M]_+} (1 + \|\!| K_{m,0} |\!\|^2) \|\!| P_m |\!\|^{10} \log \frac{\Psi_{B,m}^2 \mathcal{J}_{m,0}}{\delta}\right), \qquad (6.20)
\end{aligned}
$$

ignoring the poly-logarithmic terms.[4]
In (6.20)*,* $\boldsymbol{\kappa} = \max_{m \in [M]_+} d \log\left(\left(1 + \frac{2\|\!| P_m |\!\|}{\sigma_m^2}\right)\left(d\Psi_{B,m}^2 \|\!| P_m |\!\|^4 \log \frac{1}{\delta}\right)\right)$.

[4]The terms $\|\!| P_m |\!\|$ and $\mathcal{P}_{m,0}$ hide the dependence on the Jordan forms of the closed loop matrices $L_{m,j}$ in Algorithm 6.2, and therefore, do not contradict the claims of such dependencies discussed in detail in Section 6.3.

Algorithm 6.3 SAFEROUNDINIT$\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}, \mathrm{Conf}, \delta\right)$

Input: Stabilizable pair $\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}, \mathrm{Conf}, \delta\right)$.

return $\mathcal{B}^m_{\mathrm{safe}} := \mathcal{B}_{\mathrm{op}}\left(\mathrm{Conf}; \widehat{A}_{m,j}, \widehat{B}_{m,j}\right)$ and

$\sigma_m^2 := \left\|\left\| P_\infty\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}\right)\right\|\right\|^{9/2} \max\left(1, \left\|\left\|\widehat{B}_{m,j}\right\|\right\|\right) \sqrt{\log \frac{\left\|\left\| P_\infty\left(\widehat{A}_{m,j}, \widehat{B}_{m,j}\right)\right\|\right\|}{\delta}}$.

We defer the proof of Theorem 6.4 to Section 6.8. The regret bound in Theorem 6.4 has three terms which can be interpreted as follows:

1. The first term signifies the main contribution of the joint estimator in Algorithm 6.2. This term is proportional to the average estimation error summed over all epochs across systems and has an improved dependence on the dimensionality. We discuss this in detail later.

2. The second term has two components: the first part can be thought of as the penalty paid for injecting random exploration noise which is proportional to σ_m^2 for each system. The second expression arises while bounding the random fluctuations in cost due to the system noise process $\eta_m(t)$ and $g_m(t)$. While the second term is unavoidable, the first term can be appropriately bounded by tuning the value of σ_m^2.

3. The third term is independent of T and accounts for the burn-in phase of the algorithm. Note that this term also depends on the cost of the initial controller $\mathcal{J}_{m,0}$ which can be arbitrarily large.

To compare the result in Theorem 6.4 against adaptive control of a single system, note that, Simchowitz and Foster (2020) showed the following finite time regret for a certainty equivalent based adaptive controller:

$$\sqrt{d_u^2 d_x T \cdot \Psi_B^2 \left\|\left\| P\right\|\right\|^{11} \log 1/\delta} + \max\left(1, \frac{d_u}{d_x}\right) d^2 \cdot \mathcal{P}_0 \Psi_B^6 \left\|\left\| P\right\|\right\|^1 1\left(1 + \left\|\left\| K_0\right\|\right\|^2\right) \log \frac{d \Psi_B \mathcal{P}_0}{\delta} \log^2 \frac{1}{\delta} \tag{6.21}$$

The first thing to compare between the two regret bounds is the factor arising due to the random fluctuations and the noise used for random exploration. Specifically, the random fluctuation due to the inherent noise of the system leads to the same term of order $\sqrt{T} \cdot \sqrt{d \left\|\left\| P\right\|\right\|^8 \log \frac{1}{\delta}}$ for both controllers. However, for the term arising due to persistent excitation, we can show a slight advantage as follows: for the single task case, the penalty scales as $\sqrt{T}\sqrt{d_u^2 d_x}$ whereas the multi-task controller incurs a penalty of $O\left(\sqrt{d_u^2 \cdot T}\right)$. Thus, when $d_x \gg 1$, this component of the regret bound improves significantly.

More importantly, in the term arising due to the estimation errors in the system matrices, we can again claim a multi-task improvement under the assumption that the multiplicities of the Jordan blocks of closed loop matrices $L_{m,j}$ are not too large for all m and j. Specifically, the two expressions for the single vs multi-task controllers are: $O\left(\sqrt{T \cdot d_u^2 d_x \left\|\!\left| P_m \right|\!\right\|^{11}}\right)$ vs $O\left(\left(k + \frac{k(d_u+d_x)d_x}{M}\right)\sqrt{T \cdot \left\|\!\left| P_m \right|\!\right\|^{9}}\right)$. Therefore, the joint rate improves significantly when:

$$k \ll \sqrt{d_u^2 d_x}$$
$$\frac{k}{M} \ll \sqrt{\frac{d_u^2}{d_x(d_u+d_x)^2}}$$

Thus, when $d_u < d_x$, we can reduce the two conditions to $k \ll d_u^{3/2}$ and $k/M \ll \sqrt{d_u/d_x^3}$ as a sufficient condition for the multi-task rate to improve. On the other hand, when $d_x \leq d_u$, having $k \ll d_x^{3/2}$ and $k/M \ll \sqrt{1/d_x}$ suffices to guarantee improvement. Therefore, for a large number of systems M and a sufficiently small size for the shared basis k, we can show an improvement in this component of the regret bound.

Note Despite the improvement in the two factors arising due to random exploration and certainty equivalence, we cannot show a significant improvement in the overall high-probability regret bound as the inherent stochasticity of the systems adds a penalty which is of the same order for both controllers. For an expected regret guarantee, we can circumvent the dependence on these natural random fluctuations by modifying the method in the two following ways: (1) account for the event when the states grow too large by aborting the adaptive controller and falling back to the stabilizing controller $K_{m,0}$ if the states grow too large, (2) ensure exploration by either injecting exploratory noise, or ensuring that the controller $\widehat{K}_{m,j}$ satisfies $\widehat{K}_{m,j}\widehat{K}_{m,j}^\top \succeq \mu_0 I$, such that exploration is ensured even when noise is not added, and (3) setting failure probability of all events to $O(T^{-2})$. For this case, we can set $\sigma_m = O(1)$ while ensuring that the total incurred expected regret is $O(\log T)$. For more details, the reader can refer to the work of Cassel et al. (2020).

6.5 Related Work

Multi-task learning The problem of joint learning in multiple systems has been studied in the statistical learning literature for a long time. Various joint learning algorithms have been proposed in the literature (Caruana, 1997) and their theoretical guarantees are established for supervised learning and online settings (Ando and Zhang, 2005; Maurer, 2006; Maurer et al., 2016; Alquier et al., 2017).

Joint learning with linear functions The closest to our work on multi-task adaptive control would be the recent results on provably efficient few-shot learning as established in Du et al. (2020); Jin et al. (2020). In these works, the authors consider a multi-task regression problem where the regression parameters can be written using a low-dimensional representation (of size k) and a shared projection matrix (of size $d \times k$). The analysis, thereof, shows that a given set of tasks can be learnt at a faster rate if a joint estimator is used, given that, the number of tasks $M \gg T$ where T is the sample size per task and $k \ll d$. Our analysis borrows key elements from Du et al. (2020) and then shows that the sequential setting has fundamental differences in the joint estimation properties compared to the iid univariate regression setting. Similarly, Hu et al. (2021); Lu et al. (2021) have recently studied a multi-task representation learning setting for linear contextual bandits and linear MDPs.

Related structural assumptions We note that the the shared basis assumption is similar to the linear CMDP setting (Example 3.2) and the model ensemble setting (Definition 4.1). For episodic LQR systems, Du et al. (2019c) considered an online contextual control setting with a linear structural assumption where the context is observed at the beginning of each episode. Using this observed context, the authors propose a simple modification of the algorithm by Abbasi-Yadkori and Szepesvári (2011) to show efficient regret bounds. However, a key difference between these and the structure in Assumption 6.1 is that both the combination coefficients and the shared parameters are unobserved and need to be learnt.

6.6 Discussion

Multi-task sequential decision making is a practically relevant problem and is important for expanding the application scenarios where reinforcement learning can be feasibly deployed. In this chapter, we have taken one step in this direction by giving the first (to our knowledge) finite time regret analysis for a multi-task LQR setting. However, as this is still in the nascent stages, below we outline a few missing pieces yet to be filled in the multi-task control problem:

Per-task estimation error rates In the first part of this chapter, we analyzed a joint estimator for system identification which builds upon the recent results of Du et al. (2020) and Tripuraneni et al. (2021). In our main estimation error result, we showed an improved rate for the average error across the given M tasks. However, it is desirable to have an improved estimation error rate result for each task individually as it can further improve the end-to-end online control scheme. For instance, in Algorithm 6.2 we use the ordinary least squares estimator in the first phase, so that the perturbation results in Theorem 6.6 can be invoked for each system individually. Establishing such per-task

guarantees will likely require additional and potentially strong assumptions (see assumptions in Tripuraneni et al. (2021) for instance) and we leave this for future work.

Adaptive multi-task stabilization Despite the recent flurry of recent works in adaptive optimal control for LQR, learning a stabilizing controller hasn't been studied much (see Faradonbeh et al. (2018b)). As such, most provably efficient algorithms use an assumption of access to a stabilizing controller in their algorithms. Addressing the same issue for stabilizable systems in the multi-task setting will require a careful analysis of the estimation error along with adaptive choices of control inputs in the stabilization phase. At this point, we are not certain if an improved rate without strong assumptions can be obtained and therefore further investigation is required.

Optimal dimension dependence Finally, the multi-task regret result shown in Theorem 6.4 improves over the single-task rates when the basis size k is small and the number of tasks M is large. The optimal rates for our setting has not been derived and we further don't have a lower bound to complete the picture. Both of these are interesting and challenging problems which are left to future work.

6.7 Summary

In this chapter, we studied the problem of joint learning and control in LTI dynamical systems with quadratic costs, under the assumption that their transition matrices can be expressed using a shared basis. Our finite time analyses for the joint estimator show that pooling data across systems can provably improve over individual estimators, even in presence of moderate misspecifications. Our results highlight the critical role of the spectral properties of the linear systems, along with the size of the basis, in the efficiency of joint estimation and formally illustrate a fundamental difference between joint estimation for independent observations and sequentially dependent data. Further, our regret result shows that the joint estimator can be used in a certainty equivalence based procedure to adaptively and concurrently control multiple LQR systems with better finite time regret guarantees.

Considering different structural assumptions and extensions to explosive systems, high-dimensional settings and non-linear dynamical systems are interesting avenues for future work and this work paves the road towards them. Further, there are many open questions which remain to be addressed for multi-task control and are left for future work.

6.8 Proof of Main Regret Bound

Preliminaries In this section, we will not use the Jordan normal forms of the matrices for our analysis. Instead, we will reference standard control theory results using the following discrete time Lyapunov operator:

Definition 6.1. *Let $Y \in \mathbb{R}^{d\times d}$ be a symmetric matrix and $X \in \mathbb{R}^{d\times d}$ have $\rho(X) < 1$. Then, we define the following quantity:*

$$\operatorname{dlyap}(X, Y) \coloneqq \sum_{i=0}^{\infty} \left(X^\top\right)^i Y X^i.$$

Also, define $P_\infty(K; A, B) \coloneqq \operatorname{dlyap}\left(A + BK, R_x + K^\top R_u K\right)$ for any stable closed loop matrix $A + BK$. With this definition, we can state the following bounds on the discrete time Lyapunov operator from Simchowitz and Foster (2020):

Lemma 6.5. *The following bounds hold:*

1. *If $Y \preceq Z$ and A is stable, then $\operatorname{dlyap}(A, X) \preceq \operatorname{dlyap}(A, Y)$. Similarly, for $Y \succeq 0$, we have $\operatorname{dlyap}(A, Y) \succeq Y$.*

2. *Suppose $R_x \succeq I$ and $A + BK$ is stable. Then:*

$$\pm\operatorname{dlyap}(A + BK, Y) \preceq \operatorname{dlyap}(A + BK, I) \left\|\!\left| Y \right|\!\right\| \preceq \left\|\!\left| Y \right|\!\right\| \cdot P_\infty(K; A, B).$$

3. *When $R_x \succeq I$, $\operatorname{dlyap}(A + BK, I) \preceq P_\infty(K; A, B)$ and $I \preceq \operatorname{dlyap}(A + BK_\infty(A, B), I) \preceq P_\infty(A, B)$.*

4. *If A_*, B_* is stabilizable and $A_* + B_* K$ is stable, then*

$$P_\infty(K; A_*, B_*) \succeq P_\infty(A_*, B_*) = P_\infty(K_*; A_*, B_*).$$

Further, $\mathcal{J}_K(A_*, B_*) = \operatorname{tr} P_\infty(K; A_*, B_*)$. Therefore, $\mathcal{J}_K(A_*, B_*) \geq \mathcal{J}^*(A_*, B_*) \geq d_x$.

Similarly, we use the following notation in the arguments below:

$$P_{m,j} = P_\infty\left(\widehat{K}_j; A_m, B_m\right), \quad \mathcal{P}_{m,0} \coloneqq \frac{\mathcal{J}_{m,0}}{d_x} \leq \left\|\!\left| P_\infty(K_{m,0}; A_m, B_m) \right|\!\right\|,$$
$$\mathcal{J}_{m,j} = \mathcal{J}_{\widehat{K}_{m,j}}(A_m, B_m), \quad \mathcal{J}_{m,0} = \mathcal{J}_{K_{m,0}}(A_m, B_m).$$

For the closed loop system matrices, we will use the notation, $L_m \coloneqq A_m + B_m K_m$, $L_{m,j} \coloneqq A_m + B_m \widehat{K}_{m,j}$ and $L_{m,0} \coloneqq A_m + B_m K_{m,0}$.

Define To begin, we first state the following result directly taken from Simchowitz and Foster (2020):

Theorem 6.6 (Corectness of Perturbations)**.** *On the event*

$$\mathcal{E}_{\text{safe}} := \left\{ \left\| \widehat{A}_{m,j_{\text{safe}}} - A_m \middle| \widehat{B}_{m,j_{\text{safe}}} - B_m \right\| \leq \text{Conf}_{m,j_{\text{safe}}}, \forall m \in [M]_+ \right\},$$

the following bounds hold for all $j \geq j_{\text{safe}}$ *and* $m \in [M]_+$*:*

1. *For the second phase, we have*

$$\begin{aligned} \mathcal{J}_{m,j} - \mathcal{J}_m &\leq C_{\text{est}}(A_m, B_m) \left(\left\| \widehat{A}_{m,j} - A_m \right\|_F^2 + \left\| \widehat{B}_{m,j} - B_m \right\|_F^2 \right) \\ &\lesssim \|\!| P_m |\!\|^8 \left(\left\| \widehat{A}_{m,j} - A_m \right\|_F^2 + \left\| \widehat{B}_{m,j} - B_m \right\|_F^2 \right) \end{aligned}$$

2. $\mathcal{J}_{m,j} \lesssim \mathcal{J}_m$ *and* $\|\!| P_{m,j} |\!\| \lesssim \|\!| P_m |\!\|$.
3. $\left\| \widehat{K}_{m,j} \right\|^2 \leq \frac{21}{20} \|\!| P_m |\!\|$.
4. $\|L_{m,j}\|_{\mathcal{H}_\infty} \lesssim \|L_m\|_{\mathcal{H}_\infty} \lesssim \|\!| P_m |\!\|^{3/2}$.
5. $L_{m,j}^\top \text{dlyap}\left(L_m, I\right) L_{m,j} \preceq \left(1 - \frac{1}{2} \|\!| \text{dlyap}\left(L_m\right) |\!\|^{-1}\right)$*, where* $0 \preceq \text{dlyap}\left(L_m\right) \preceq P_m$.
6. $\sigma_m^2 = \Theta\left(\|\!| P_m |\!\|^{9/2} \Psi_{B,m} \sqrt{\log \frac{\|\!| P_m |\!\|}{\delta}} \right)$.

6.8.1 Regret incurred in initial rounds

In Algorithm 6.2, for the first phase, we use the same steps based on the stabilizing controllers as used in Simchowitz and Foster (2020), for each system $m \in [M]_+$. Therefore, in the following result, we show that their analysis can be easily adapted to the multi-task case. The key difference here would be to account for the possibly different value of j_{safe} in the multi-task case compared to the individual controller results established in Simchowitz and Foster (2020).

Lemma 6.7. *The event* $\mathcal{E}_{\text{safe}}$ *holds with probability* $1 - \frac{M\delta}{2}$*, and the following event* $\mathcal{E}_{\text{reg,init}}$ *holds with probability* $1 - \frac{M\delta}{8}$*:*

$$\begin{aligned} &\sum_{t=0}^{\tau_{j_{\text{safe}}}-1} x_m(t)^\top R_x x_m(t) + u_m(t)^\top R_u u_t \\ &\lesssim d^2 \Psi_{B,m}^2 \mathcal{P}_{m,0} \log \frac{1}{\delta} \left(\max_{m \in [M]_+} \left(1 + \|\!| K_{m,0} |\!\|^2\right) \|\!| P_m |\!\|^{10} \log \frac{\Psi_{B,m}^2 \mathcal{J}_{m,0}}{\delta} \right) \end{aligned}$$

Proof. Since, the algorithm in the first phase is exactly the same as the certainty equivalent controller in Simchowitz and Foster (2020), we directly use their results in the proof while accounting for the discrepancy in $\tau_{j_{\text{safe}}}$. The proof of the results we use rely on the fact that in the complete first phase, the stabilizing controller $K_{m,0}$ is used. Thus, we can use the same results despite the difference between the eventual upper bound on j_{safe} in the multi-task and single task case.

The first part of the lemma, about $\mathcal{E}_{\text{safe}}$, follows directly from Lemma G.8 from Simchowitz and Foster (2020) with a union bound over the M systems.

For showing the regret upper bound, we start with the following result for each system which was shown in Simchowitz and Foster (2020) by using Hanson-Wright inequality in the regret decomposition:

Lemma 6.8 (Lemma G.9, Simchowitz and Foster (2020))**.** *For $\delta \leq 1/T$, the following holds with probability $1 - M\delta$ for all $m \in [M]_+$:*

$$\sum_{t=0}^{\tau_{j_{\text{safe}}}-1} x_m(t)^\top R_x x_m(t) + u_m(t)^\top R_u u_t \lesssim d\tau_{j_{\text{safe}}} \Psi_{B,m}^2 \mathcal{P}_{m,0} \log \frac{1}{\delta}$$

This above result can be proven similarly as Lemma G.9 in Simchowitz and Foster (2020) with an additional step for taking a union bound. Now, we state the upper bound on $\tau_{j_{\text{safe}}}$, as follows. To begin, we state the following result:

Lemma 6.9 (Lemma G.10, Simchowitz and Foster (2020))**.** *Suppose $\mathcal{E}_{\text{safe}}$ holds. Then, for all $j < j_{\text{safe}}$ for which $\Sigma_{m,j} \succeq I$, we must have that $\text{Conf}_{m,j} \gtrsim \varepsilon_{\text{safe}}$, where $\varepsilon_{\text{safe}} = \|\!| P_m |\!\|^{-10}$.*

The above result indicates that, for all $\varepsilon \in (0, 1)$

$$\text{if } \tau_j \geq \frac{d(1 + \|\!| K_{m,0} |\!\|^2)}{\varepsilon} \log \frac{\Psi_{B,m}^2 \mathcal{J}_{m,0}}{\delta}, \quad \text{then } \text{Conf}_{m,j} \leq \varepsilon, \Sigma_{m,j} \succeq I$$

with probability at least $1 - O(\delta)$. Thus, using the result, we know that $\tau_{j_{\text{safe}}} \lesssim \max_{m \in [M]_+} d(1 + \|\!| K_{m,0} |\!\|^2) \|\!| P_m |\!\|^{10} \log \frac{\Psi_{B,m}^2 \mathcal{J}_{m,0}}{\delta}$. Substituting this upper bound on $\tau_{j_{\text{safe}}}$ in Lemma 6.8 gives the desired result which now holds with probability $1 - M/8\delta$. □

6.8.2 Regret incurred in safe rounds

We again begin with a decomposition of the algorithm's regret which holds when conditioned on the event $\mathcal{E}_{\text{safe}}$:

Lemma 6.10 (Lemma 5.2, Simchowitz and Foster (2020))**.** *There is an event $\mathcal{E}_{\text{reg}}$ which holds with probability at least $1 - M\delta/6$ such that, on $\mathcal{E}_{\text{reg}} \cap \mathcal{E}_{\text{safe}}$, the following bound holds for each system*

$m \in [M]$:

$$\sum_{t=\tau_{j_{\text{safe}}}}^{T} \left(c(x_m(t), u_m(t)) - \mathcal{J}_m\right) \lesssim \sum_{j=j_{\text{safe}}}^{\log T} \tau_j \left(\mathcal{J}_{m,j} - \mathcal{J}_m\right) + \log T \max_{j \le \log T} \|x_m(\tau_j)\|_2^2 + \sqrt{T}\left(d_u \sigma_m^2 \Psi_{B,m}^2 \|\!| P_m |\!\| + \sqrt{d \log \tfrac{1}{\delta}} \|\!| P_m |\!\|^4\right) + \log^2 \frac{1}{\delta}\left(1 + \sqrt{d}\sigma_m^2 \Psi_{B,m}^2\right) \|\!| P_m |\!\|^4 . \tag{6.22}$$

In (6.22), the first term $\sum_{j=j_{\text{safe}}}^{\log T} \tau_j \left(\mathcal{J}_{m,j} - \mathcal{J}_m\right)$ quantifies the sub-optimality of the controllers $\widehat{K}_{m,j}$ selected in each epoch j. We tackle this term by using the perturbation bound from 1. in Theorem 6.6, in terms of the estimation error $\left\|\widehat{A}_{m,j} - A_m\right\|_F^2 + \left\|\widehat{B}_{m,j} - B_m\right\|_F^2$. The next term, $\log T \max_{j \le \log T} \|x_m(\tau_j)\|_2^2$, quantifies the effect of switching controllers at each epoch and appears in the analysis of most regret minimizing algorithms (Abbasi-Yadkori and Szepesvári, 2011; Dean et al., 2018; Simchowitz et al., 2018). The third term which is proportional to $\sqrt{T}$ appears due to the effect of the exploratory noise added to the certainty equivalent controller as well as the deviation from the expected cost due to the random noise processes. Lastly, the last term is of lower order $(\text{poly}(\log T))$.

For the analysis of the joint estimator (Algorithm 6.1) and for bounding the second term in (6.22), we will use the following lemma:

Lemma 6.11 (Lemma 5.3, Simchowitz and Foster (2020))**.** *There is an event $\mathcal{E}_{\text{bound}}$ which holds with probability at least $1 - M\delta/6$ such that, conditioned on $\mathcal{E}_{\text{safe}} \cap \mathcal{E}_{\text{bound}}$,*

$$\left\|x_{\tau_j}\right\| \le \sqrt{x_m(\tau_j)^\top \text{dlyap}\left(L_m, I\right) x_m(\tau_j)} \lesssim \sqrt{\Psi_{B,m} \mathcal{J}_{m,0} \log \frac{1}{\delta}} \|\!| P_m |\!\|^{3/2}, \quad \forall j \ge j_{\text{safe}}$$

Now, we will show the main result which bounds the estimation error across the M systems for the joint estimator in (6.19):

Theorem 6.12 (Corollary of Theorem 6.2)**.** *Under Assumption 6.1, and for epochs j such that*

$$\tau_j \gtrsim \max\left(\left(d \log\left(\left(1 + \frac{2\|\!| P_m |\!\|}{\sigma_m^2}\right)\left(d\Psi_{B,m}^2 \|\!| P_m |\!\|^4 \log \frac{1}{\delta}\right)\right)\right), \mathcal{J}_{m,0} \|\!| P_m |\!\|^3, \sigma_m^4 \Psi_{B,m}^4\right),$$

the estimator in (6.19) returns $(\widehat{A}_{m,j}, \widehat{B}_{m,j})$ for each system $m \in [M]$ and epoch j, such that with probability at least $1 - M\delta/6$, the following holds:

$$\frac{1}{M}\sum_{m=1}^{M}\left(\left\|\widehat{A}_{m,j} - A_m\right\|_F^2 + \left\|\widehat{B}_{m,j} - B_m\right\|_F^2\right) \lesssim \max_{m \in [M]_+} \frac{\left(\sigma_{m,j}^2 + 2\|\!| P_m |\!\|\right)\left(k \log \boldsymbol{\kappa} + \frac{dd_x k}{M} \log \frac{d\boldsymbol{\kappa}}{\delta}\right)}{\sigma_{m,j}^2 \tau_j}$$

where $\boldsymbol{\kappa} = \max_{m\in[M]_+} d\log\left(\left(1+\frac{2\|P_m\|}{\sigma_m^2}\right)\left(d\Psi_{B,m}^2\|P_m\|^4\log\frac{1}{\delta}\right)\right)$.

A complete and detailed proof is deferred to Section 6.9.4. Further, we will denote the lower bound on τ_j, as required for the joint estimation error to take effect as τ_{je}. Now, combining the results in Lemma 6.10 and Lemma 6.11, we get:

$$\begin{aligned}\frac{1}{M}\sum_{m=1}^{M}\sum_{t=\tau_{j_{\text{safe}}}}^{T}\left(c(x_m(t),u_m(t))-\mathcal{J}_m\right) \lesssim \frac{1}{M}\sum_{m=1}^{M}\left(\sum_{j=j_{\text{safe}}}^{\log T}\tau_j\left(\mathcal{J}_{m,j}-\mathcal{J}_m\right)\right)\\ +\log T\frac{1}{M}\sum_{m=1}^{M}\sqrt{\Psi_{B,m}\mathcal{J}_{m,0}\log\frac{1}{\delta}}\|P_m\|^{3/2}\\ +\sqrt{T}\frac{1}{M}\sum_{m=1}^{M}\left(d_u\sigma_m^2\Psi_{B,m}^2\|P_m\|+\sqrt{d\log\tfrac{1}{\delta}}\|P_m\|^4\right)\\ +\frac{1}{M}\sum_{m=1}^{M}\log^2\frac{1}{\delta}\left(1+\sqrt{d}\sigma_m^2\Psi_{B,m}^2\right)\|P_m\|^4.\end{aligned}$$

We first unfold the first term using Theorem 6.6 and Theorem 6.12 as follows:

$$\begin{aligned}&\frac{1}{M}\sum_{m=1}^{M}\left(\sum_{j=j_{\text{safe}}}^{\log T}\tau_j\left(\mathcal{J}_{m,j}-\mathcal{J}_m\right)\right)\\ &=\frac{1}{M}\sum_{m=1}^{M}\left(\sum_{j>\tau_{\text{je}}}\tau_j\left(\mathcal{J}_{m,j}-\mathcal{J}_m\right)\right)+\frac{1}{M}\sum_{m=1}^{M}\mathcal{J}_m\sum_{j\le c\tau_{\text{je}}}\tau_j\\ &\le\sum_{j>\tau_{\text{je}}}\max_{m\in[M]_+}\frac{\|P_m\|^8\left(\sigma_{m,j}^2+2\|P_m\|\right)\left(k\log\boldsymbol{\kappa}+\frac{dd_xk}{M}\log\frac{d\boldsymbol{\kappa}}{\delta}\right)}{\sigma_{m,j}^2\tau_j}\\ &\quad+\frac{1}{M}\sum_{m=1}^{M}\mathcal{J}_m\tau_{\text{je}}.\end{aligned}$$

Substituting $\sigma_{m,j}^2 \coloneqq \sigma_m^2\tau_j^{-1/2}$ gives:

$$\begin{aligned}&\frac{1}{M}\sum_{m=1}^{M}\left(\sum_{j=j_{\text{safe}}}^{\log T}\tau_j\left(\mathcal{J}_{m,j}-\mathcal{J}_m\right)\right)\\ &\lesssim\sum_{j>\tau_{\text{je}}}\max_{m\in[M]_+}\|P_m\|^9\left(\frac{k\log\boldsymbol{\kappa}}{\sigma_m^2\sqrt{\tau_j}}+\frac{dd_xk\log\frac{d\boldsymbol{\kappa}}{\delta}}{\sigma_m^2\sqrt{\tau_j}}\right)+\frac{1}{M}\sum_{m=1}^{M}\mathcal{J}_m\tau_{\text{je}}\\ &\le\sqrt{T}\max_{m\in[M]_+}\left(\frac{\|P_m\|^9}{\sigma_m^2}\left(k\log\boldsymbol{\kappa}+dd_xk\log\frac{d\boldsymbol{\kappa}}{\delta}\right)\right)+\frac{1}{M}\sum_{m=1}^{M}\mathcal{J}_m\tau_{\text{je}}\end{aligned}$$

6.8.3 Final regret bound for the multi-task certainty equivalent controller

In this section, we finally combine the bounds for the initial and the safe rounds as follows:

$$
\begin{aligned}
&\frac{1}{M}\sum_{m=1}^{M}\sum_{t=0}^{T}\left(c_m(x_m(t),u_m(t))-\mathcal{J}_m\right)\\
&=\frac{1}{M}\sum_{m=1}^{M}\sum_{t=0}^{\tau_{j_{\text{safe}}}-1}\left(c_m(x_m(t),u_m(t))-\mathcal{J}_m\right)+\frac{1}{M}\sum_{m=1}^{M}\sum_{t>\tau_{j_{\text{safe}}}}\left(c_m(x_m(t),u_m(t))-\mathcal{J}_m\right)\\
&\lesssim\frac{1}{M}\sum_{m=1}^{M}d^2\Psi_{B,m}^2\mathcal{P}_{m,0}\log\frac{1}{\delta}\left(\max_{m\in[M]_+}\left(1+\left\|K_{m,0}\right\|^2\right)\left\|P_m\right\|^{10}\log\frac{\Psi_{B,m}^2\mathcal{J}_{m,0}}{\delta}\right)\\
&\quad+\sqrt{T}\max_{m\in[M]_+}\left(\frac{\left\|P_m\right\|^9}{\sigma_m^2}\left(k\log\boldsymbol{\kappa}+dd_xk\log\frac{d\boldsymbol{\kappa}}{\delta}\right)\right)+\frac{1}{M}\sum_{m=1}^{M}\mathcal{J}_m\tau_{\text{je}}\\
&\quad+\log T\frac{1}{M}\sum_{m=1}^{M}\sqrt{\Psi_{B,m}\mathcal{J}_{m,0}\log\frac{1}{\delta}}\left\|P_m\right\|^{3/2}\\
&\quad+\sqrt{T}\frac{1}{M}\sum_{m=1}^{M}\left(d_u\sigma_m^2\Psi_{B,m}^2\left\|P_m\right\|+\sqrt{d\log\tfrac{1}{\delta}}\left\|P_m\right\|^4\right)\\
&\quad+\frac{1}{M}\sum_{m=1}^{M}\log^2\frac{1}{\delta}\left(1+\sqrt{d}\sigma_m^2\Psi_{B,m}^2\right)\left\|P_m\right\|^4.
\end{aligned}
$$

Collecting the $\sqrt{T}$ terms, we get:

$$
\begin{aligned}
\text{Regret}\left(\{\Theta_m\}_{m=1}^M,T\right)&\lesssim\sqrt{T}\max_{m\in[M]_+}\left(\frac{\left\|P_m\right\|^9}{\sigma_m^2}\left(k\log\boldsymbol{\kappa}+dd_xk\log\frac{d\boldsymbol{\kappa}}{\delta}\right)\right)\\
&\quad+\sqrt{T}\frac{1}{M}\sum_{m=1}^{M}\left(d_u\sigma_m^2\Psi_{B,m}^2\left\|P_m\right\|+\sqrt{d\log\tfrac{1}{\delta}}\left\|P_m\right\|^4\right)
\end{aligned}
$$

Substituting $\sigma_m^2\approx\left\|P_m\right\|^{9/2}\Psi_{B,m}\sqrt{\log\frac{\left\|P_m\right\|}{\delta}}$, we get:

$$
\begin{aligned}
\text{Regret}\left(\{\Theta_m\}_{m=1}^M,T\right)&\lesssim\sqrt{T\max_m\left\|P_m\right\|^9\Psi_{B,m}^2\log\frac{\left\|P_m\right\|}{\delta}}\left(k\log\boldsymbol{\kappa}+\frac{dd_xk}{M}\log\frac{d\boldsymbol{\kappa}}{\delta}\right)\\
&\quad+\sqrt{T}\frac{1}{M}\sum_{m=1}^{M}\left(d_u\Psi_{B,m}^3\left\|P_m\right\|^{11/2}+\sqrt{d\log\tfrac{1}{\delta}}\left\|P_m\right\|^4\right)
\end{aligned}
$$

The total failure probability can be bounded by accounting for events $\mathcal{E}_{\text{safe}}$, $\mathcal{E}_{\text{bounded}}$, $\mathcal{E}_{\text{reg}}$ and the estimation error bound in Theorem 6.12: $\frac{M\delta}{2}+\frac{M\delta}{6}+\frac{M\delta}{6}+\frac{M\delta}{6}=M\delta$.

6.9 Proof of Joint Learning Results

6.9.1 Preliminary inequalities and supporting lemmas

Here, we outline the some probabilistic inequalities and intermediate results which will be used for proving the main results of the paper.

Proposition 6.1 (Bounding the noise sequence)**.** *For $T = 0, 1, \ldots$, and $0 < \delta < 1$, let $\mathcal{E}_{\text{bdd}}$ be the event*

$$\mathcal{E}_{\text{bdd}}(\delta) := \left\{ \max_{1 \le t \le T, m \in [M]_+} \|\eta_m(t)\|_\infty \le \sqrt{2\sigma^2 \log \frac{2dMT}{\delta}} \right\}. \tag{6.23}$$

Then, we have $\mathbb{P}[\mathcal{E}_{\text{bdd}}] \ge 1 - \delta$. For simplicity, we denote the aboove upper-bound by $b_T(\delta)$.

Proof. Let e_i be the i-th member of the standard basis of $\mathbb{R}^d$. Using the sub-Gaussianity of the random vector $\eta_m(t)$ given the sigma-field $\mathcal{F}_{t-1}$, we have

$$\mathbb{P}\left[|\langle e_i, \eta_m(t) \rangle| > \sqrt{2\sigma^2 \log \frac{2}{\delta'}} \right] \le \delta'.$$

Therefore, taking a union bound over all basis vectors $i = 1, \cdots, d$, all systems $m \in [M]_+$, and all time steps $t = 1, \cdots, T$ that $\eta_m(t) > \sqrt{2\sigma^2 \log \frac{2}{\delta'}}$, we get the desired result by letting $\delta' = \delta(dMT)^{-1}$. □

Proposition 6.2 (Noise covariance concentration)**.** *For $T = 0, 1, \ldots$ and $0 < \delta < 1$, let $\mathcal{E}_\eta$ be the event*

$$\mathcal{E}_\eta(\delta) := \left\{ \frac{3\lambda_{\min}(C)}{4} I \preceq \frac{1}{T} \sum_{t=1}^{T} \eta_m(t) \eta_m(t)^\top \preceq \frac{5\lambda_{\max}(C)}{4} I \right\}.$$

Then, we have $\mathbb{P}[\mathcal{E}_{\text{bdd}}(\delta) \cap \mathcal{E}_\eta(\delta)] \ge 1 - 2\delta$.

Proof. Here, we will bound the largest eigenvalue of the deviation matrix $\sum_{t=1}^{T} \eta_m(t)\eta_m(t)^\top - TC$. For the spectral norm of this matrix, using Lemma 5.4 from Vershynin (2018), we have:

$$\left\| \sum_{t=1}^{T} \eta_m(t)\eta_m(t)^\top - TC \right\|_2 \le \frac{1}{1 - 2\tau} \sup_{v \in \mathcal{N}_\tau} \left| v^\top \left(\sum_{t=1}^{T} \eta_m(t)\eta_m(t)^\top - TC \right) v \right|,$$

where $\mathcal{N}_\tau$ is a τ-cover of the unit sphere $\mathcal{S}^{d-1}$. Now, it holds that $|\mathcal{N}_\tau| \leq (1+2/\tau)^d$. Thus, we get:

$$\mathbb{P}\left[\left\|\sum_{t=1}^{T}\eta_m(t)\eta_m(t)^\top - TC\right\|_2 \geq \epsilon\right] \leq \mathbb{P}\left[\sup_{v\in\mathcal{N}_\tau}\left|v^\top\left(\sum_{t=1}^{T}\eta_m(t)\eta_m(t)^\top - TC\right)v\right| \geq (1-2\tau)\epsilon\right]$$

Using some martingale concentration arguments, we first bound the probability on the RHS for a fixed vector $v \in \mathcal{N}_\tau$. Then, taking a union bound over all $v \in \mathcal{N}_\tau$ will lead to the final result.

For a given t, since $\eta_m(t)^\top v$ is conditionally sub-Gaussian with parameter σ, the quantity $v^\top\eta_m(t)\eta_m(t)^\top v - v^\top Cv$ is a conditionally sub-exponential martingale difference. Using Theorem 2.19 of Wainwright (2019), for small values of ϵ, we have

$$\mathbb{P}\left[\left|v^\top\left(\sum_{t=1}^{T}\eta_m(t)\eta_m(t)^\top - TC\right)v\right| \geq (1-2\tau)\epsilon\right] \leq 2\exp\left(-\frac{c_\eta(1-2\tau)^2\epsilon^2}{T\sigma^2}\right),$$

where c_η is some universal constant. Taking a union bound, setting total failure probability to δ, and letting $\tau = 1/4$, we obtain that with probability at least $1-\delta$, it holds that

$$\lambda_{\max}\left(\sum_{t=1}^{T}\eta_m(t)\eta_m(t)^\top - TC\right) \leq c_\eta\sigma\sqrt{T\log\left(\frac{2\cdot 9^d}{\delta}\right)}.$$

According to Weyl's inequality, for $T \geq T_\eta(\delta) \coloneqq \frac{c_\eta d\sigma^2}{\lambda_{\min}(C)^2}\log 18/\delta$, we have:

$$\frac{3\lambda_{\min}(C)}{4}I \preceq \frac{1}{T}\sum_{t=1}^{T}\eta_m(t)\eta_m(t)^\top \preceq \frac{5\lambda_{\max}(C)}{4}I.$$

□

Recall, that in the main text, we define $Z \in \mathbb{R}^{dT\times M}$ as the pooled noise matrix as follows:

$$Z = [\tilde{\eta}_1(T)|\tilde{\eta}_2(T)\cdots|\tilde{\eta}_M(T)], \tag{6.24}$$

with each column vector $\eta_m(T) \in \mathbb{R}^{dT}$ as the concatenated noise vector $(\eta_m(1), \eta_m(2), \ldots, \eta_m(T))$ for the m-th system.

Proposition 6.3 (Bounding total magnitude of noise)**.** *For the joint noise matrix $Z \in \mathbb{R}^{dT\times M}$ defined in* (6.24)*, with probability at least $1-\delta$, we have:*

$$\|Z\|_F^2 \leq MT\,\mathrm{tr}\,C + \log\frac{2}{\delta}.$$

We denote the above event by $\mathcal{E}_Z(\delta)$.

Proof. For each system m, we know that $\mathbb{E}[\eta_m(t)[i]^2|\mathcal{F}_{t-1}] = C_{ii}^2$. Similar to the previous proof, we know that $\eta_m(t)[i]^2$ follows a conditionally sub-exponential distribution given $\mathcal{F}_{t-1}$. Using the sub-exponential bound for martingale difference sequences, for large enough T we get:

$$\|Z\|_F^2 = \sum_{m=1}^{M}\sum_{t=1}^{T}\|\eta_m(t)\|^2 \leq MT\,\mathrm{tr}\,C + \log\frac{2}{\delta},$$

with probability at least $1-\delta$. □

The following self-normalized martingale bound shows a result similar to Lemma A.4 for vector valued noise processes.

Proposition 6.4. *For the system in* (6.4)*, for any* $0 < \delta < 1$ *and system* $m \in [M]_+$*, with probability at least* $1-\delta$*, we have:*

$$\left\|\bar{V}_m^{-1/2}(T-1)\sum_{t=0}^{T-1}X_m(t)\eta_m(t+1)^\top\right\|_2 \leq \sigma\sqrt{8d\log\left(\frac{5\det\left(\bar{V}_m(T-1)\right)^{1/2d}\det\left(V\right)^{-1/2d}}{\delta^{1/d}}\right)},$$

where $\bar{V}_m(s) = \sum_{t=0}^{s}X_m(t)X_m(t)^\top + V$ *and* V *is a deterministic positive definite matrix.*

Proof. We first state the following discretization-based concentration bound shown in Vershynin (2018) for random matrices:

Proposition 6.5. *Let* M *be a random matrix. For any* $\epsilon < 1$*, let* $\mathcal{N}_\epsilon$ *be an* ϵ*-net of* $\mathcal{S}^{d-1}$ *such that for any* $w \in \mathcal{S}^{d-1}$*, there exists* $\bar{w} \in \mathcal{N}_\epsilon$ *with* $\|w-\bar{w}\| \leq \epsilon$*. Then for any* $\epsilon < 1$*, we have:*

$$\mathbb{P}\left[|||M|||_2 > z\right] \leq \mathbb{P}\left[\max_{\bar{w}\in\mathcal{N}_\epsilon}\|M\bar{w}\| > (1-\epsilon)z\right]$$

For the partial sum $S_m(t) = \sum_{s=0}^{t}X_m(s)\eta_m(s+1)^\top$, using Proposition 6.5 with $\epsilon = 1/2$, we get:

$$\mathbb{P}\left[\left|\left|\left|\bar{V}_m^{-1/2}(T-1)S_m(T-1)\right|\right|\right|_2 \geq y\right] \leq \sum_{\bar{w}\in\mathcal{N}_\epsilon}\mathbb{P}\left[\left\|\bar{V}_m^{-1/2}(T-1)S_m(T-1)\bar{w}\right\|^2 \geq \frac{y^2}{4}\right]$$

where $\bar{w}$ is a fixed unit norm vector in $\mathcal{N}_\epsilon$. We can now apply Lemma A.4 with the σ sub-Gaussian noise sequence $\eta_t^\top w$ to get the final high probability bound. □

Lemma 6.13 (Covering number for low-rank matrices, (Du et al., 2020))**.** *Let* $O^{d\times d'}$ *be the set of matrices with orthonormal columns* ($d > d'$)*. Then there exists a subset* $\mathcal{N}_\epsilon \subset O^{d\times d'}$ *that forms an* ϵ*-net of* $O^{d\times d'}$ *in Frobenius norm such that* $|\mathcal{N}_\epsilon| \leq (\frac{6\sqrt{d'}}{\epsilon})^{dd'}$*, i.e., for every* $V \in O^{d\times d'}$*, there exists* $V' \in \mathcal{N}_\epsilon$ *and* $\|V-V'\|_F \leq \epsilon$*.*

6.9.2 Proof of bounds on covariance matrix

We show high probability bounds for each LTIDS' covariance matrix $\Sigma_m = \Sigma_m(T) = \sum_{t=0}^{T} X_m(t) X_m(t)^\top$.

6.9.2.1 An upper bound on LTIDS covariance matrix

In order to prove an upper bound on each system covariance matrix, we can use the technique proposed for general LTI systems by Faradonbeh et al. (2018a). Recall, that for any system A_m, we use l_m^* to denote the size of the largest Jordan block in the Jordan form of A_m. Using this, we first define the following quantity:

$$\alpha(A_m) \coloneqq \begin{cases} \left\|\!\left| P_m^{-1} \right|\!\right\|_{\infty\to 2} \left\|\!\left| P_m \right|\!\right\|_{\infty} e^{1/|\lambda_{m,1}|} \left[\frac{l_m^*-1}{-\log|\lambda_{m,1}|} + \frac{(l_m^*-1)!}{(-\log|\lambda_{m,1}|)^{l_{m,1}^*}} \right], & \text{if } |\lambda_1(A_m)| < 1 \\ \left\|\!\left| P_m^{-1} \right|\!\right\|_{\infty\to 2} \left\|\!\left| P_m \right|\!\right\|_{\infty} e^{\rho+1}, & \text{if } |\lambda_1(A_m)| \le 1 + \frac{\rho}{T}. \end{cases} \tag{6.25}$$

Using this definition, the first step is to bound the sizes of all state vectors under the event $\mathcal{E}_{\text{bdd}}(\delta)$ in Proposition 6.1.

Proposition 6.6 (Bounding $\|x_m(t)\|$)**.** *For all $t \in [T], m \in [M]_+$, under the event $\mathcal{E}_{\text{bdd}}(\delta)$, with probability at least $1-\delta$ we have:*

$$\|x_m(t)\| \le \begin{cases} \alpha(A_m)\left(b_T(\delta) + \|x_m(0)\|_\infty\right), & \text{if } |\lambda_{m,1}| < 1, \\ \alpha(A_m)\left(b_T(\delta) + \|x_m(0)\|_\infty\right) t^{l_m^*}, & \text{if } |\lambda_{m,1}| \le 1 + \frac{\rho}{T} \end{cases}. \tag{6.26}$$

Proof. As stated before, each transition matrix A_m admits a Jordan normal form as follows: $A_m = P_m^{-1} \Lambda_m P_m$, where Λ_m is a block-diagonal matrix $\Lambda_m = \operatorname{diag}(\Lambda_{m,q}, \ldots, \Lambda_{m,q})$. Each Jordan block $\Lambda_{m,i}$ is of size $l_{m,i}$. To begin, note that for each system, each state vector satisfies:

$$\begin{aligned} x_m(t) &= \sum_{s=1}^{t} A_m^{t-s} \eta_m(s) + A_m^t x_m(0) \\ &= \sum_{s=1}^{t} P_m^{-1} \Lambda_m^{t-s} P_m \eta_m(s) + P_m^{-1} \Lambda_m^t P x_m(0). \end{aligned}$$

Now, letting $b_T(\delta)$ be the same as in Proposition 6.1, we can bound the 2-norm of the state vector as

follows:

$$\begin{aligned}\|x_m(t)\| &\leq \left\|\left\|P_m^{-1}\right\|\right\|_{\infty\to 2} \left\|\left\|\sum_{s=1}^{t}\Lambda_m^{t-s}\right\|\right\|_{\infty} \|P_m\|_\infty\, b_T(\lambda) + \left\|\left\|P_m^{-1}\right\|\right\|_{\infty\to 2} \left\|\left\|\Lambda_m^{t}\right\|\right\|_{\infty} \|P_m\|_\infty \|x_m(0)\|_\infty \\ &\leq \left\|\left\|P_m^{-1}\right\|\right\|_{\infty\to 2} \left(\sum_{s=0}^{t}\left\|\left\|\Lambda_m^{t-s}\right\|\right\|_{\infty}\right) \|P_m\|_\infty \left(b_T(\delta) + \|x_m(0)\|_\infty\right).\end{aligned}$$

For any matrix, the ℓ_∞ norm is equal to the maximum row sum. Since the powers of a Jordan matrix will follow the same block structure as the original one, we can bound the operator norm $\left\|\left\|A_m^{t-s}\right\|\right\|_\infty$ by the norm of each block. For any Jordan matrix of size l and eigenvalue λ, we have:

$$\Lambda^s = \begin{bmatrix} \lambda^s & \binom{s}{1}\lambda^{s-1} & \dots & \binom{s}{l-1}\lambda^{s-l+1} \\ 0 & \lambda^s & \dots & \binom{s}{l-2}\lambda^{s-l+2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda^s \end{bmatrix},$$

Thus, the maximum row sum for the s-th power of a Jordan block is: $\sum_{j=0}^{l-1}\binom{s}{j}\lambda^{s-j}$. Using this, we will bound the size of each state vector for the case when

(I) the spectral radius of A_m satisfies $\lambda_1(A_m) < 1$ and,

(II) when $\lambda_1(A_m) \leq 1 + \frac{\rho}{T}$ for some constant $\rho > 0$.

Case I When the Jordan block for a system matrix has eigenvalues strictly less than 1, we can state the following bound:

$$\begin{aligned}\sum_{s=0}^{t}\left\|\left\|\Lambda_m^{t-s}\right\|\right\|_\infty &\leq \max_{i\in[q_m]}\sum_{s=0}^{t}\sum_{j=0}^{l_{m,i}-1}\binom{s}{j}\lambda_{m,i}^{s-j} \leq \sum_{s=0}^{t}\sum_{j=0}^{l_m^*-1}\binom{s}{j}\lambda_{m,1}^{s-j} \\ &\leq \sum_{s=0}^{t}\lambda_{m,1}^s\sum_{j=0}^{l_m^*-1}\frac{s^j}{j!}\lambda_{m,1}^{-j} \leq \sum_{s=0}^{t}\lambda_{m,1}^s s^{l_m^*-1}\sum_{j=0}^{l_m^*-1}\frac{\lambda_{m,1}^{-j}}{j!} \\ &\leq e^{1/|\lambda_{m,1}|}\sum_{s=0}^{t}\lambda_{m,1}^s s^{l_m^*-1} \leq e^{1/|\lambda_{m,1}|}\sum_{s=0}^{\infty}\lambda_{m,1}^s s^{l_m^*-1} \\ &\lesssim e^{1/|\lambda_{m,1}|}\left[\frac{l_m^*-1}{-\log|\lambda_{m,1}|} + \frac{(l_m^*-1)!}{(-\log|\lambda_{m,1}|)^{l_m^*}}\right].\end{aligned}$$

Thus, for this case, the magnitude of each state vector can be upper bounded as $\|x_m(t)\| \leq \alpha(A_m)\left(b_T(\delta) + \|x_m(0)\|_\infty\right)$.

When the matrix A_m is diagonalizable, each Jordan block is of size 1, which leads to the

upper-bound $\sum_{s=0}^{t} \left\|\Lambda_m^{t-s}\right\|_\infty \leq \frac{1}{1-\lambda_1}$, for all $t \geq 0$. Therefore for diagonalizable A_m, we can let $\alpha(A_m) = \frac{\left\|\left|P_m^{-1}\right|\right\|_{\infty\to 2}\|P_m\|_\infty}{1-\lambda_1}$.

Case II When $|\lambda_{m,1}| \leq 1 + \frac{\rho}{T}$, we get $|\lambda_{m,1}|^t \leq e^\rho$, for all $t \leq T$. Therefore, with l_m^* as the largest multiplicity of the (near)-unit root, we have:

$$\begin{aligned}
\sum_{s=0}^{t} \left\|\left|\Lambda_m^{t-s}\right|\right\|_\infty &\leq \sum_{s=0}^{t}\sum_{j=0}^{l_m^*-1} \binom{s}{j} \lambda_{m,1}^{s-j} \leq e^\rho \sum_{s=0}^{t}\sum_{j=0}^{l_m^*-1} \binom{s}{j} \\
&\leq e^\rho \sum_{s=0}^{t}\sum_{j=0}^{l_m^*-1} s^j/j! \leq e^\rho \sum_{s=0}^{t} s^{l_m^*-1} \sum_{j=0}^{l_m^*-1} 1/j! \\
&\leq e^{\rho+1} \sum_{s=0}^{t} s^{l_m^*-1} \\
&\lesssim e^{\rho+1} t^{l_m^*}.
\end{aligned}$$

Therefore, the magnitude of each state vector grows polynomially with t and further depends on the multiplicity of the unit root. When the matrix A_m is diagonalizable, the Jordan block for the unit root is of size 1 which bounds the term as $\sum_{s=0}^{t} \left\|\Lambda_m^{t-s}\right\|_\infty \leq e^\rho t$.

Therefore, for system matrices with unit roots, the bound on each state vector is $\|x_m(t)\| \leq \alpha(A_m)\left(b_T(\delta) + \|x_m(0)\|_\infty\right) t^{l_m^*}$. □

Using the high probability upper bound on the size of each state vector, we can upper bound the covariance matrix for each system as follows:

Lemma 6.14 (Upper bound on Σ_m)**.** *For all $m \in [M]_+$, under the event $\mathcal{E}_{\mathrm{bdd}}(\delta)$, with probability at least $1-\delta$ and $m \in [M]_+$, the sample covariance matrix Σ_m of system m can be upper bounded as follows:*

(I) When all eigenvalues of the matrix A_m are strictly less than 1 in magnitude ($|\lambda_{m,i}| < 1$), we have

$$\lambda_{\max}(\Sigma_m) \leq \alpha(A_m)^2 \left(b_T(\delta) + \|x_m(0)\|_\infty\right)^2 T.$$

(II) When some eigenvalues of the matrix A_m are close to 1, i.e. $|\lambda_1(A_m)| \leq 1 + \frac{\rho}{T}$, we have:

$$\lambda_{\max}(\Sigma_m) \leq \alpha(A_m)^2 \left(b_T(\delta) + \|x_m(0)\|_\infty\right)^2 T^{2l_{m,1}+1}.$$

Proof. First note that we have:

$$\lambda_{\max}(\Sigma_m) = \|\Sigma_m\|_2 = \left\| \sum_{t=0}^{T} x_m(t) x_m(t)^\top \right\|_2 \leq \sum_{t=0}^{T} \|x_m(t)\|_2^2.$$

Therefore, when all eigenvalues of A_m are strictly less than 1, we have:

$$\lambda_{\max}(\Sigma_m) \leq \sum_{t=0}^{T} \|x_m(t)\|_2^2 \leq T\alpha(A_m)^2 \left(b_T(\delta) + \|x_m(0)\|_\infty\right)^2.$$

For the case when $\lambda_1(A_m) \leq 1 + \frac{\rho}{T}$, we get:

$$\lambda_{\max}(\Sigma_m) \leq \sum_{t=0}^{T} \|x_m(t)\|_2^2 \leq \alpha(\Lambda_m)^2 \sum_{t=0}^{T} t^{2l_{m,1}} \leq \alpha(A_m)^2 \left(b_T(\delta) + \|x_m(0)\|_\infty\right)^2 T^{2l_{m,1}+1}.$$

□

6.9.2.2 Lower bound for covariance matrix of each LTIDS

A lower bound result for the idiosyncratic covariance matrices can be derived using the inequalities in Section 6.9.1 and the proof outline used in Proposition 10.1 in the work of Sarkar and Rakhlin (2019). We provide a detailed proof below.

Lemma 6.15 (Covariance lower bound.)**.** *For all $m \in [M]_+$ and $\bar{b}_m := b_T(\delta) + \|x_m(0)\|_\infty$, if the sample size per system is large enough such that:*

$$T \geq \begin{cases} \frac{d\sigma^2}{\lambda_{\min}(C)^2} \max\left(c_\eta \log\frac{18}{\delta}, 16\left(\log\left(\alpha(A_m)^2\bar{b}_m^2 + 1\right) + 2\log\frac{5}{\delta}\right)\right), & \text{if } \lambda_{m,1} < 1 \\ \frac{d\sigma^2}{\lambda_{\min}(C)^2} \max\left(c_\eta \log\frac{18}{\delta}, 16\left(\log\left(\alpha(A_m)^2\bar{b}_m^2 T^{2l_m^*} + 1\right) + 2\log\frac{5}{\delta}\right)\right), & \text{if } \lambda_{m,1} \leq 1 + \frac{\rho}{T}, \end{cases}$$

then with probability at least $1 - 3\delta$, the sample covariance matrix Σ_m for system m can be bounded from below as follows: $\Sigma_m(T) \succeq \frac{T\lambda_{\min}(C)}{4} I$.

Proof. We bound the covariance matrix under the events $\mathcal{E}_{\text{bdd}}(\delta)$, $\mathcal{E}_\eta(\delta)$, and when the event in Proposition 6.4 holds. As we consider a bound for all systems, we drop the system subscript m here. Under $\mathcal{E}_{\text{bdd}}(\delta)$, by Proposition 6.2, with probability at least $1 - \delta$, we have:

$$\begin{aligned} \Sigma(T) &\succeq A\Sigma(T-1)A^\top + \sum_{t=0}^{T-1} \left(Ax(t)\eta(t+1)^\top + \eta(t+1)x(t)^\top A^\top\right) + \sum_{t=1}^{T} \eta(t)\eta(t)^\top \\ &\succeq A\Sigma(T-1)A^\top + \sum_{t=0}^{T-1} \left(Ax(t)\eta(t+1)^\top + \eta(t+1)x(t)^\top A^\top\right) + \frac{3\lambda_{\min}(C)T}{4}. \end{aligned}$$

Thus, for any vector $u \in \mathcal{S}^{d-1}$, we have

$$u^\top \Sigma(T) u \succeq u^\top A\Sigma(T-1)A^\top u + \sum_{t=0}^{T-1} u^\top \left(Ax(t)\eta(t+1)^\top + \eta(t+1)x(t)^\top A^\top\right) u + \frac{3\lambda_d(C)T}{4}.$$

Now, in Proposition 6.4, with $V = T \cdot I$, we can show the same result for martingale term $\sum_{t=0}^{T-1} A_m X_m(t)\eta_m(t+1)^\top$ and $\bar{V}_m(s) \coloneqq \sum_{t=0}^{s} A_m X_m(t) X_m(t)^\top A_m^\top + V$. Hence, with probability at least $1-\delta$, we have:

$$\begin{aligned}
&\left\| \sum_{t=0}^{T-1} Ax(t)\eta(t+1)^\top u \right\| \\
&\leq \sqrt{u^\top A\Sigma(T-1)A^\top u + T} \sqrt{8d\sigma^2 \log\left(\frac{5 \det\left(\bar{V}_m(T-1)\right)^{1/2d} \det\left(TI\right)^{-1/2d}}{\delta^{1/d}} \right)}.
\end{aligned}$$

Thus, we get:

$$\begin{aligned}
u^\top \Sigma(T) u \succeq\ & u^\top A\Sigma(T-1)A^\top u + \frac{3\lambda_{\min}(C)T}{4} \\
& - \sqrt{u^\top A\Sigma(T-1)A^\top u + T} \sqrt{16d\sigma^2 \log\left(\frac{\lambda_{\max}(\bar{V}(T-1))}{T} \right) + 32d\sigma^2 \log\frac{5}{\delta}}.
\end{aligned}$$

Hence, we have:

$$\begin{aligned}
u^\top \frac{\Sigma(T)}{T} u &\succeq u^\top \frac{A\Sigma(T-1)A^\top}{T} u - \sqrt{u^\top \frac{A\Sigma(T-1)A^\top}{T} u + 1} \frac{\lambda_{\min}(C)}{2} + \frac{3\lambda_{\min}(C)}{4} \\
&\succeq \frac{\lambda_{\min}(C)}{4},
\end{aligned}$$

whenever

$$\begin{aligned}
T &\geq \frac{16d\sigma^2}{\lambda_{\min}(C)^2} \left(\log\left(\frac{\lambda_{\max}(\bar{V}(T-1))}{T} \right) + 2\log\frac{5}{\delta} \right) \\
&= \frac{16d\sigma^2}{\lambda_{\min}(C)^2} \left(\log\left(\frac{\lambda_{\max}\left(\sum_{t=0}^{T-1} AX(t)X(t)^\top A^\top\right)}{T} + 1 \right) + 2\log\frac{5}{\delta} \right).
\end{aligned}$$

Using the upper bound analysis in Lemma 6.14, we show that it suffices for T to be lower bounded as

$$T \geq \frac{16d\sigma^2}{\lambda_{\min}(C)^2} \left(\log\left(\alpha(A)^2 \left(b_T(\delta) + \|x(0)\|_\infty \right)^2 + 1 \right) + 2\log\frac{5}{\delta} \right),$$

when A is strictly stable and

$$T \geq \frac{16d\sigma^2}{\lambda_{\min}(C)^2}\left(\log\left(\alpha(A)^2\left(b_T(\delta)+\|x(0)\|_\infty\right)^2 T^{2l^*}+1\right)+2\log\frac{5}{\delta}\right),$$

when $\lambda_1(A) \leq 1+\frac{\rho}{T}$. Since, both these quantities on the RHS grow at most logarithmically with T, there exists T_0 such that it holds for all $T \geq T_0$. Combining the failure probability for all events, we get the desired result. □

6.9.3 Proof of estimation error results

In this section, we provide a detailed analysis of the average estimation error across the M systems for the estimator in (6.6) in presence of misspecifications $D_m \in \mathbb{R}^{d\times d}$:

$$A_m = \left(\sum_{i=1}^{k}\beta_m^*[i]W_i^*\right)+D_m, \quad \text{where } \|D_m\|_F \leq \zeta_m$$

In the presence of misspecifications, we have $\Delta \coloneqq \tilde{\Theta}^* - \widehat{\Theta} = VR + D$ where $V \in O^{d^2\times 2k}$ is an orthonormal matrix, $R \in \mathbb{R}^{2k\times M}$ and $D \in \mathbb{R}^{d^2\times M}$ is the misspecification error. The result in Theorem 6.2 can be obtained by simply substituting $\zeta_m = \bar{\zeta} = 0$ for all $m \in [M]_+$.

We start by fact that $(\widehat{W}, \widehat{B})$ minimize the squared loss in (6.6). However, in this case, we get an additional term dependent on the misspecifications D_m as follows:

$$\begin{aligned}&\frac{1}{2}\sum_{m=1}^{M}\left\|\tilde{X}_m(W^*\beta_m^* - \widehat{W}\hat{\beta}_m)\right\|_2^2\\ &\leq \sum_{m=1}^{M}\left\langle\tilde{\eta}_m, \tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\rangle + \sum_{m=1}^{M}2\left\langle\tilde{X}_m\tilde{D}_m, \tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\rangle. \end{aligned} \tag{6.27}$$

We can rewrite $\widehat{W}\hat{\beta}_m - W^*\beta_m^* = Ur_m$, for all $m \in [M]_+$, where $r_m \in \mathbb{R}^{2k}$ is an idiosyncratic projection vector for system m. We now show that the term on the RHS can be decomposed as follows:

Lemma 6.16. *Under Assumption 6.7, for any fixed orthonormal matrix $\bar{U} \in \mathbb{R}^{d^2\times 2k}$, the low rank*

part of the total squared error can be decomposed as follows:

$$\begin{aligned}
&\frac{1}{2}\sum_{m=1}^{M}\left\|\tilde{X}_m(W^*\beta_m^* - \widehat{W}\hat{\beta}_m)\right\|_F^2 \\
&\leq \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{2\sum_{m=1}^{M}\left\|\tilde{X}_m(W^*\beta_m^* - \widehat{W}\hat{\beta}_m)\right\|_2^2} + \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m\right\rangle \\
&\quad + \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{2\sum_{m=1}^{M}\left\|\tilde{X}_m(\bar{U}-U)r_m\right\|^2} + 2\sqrt{\bar{\lambda}}\bar{\zeta}\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\|_2^2}
\end{aligned} \tag{6.28}$$

Proof. We first define $\tilde{\Sigma}_{m,\mathrm{up}}$ and $\tilde{\Sigma}_{m,\mathrm{dn}}$ as the block diagonal matrices in $\mathbb{R}^{d^2\times d^2}$, with each $d\times d$ block of $\tilde{\Sigma}_{m,\mathrm{up}}$ and $\tilde{\Sigma}_{m,\mathrm{dn}}$ containing $\bar{\Sigma}_m$ and $\underline{\Sigma}_m$, respectively. Let $V_m = U^\top\tilde{\Sigma}_m U + U^\top\tilde{\Sigma}_{m,\mathrm{dn}}U$ be the regularized covairance matrix of projected covariates $\tilde{X}_m U$. For any orthonormal matrix $\bar{U}\in\mathbb{R}^{d^2\times 2k}$, we define $\bar{V}_m = \bar{U}^\top\tilde{\Sigma}_m\bar{U} + \bar{U}^\top\tilde{\Sigma}_{m,\mathrm{dn}}\bar{U}$, and proceed as follows:

$$\begin{aligned}
&\frac{1}{2}\sum_{m=1}^{M}\left\|\tilde{X}_m(W^*\beta_m^* - \widehat{W}\hat{\beta}_m)\right\|_2^2 \\
&\leq \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\rangle + \sum_{m=1}^{M} 2\left\langle \tilde{X}_m\tilde{D}_m, \tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\rangle \\
&\leq \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m\bar{U}r_m\right\rangle + \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m\right\rangle + \sum_{m=1}^{M} 2\left\langle \tilde{X}_m\tilde{D}_m, \tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\rangle \\
&\leq \sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top\tilde{X}_m\bar{U}\right\|_{\bar{V}_m^{-1}}\|r_m\|_{\bar{V}_m} + \sum_{m=1}^{M} 2\left\|\tilde{X}_m\tilde{D}_m\right\|_2\left\|\tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\|_2 \\
&\quad + \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m\right\rangle \\
&\leq \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top\tilde{X}_m\bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{\sum_{m=1}^{M}\|r_m\|_{V_m}^2} + 2\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m\tilde{D}_m\right\|_2^2}\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\|_2^2} \\
&\quad + \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top\tilde{X}_m\bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{\sum_{m=1}^{M}\left(\|r_m\|_{\bar{V}_m} - \|r_m\|_{V_m}\right)^2} + \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m\right\rangle.
\end{aligned}$$

The first equality uses the fact that the error matrix is low rank upto a misspecification term. The first inequality and last inequaliteis follow by using Cauchy-Schwarz inequality. Now, we can rewrite the error as:

$$
\begin{aligned}
&\frac{1}{2}\sum_{m=1}^{M}\left\|\tilde{X}_m(W^*\beta_m^* - \widehat{W}\hat{\beta}_m)\right\|_2^2 \\
&\leq \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{2\sum_{m=1}^{M}\|Ur_m\|_{\tilde{\Sigma}_m}^2} + \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m\right\rangle \\
&\quad + \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m(\bar{U}-U)r_m\right\|^2} \\
&\quad + 2\sqrt{\bar{\lambda}_m\zeta_m^2}\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\|_2^2} \\
&\leq \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{2\sum_{m=1}^{M}\left\|\tilde{X}_m(W^*\beta_m^* - \widehat{W}\hat{\beta}_m)\right\|_2^2} + \sum_{m=1}^{M}\left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m\right\rangle \\
&\quad + \sqrt{\sum_{m=1}^{M}\left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2}\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m(\bar{U}-U)r_m\right\|^2} + 2\sqrt{\bar{\lambda}}\bar{\zeta}\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m\left(\widehat{W}\hat{\beta}_m - W^*\beta_m^*\right)\right\|_2^2}.
\end{aligned}
$$

□

We will now bound each term individually. For the matrix $\bar{U}$, we choose it to be an element of $\mathcal{N}_\epsilon$ which is an ϵ-cover of the set of orthonormal matrices in $\mathbb{R}^{d^2\times 2k}$. Therefore, for any U, there exists $\bar{U}$ such that $\left\|\bar{U}-U\right\|_F \leq \epsilon$. We can bound the size of such a cover using Lemma 6.13 as $|\mathcal{N}_\epsilon| \leq \left(\frac{6\sqrt{d}}{\epsilon}\right)^{2d^2k}$.

Proposition 6.7 (Bounding $\sum_{m=1}^{M}\left\|\tilde{X}_m(\bar{U}-U)r_m\right\|^2$)**.** *Under Assumption 6.7, for the noise process $\{\eta_m(t)\}_{t=1}^{\infty}$ defined for each system, with probability at least $1-\delta_Z$, we have:*

$$
\sum_{m=1}^{M}\left\|\tilde{X}_m(\bar{U}-U)r_m\right\|^2 \lesssim \boldsymbol{\kappa}\epsilon^2\left(MT\,\mathrm{tr}\,C + \sigma^2\log\frac{2}{\delta_Z} + \bar{\lambda}\bar{\zeta}^2\right). \tag{6.29}
$$

Proof. In order to bound the term above, we use the squared loss inequality in (6.27) and Assump-

tion 6.7 as follows:

$$
\begin{aligned}
&\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
&\leq 2\left\langle Z, \mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\rangle + 2\sum_{m=1}^{M}\left\langle \tilde{X}_m\tilde{D}_m, \tilde{X}_m\left(W^*\beta_m^* - \widehat{W}\hat{\beta}_m\right)\right\rangle \\
&\leq 2\|Z\|_F\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + 2\sqrt{\sum_{m=1}^{M}\left\|\tilde{X}_m\tilde{D}_m\right\|_2^2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\leq 2\|Z\|_F\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + 2\sqrt{\sum_{m=1}^{M}\bar{\lambda}_m\|D_m\|_F^2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\leq 2\|Z\|_F\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + 2\sqrt{\bar{\lambda}\bar{\zeta}^2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F,
\end{aligned}
$$

which leads to the inequality $\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \leq 2\|Z\|_F + 2\sqrt{\bar{\lambda}\bar{\zeta}^2}$. Using the concentration result in Proposition 6.3, with probability at least $1 - \delta_Z$, we get

$$
\|Z\|_F \lesssim \sqrt{MT\,\mathrm{tr}\,C + \sigma^2\log\frac{1}{\delta_Z}}.
$$

Thus, we have $\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \lesssim 2\sqrt{MT\,\mathrm{tr}\,C + \sigma^2\log\frac{1}{\delta_Z}} + 2\sqrt{\bar{\lambda}\bar{\zeta}^2}$ with probability at least $1 - \delta_Z$. We now use this to bound the initial term:

$$
\begin{aligned}
\sum_{m=1}^{M}\left\|\tilde{X}_m(\bar{U} - U)r_m\right\|^2 &\leq \sum_{m=1}^{M}\left\|\tilde{X}_m\right\|^2\left\|\bar{U} - U\right\|^2\|r_m\|^2 \\
&\leq \sum_{m=1}^{M}\bar{\lambda}_m\epsilon^2\|r_m\|^2 = \sum_{m=1}^{M}\bar{\lambda}_m\epsilon^2\|Ur_m\|^2 \\
&\leq \sum_{m=1}^{M}\bar{\lambda}_m\epsilon^2\frac{\left\|\tilde{X}_mUr_m\right\|^2}{\underline{\lambda}_m} \leq \boldsymbol{\kappa}\epsilon^2\sum_{m=1}^{M}\left\|\tilde{X}_mUr_m\right\|^2 \\
&= \boldsymbol{\kappa}\epsilon^2\sum_{m=1}^{M}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
&\lesssim \boldsymbol{\kappa}\epsilon^2\left(MT\,\mathrm{tr}\,C + \sigma^2\log\frac{1}{\delta_Z} + \bar{\lambda}\bar{\zeta}^2\right).
\end{aligned}
$$

□

Proposition 6.8 (Bounding $\sum_{m=1}^{M}\left\langle\tilde{\eta}_m, \tilde{X}_m(U - \bar{U})r_m\right\rangle$)**.** *Under Assumption 6.5, Assumption 6.6*

and Assumption 6.7, with probability at least $1 - \delta_Z$ *we have:*

$$\sum_{m=1}^{M} \left\langle \tilde{\eta}_m, \tilde{X}_m(U - \bar{U})r_m \right\rangle \lesssim \sqrt{\boldsymbol{\kappa}}\epsilon \left(MT \operatorname{tr} C + \sigma^2 \log \frac{1}{\delta_Z} \right) + \sqrt{\boldsymbol{\kappa}\bar{\lambda}} \sqrt{MT \operatorname{tr} C + \sigma^2 \log \frac{1}{\delta_Z}} \epsilon\bar{\zeta}. \tag{6.30}$$

Proof. Using Cauchy-Schwarz inequality, we bound the term as follows:

$$\begin{aligned}
\sum_{m=1}^{M} \left\langle \tilde{\eta}_m, \tilde{X}_m(U - \bar{U})r_m \right\rangle &\leq \sqrt{\sum_{m=1}^{M} \|\tilde{\eta}_m\|^2} \sqrt{\sum_{m=1}^{M} \left\| \tilde{X}_m(\bar{U} - U)r_m \right\|^2} \\
&\lesssim \sqrt{MT \operatorname{tr} C + \sigma^2 \log \frac{1}{\delta_Z}} \sqrt{\boldsymbol{\kappa}\epsilon^2 \left(MT \operatorname{tr} C + \sigma^2 \log \frac{1}{\delta_Z} + \bar{\lambda}\bar{\zeta}^2 \right)} \\
&\lesssim \sqrt{\boldsymbol{\kappa}}\epsilon \left(MT \operatorname{tr} C + \sigma^2 \log \frac{1}{\delta_Z} \right) + \sqrt{\boldsymbol{\kappa}\bar{\lambda}} \sqrt{MT \operatorname{tr} C + \sigma^2 \log \frac{1}{\delta_Z}} \epsilon\bar{\zeta}.
\end{aligned}$$

□

Bounding $\sum_{m=1}^{M} \left\| z_m^\top \tilde{X}_m \bar{U} \right\|^2_{\bar{V}_m^{-1}}$: Now, we will show a martingale concentration result similar to Lemma A.4, but with a projection step to a low-rank subspace.

Proposition 6.9 (Multi-task self-normalized martingale bound)**.** *For an arbitrary orthonormal matrix* $\bar{U} \in \mathbb{R}^{d^2 \times 2k}$ *in the* ϵ*-cover* $\mathcal{N}_\epsilon$ *defined in Lemma 6.13, let* $\Sigma \in \mathbb{R}^{d^2 \times d^2}$ *be a positive definite matrix, and define* $S_m(\tau) = \tilde{\eta}_m(\tau)^\top \tilde{X}_m(\tau)\bar{U}$, $\bar{V}_m(\tau) = \bar{U}^\top \left(\tilde{\Sigma}_m(\tau) + \Sigma \right) \bar{U}$, *and* $V_0 = \bar{U}^\top \Sigma \bar{U}$. *Then, consider the following event:*

$$\mathcal{E}_1(\delta_U) \coloneqq \left\{ \omega \in \Omega : \sum_{m=1}^{M} \|S_m(T)\|^2_{\bar{V}_m^{-1}(T)} \leq 2\sigma^2 \log \left(\frac{\Pi_{m=1}^{M} \det(\bar{V}_m(T)) \det(V_0)^{-1}}{\delta_U} \right) \right\}.$$

For $\mathcal{E}_1(\delta_U)$*, we have:*

$$\mathbb{P}\left[\mathcal{E}_1(\delta_U)\right] \geq 1 - \left(\frac{6\sqrt{2k}}{\epsilon} \right)^{2d^2 k} \delta_U. \tag{6.31}$$

Proof of Proposition 6.9 First, using the vectors $\tilde{x}_{m,j}(t)$ defined in Section 6.3, for the matrix $\bar{U}$, define $\bar{x}_{m,j}(t) = \bar{U}^\top \tilde{x}_{m,j}(t) \in \mathbb{R}^{2k}$. It is straightforward to see that $\bar{V}_m(t) = \sum_{s=1}^{t} \sum_{j=1}^{d} \bar{x}_{m,j}(s)\bar{x}_{m,j}(s)^\top + V_0$.

Now, we show that the result can essentially be stated as a corollary of the following result of Hu et al. (2021) for stated for a univariate regression setting

Lemma 6.17 (Lemma 2 of Hu et al. (2021)). *Consider a fixed matrix $\bar{U} \in \mathbb{R}^{p\times 2k}$ and let $\bar{V}_m(t) = \bar{U}^\top(\sum_{s=0}^t x_m(s)x_m(s)^\top)\bar{U} + \bar{U}^\top V_0 \bar{U}$. Consider a noise process $w_m(t+1) \in \mathbb{R}$ adapted to the filtration $\mathcal{F}_t = \sigma(w_m(1),\dots,w_m(t), X_m(1),\dots,X_m(t))$. If the noise $w_m(t)$ is conditionally sub-Gaussian for all t: $\mathbb{E}\left[\exp\left(\lambda \cdot w_m(t+1)\right)\right] \le \exp\left(\lambda^2\sigma^2/2\right)$, then with probability at least $1-\delta$, for all $t \ge 0$, we have:*

$$\sum_{m=1}^M \left\| \sum_{s=0}^t w_m(s+1)\bar{U}^\top x_m(s) \right\|^2_{\bar{V}_m^{-1}(t)} \le 2\log\left(\frac{\Pi_{m=1}^M \left(\det(\bar{V}_m(t))\right)^{1/2} \left(\bar{U}^\top V_0 \bar{U}\right)^{-1/2}}{\delta} \right)$$

In order to use the above result in our case, we consider the martingale sum $\sum_{t=0}^T \sum_{j=1}^d \tilde{\eta}_{m,j}(t)\bar{U}^\top \tilde{x}_{m,j}(t)$. Under Assumption 6.5, we can use the same argument as in the proof of Lemma 2 in Hu et al. (2021) as:

$$\exp\left(\sum_{j=1}^d \frac{\eta_m(t+1)[j]}{\sigma} \langle \lambda, \bar{x}_{m,j}(t)\rangle \right) \le \exp\left(\sum_{j=1}^d \frac{1}{2}\langle \lambda, \bar{x}_{m,j}(t)\rangle^2 \right).$$

Thus, for a fixed matrix $\bar{U}$ and $T \ge 0$, with probability at least $1-\delta_U$,

$$\sum_{m=1}^M \|S_m(T)\|^2_{\bar{V}_m^{-1}(T)} \le 2\sigma^2 \log\left(\frac{\Pi_{m=1}^M \det(\bar{V}_m(T))\det(V_0)^{-1}}{\delta_U} \right)$$

Finally, we take a union bound over the ϵ-cover set of orthonormal matrices $\mathbb{R}^{d^2\times 2k}$ to bound the total failure probability by $|\mathcal{N}_\epsilon|\,\delta_U = \left(\frac{6\sqrt{2k}}{\epsilon}\right)^{2d^2k}\delta_U$.

Proof of Theorem 6.3 We now use the bounds we have shown for each term before and give the final steps by using the error decomposition in Lemma 6.16 as follows:

Proof. From Lemma 6.16, with a we have:

$$\begin{aligned}
&\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
&\le \sqrt{2\sum_{m=1}^M \left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|^2_{\bar{V}_m^{-1}}} \left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + \sum_{m=1}^M \left\langle \tilde{\eta}_m, \tilde{X}_m(U-\bar{U})r_m \right\rangle \\
&\quad + \sqrt{\sum_{m=1}^M \left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|^2_{\bar{V}_m^{-1}}} \sqrt{\sum_{m=1}^M \left\|\tilde{X}_m(\bar{U}-U)r_m\right\|^2} + 2\sqrt{\bar{\lambda}}\bar{\zeta}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F.
\end{aligned}$$

□

Now, substituting the termwise bounds from Proposition 6.7, Proposition 6.8 and Proposition 6.9, with probability at least $1 - |\mathcal{N}_\epsilon|\,\delta_U - \delta_Z$ we get:

$$\begin{aligned}
&\frac{1}{2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
&\lesssim \sqrt{\sigma^2 \log\left(\frac{\Pi_{m=1}^M \det(\bar{V}_m(t))\det(V_0)^{-1}}{\delta_U}\right)}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + \sqrt{\bar{\lambda}}\bar{\zeta}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\quad + \sqrt{\sigma^2 \log\left(\frac{\Pi_{m=1}^M \det(\bar{V}_m(t))\det(V_0)^{-1}}{\delta_U}\right)}\sqrt{\boldsymbol{\kappa}\epsilon^2\left(MT\,\mathrm{tr}\,C + \sigma^2\log\frac{1}{\delta_Z} + \bar{\lambda}\bar{\zeta}^2\right)} \\
&\quad + \sqrt{\boldsymbol{\kappa}}\epsilon\left(MT\,\mathrm{tr}\,C + \sigma^2\log\frac{1}{\delta_Z}\right) + \sqrt{\boldsymbol{\kappa}\bar{\lambda}}\sqrt{MT\,\mathrm{tr}\,C + \sigma^2\log\frac{1}{\delta_Z}}\epsilon\bar{\zeta}. \qquad (6.32)
\end{aligned}$$

In the definition of V_0, we now substitute $\Sigma = \underline{\lambda} I_{d^2}$ thereby implying $\det(V_0)^{-1} = \det(1/\underline{\lambda} I_{2k}) = (1/\underline{\lambda})^{2k}$. Similarly, for matrix $\bar{V}_m(T)$, we get $\det(\bar{V}_m(T)) \le \bar{\lambda}^{2k}$. Thus, substituting $\delta_U = \delta/3\,|\mathcal{N}|_\epsilon$ and $\delta_C = \delta/3$ (in Lemma 6.1), with probability at least $1 - 2\delta/3$, we get:

$$\begin{aligned}
\sum_{m=1}^M \left\|\tilde{\eta}_m^\top \tilde{X}_m \bar{U}\right\|_{\bar{V}_m^{-1}}^2 &\le \sigma^2 \log\left(\frac{\Pi_{m=1}^M \det(\bar{V}_m(t))\det(V_0)^{-1}}{\delta_U}\right) \\
&\le \sigma^2 \log\left(\frac{\bar{\lambda}}{\underline{\lambda}}\right)^{2Mk} + \sigma^2 \log\left(\frac{18k}{\delta\epsilon}\right)^{2d^2k} \\
&\lesssim \sigma^2 Mk \log \boldsymbol{\kappa}_\infty + \sigma^2 d^2 k \log\frac{k}{\delta\epsilon}.
\end{aligned}$$

Substituting this in (6.32) with $\delta_Z = \delta/3$, with probability at least $1-\delta$ we have:

$$
\begin{aligned}
&\frac{1}{2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
&\lesssim \sqrt{\sigma^2 Mk\log\boldsymbol{\kappa}_\infty + \sigma^2 d^2 k\log\frac{k}{\delta\epsilon}}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + \sqrt{\bar{\lambda}\bar{\zeta}}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\quad + \sqrt{\sigma^2 Mk\log\boldsymbol{\kappa}_\infty + \sigma^2 d^2 k\log\frac{k}{\delta\epsilon}}\sqrt{\boldsymbol{\kappa}\epsilon^2\left(MT\operatorname{tr}C + \sigma^2\log\frac{1}{\delta} + \bar{\lambda}\bar{\zeta}^2\right)} \\
&\quad + \sqrt{\boldsymbol{\kappa}}\epsilon\left(MT\operatorname{tr}C + \sigma^2\log\frac{1}{\delta}\right) + \sqrt{\boldsymbol{\kappa}\bar{\lambda}}\sqrt{MT\operatorname{tr}C + \sigma^2\log\frac{1}{\delta}}\epsilon\bar{\zeta} \\
&\lesssim \sqrt{c^2 Mk\log\boldsymbol{\kappa}_\infty + c^2 d^2 k\log\frac{k}{\delta\epsilon}}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F + \sqrt{\bar{\lambda}\bar{\zeta}}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\quad + \sqrt{c^2 Mk\log\boldsymbol{\kappa}_\infty + c^2 d^2 k\log\frac{k}{\delta\epsilon}}\sqrt{\boldsymbol{\kappa}\epsilon^2\left(c^2 dMT + c^2\log\frac{1}{\delta} + \bar{\lambda}\bar{\zeta}^2\right)} \\
&\quad + \sqrt{\boldsymbol{\kappa}}\epsilon\left(c^2 dMT + c^2\log\frac{1}{\delta}\right) + \sqrt{\boldsymbol{\kappa}\bar{\lambda}}\sqrt{c^2 dMT + c^2\log\frac{1}{\delta}}\epsilon\bar{\zeta}.
\end{aligned}
$$

Noting that $k \le d$ and $\log\frac{1}{\delta} \lesssim d^2 k\log\frac{k}{\delta\epsilon}$, by setting $\epsilon = \frac{k}{\sqrt{\boldsymbol{\kappa}}dT}$ we have:

$$
\begin{aligned}
&\frac{1}{2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
&\lesssim \left(\sqrt{c^2 Mk\log\boldsymbol{\kappa}_\infty + c^2 d^2 k\log\frac{k}{\delta\epsilon}} + \sqrt{\bar{\lambda}\bar{\zeta}}\right)\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\quad + \sqrt{c^2 Mk\log\boldsymbol{\kappa}_\infty + c^2 d^2 k\log\frac{k}{\delta\epsilon}}\sqrt{\boldsymbol{\kappa}\epsilon^2\left(c^2 dMT + c^2 d^2 k\log\frac{k}{\delta\epsilon} + \bar{\lambda}\bar{\zeta}^2\right)} \\
&\quad + \sqrt{\boldsymbol{\kappa}}\epsilon\left(c^2 dMT + c^2 d^2 k\log\frac{k}{\delta\epsilon}\right) + \sqrt{\left(c^2 dMT + c^2 d^2 k\log\frac{k}{\delta\epsilon}\right)\boldsymbol{\kappa}\bar{\lambda}\epsilon^2\bar{\zeta}^2} \\
&\lesssim \left(\sqrt{c^2 Mk\log\boldsymbol{\kappa}_\infty + c^2 d^2 k\log\frac{\boldsymbol{\kappa}dT}{\delta}} + \sqrt{\bar{\lambda}\bar{\zeta}}\right)\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&\quad + \sqrt{c^2 Mk\log\boldsymbol{\kappa}_\infty + c^2 d^2 k\log\frac{\boldsymbol{\kappa}dT}{\delta}}\sqrt{c^2\left(\frac{k^2 M}{dT} + \frac{k^3}{T^2}\log\frac{\boldsymbol{\kappa}dT}{\delta} + \frac{\bar{\lambda}k^2\bar{\zeta}^2}{d^2T^2}\right)} \\
&\quad + c^2\left(Mk + \frac{dk^2}{T}\log\frac{\boldsymbol{\kappa}dT}{\delta}\right) + \sqrt{c^2\left(\frac{k^2 M}{dT} + \frac{k^3}{T^2}\log\frac{\boldsymbol{\kappa}dT}{\delta}\right)\bar{\lambda}\bar{\zeta}^2},
\end{aligned}
$$

which gives the simplified bound:

$$\begin{aligned}
&\frac{1}{2}\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \\
\lesssim\ &\left(\sqrt{c^2\left(Mk\log\boldsymbol{\kappa}_\infty + d^2k\log\frac{\boldsymbol{\kappa}dT}{\delta}\right)} + \sqrt{\bar{\lambda}}\bar{\zeta}\right)\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F \\
&+ c^2\left(Mk\log\boldsymbol{\kappa}_\infty + \frac{d^2k}{T}\log\frac{\boldsymbol{\kappa}dT}{\delta}\right) + c\sqrt{\frac{\bar{\lambda}\bar{\zeta}^2}{T}\left(Mk\log\boldsymbol{\kappa}_\infty + \frac{d^2k}{T}\log\frac{\boldsymbol{\kappa}dT}{\delta}\right)}.
\end{aligned}$$

The quadratic inequality for the prediction error $\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2$ implies the following bound with probability at least $1-\delta$:

$$\left\|\mathcal{X}(W^*B^* - \widehat{W}\widehat{B})\right\|_F^2 \lesssim c^2\left(Mk\log\boldsymbol{\kappa}_\infty + d^2k\log\frac{\boldsymbol{\kappa}dT}{\delta}\right) + \bar{\lambda}\bar{\zeta}^2.$$

Since $\underline{\lambda} = \min_m \underline{\lambda}_m$, we can convert the prediction error bound to an estimation error bound as follows:

$$\left\|W^*B^* - \widehat{W}\widehat{B}\right\|_F^2 \lesssim \frac{c^2\left(Mk\log\boldsymbol{\kappa}_\infty + d^2k\log\frac{\boldsymbol{\kappa}dT}{\delta}\right)}{\underline{\lambda}} + \boldsymbol{\kappa}_\infty\bar{\zeta}^2,$$

which finally implies the estimation error bound for the solution of (6.6):

$$\sum_{m=1}^{M}\left\|\widehat{A}_m - A_m\right\|_F^2 \lesssim \frac{c^2\left(Mk\log\boldsymbol{\kappa}_\infty + d^2k\log\frac{\boldsymbol{\kappa}dT}{\delta}\right)}{\underline{\lambda}} + (\boldsymbol{\kappa}_\infty + 1)\bar{\zeta}^2.$$

6.9.4 Incorporating control inputs in joint system identification

We will now show an estimation error result similar to Theorem 6.2 when for estimating (A, B) matrices when the linear time-series in (6.1) is used to generate the data. In the second phase of Algorithm 6.2, we use the controllers $\widehat{K}_{m,j}$ for each epoch, i.e., $u_m(t) = \widehat{K}_{m,j}x_m(t) + \sigma_{m,j}g_m(t)$ where $g_m(t) \sim N(0, I)$. Using $z_m(t)$ to denote the concatenated state-input vector, we study the following linear system for any epoch j:

$$x_m(t+1) = \begin{bmatrix} A & B \end{bmatrix} z_m(t) + \eta_m(t+1). \tag{6.33}$$

In this section, our main goal in this section will be to show an analogue of Lemma 6.1 for the concatenated linear system. To that end, redefine the covariance matrix as $\Sigma_m := \sum_{t=\tau_{j-1}}^{\tau_j - 1} z_m(t)z_m(t)^\top$. We prove the upper bounds and lower bounds for Σ_m below:

6.9.4.1 An upper bound on the covariance matrix

For the upper bound, we note that the overall system can be written down as:

$$x_m(t) = A_m x_m(t-1) + B_m u_m(t-1) + \eta_m(t), \quad u_m(t) = \widehat{K}_{m,j} x_m(t) + \sigma_{m,j} g_m(t) \tag{6.34}$$

where $\eta_m(t) \sim N(0, I_{d_x})$ (Assumption 6.3) and $g_m(t) \sim N(0, I_{d_u})$. To bound the covariance matrix, we directly use the following result from Simchowitz and Foster (2020):

Lemma 6.18 (Corollary of Lemma G.1, Simchowitz and Foster (2020))**.** *If $\widehat{K}_{m,j}$ is a stabilizing controller, then for all $\tau_{j-1} \geq \max\left(\mathcal{J}_{m,0} \left\|\!\left| P_m \right|\!\right\|^3, \sigma_m^4 \Psi_{B,m}^4\right)$, with probability $1-\delta$, we have:*

$$\sum_{s=\tau_j}^{\tau_{j+1}-1} \|x_m(s)\|_2^2 + \|u_m(s)\|_2^2 \lesssim \tau_j d \left\|\!\left| P_m \right|\!\right\| \log \frac{1}{\delta} + \sigma_m^2 \Psi_{B,m}^2 \left\|\!\left| P_m \right|\!\right\|^4 \log \frac{1}{\delta} \tag{6.35}$$

which implies the upper bound on $\Sigma_{m,j} \preceq \lambda_{m,j} I$ where $\lambda_{m,j}$ is the expression on the RHS of (6.35)*.*

Proof. We first state the result of Lemma G.1 from Simchowitz and Foster (2020). From the perturbation bounds on the controller, we firstly know that for all systems $m \in [M]_+$, the controllers $\widehat{K}_{m,j}$ are stabilizing. For such stabilizing controllers, say K, the 2nd result in Lemma G.1 states that with probability at least $1-\delta$, we have:

$$\begin{aligned}
&\sum_{s=1}^{t} x(s)^\top x(s) + u(s)^\top u(s) \\
&\leq t f_K + 2\sigma_u^2 dt \left(1 + \left\|\!\left| B \right|\!\right\|^2 \left\|\!\left| P_K \right|\!\right\|\right) + 2x(1)^\top P_K x(1) \\
&\quad + O\left(\sqrt{dt \log \frac{1}{\delta}} + \log \frac{1}{\delta}\right) \left((1 + \sigma_u^2 \left\|\!\left| B \right|\!\right\|^2) \left\|\!\left| I + K^\top K \right|\!\right\| \left\|\!\left| A + BK \right|\!\right\|_{\mathcal{H}_\infty}^2 + \sigma_u^2\right)
\end{aligned}$$

We will now substitute the value for each expression in the RHS of the above inequality for each system m in any epoch j:

1. $f_K := \mathrm{tr}(\mathrm{dlyap}\left(A_m + B_m \widehat{K}_{m,j}, I\right))$. For this term, by using Lemma 6.5 and Theorem 6.6, we have:

$$\mathrm{tr}(\mathrm{dlyap}\left(A_m + B_m \widehat{K}_{m,j}, I\right)) \leq \mathrm{tr}\left(P_\infty(\widehat{K}_{m,j}; A_m, B_m)\right) \lesssim d_x \left\|\!\left| P_m \right|\!\right\|$$

2. By using Theorem 6.6, we have $1 + \left\|\!\left| B_m \right|\!\right\|^2 \left\|\!\left| P_{m,j} \right|\!\right\| \lesssim \Psi_{B,m}^2 \left\|\!\left| P_m \right|\!\right\|$.

3. Again, using Theorem 6.6, we have

$$\begin{aligned}
&(1+\sigma_{m,j}^2 \left\|\left| B_m \right|\right\|^2) \left\|\left| I + \widehat{K}_{m,j}^\top \widehat{K}_{m,j} \right|\right\| \left\|\left| A_m + B_m \widehat{K}_{m,j} \right|\right\|_{\mathcal{H}_\infty}^2 \\
\lesssim\ &(1+\sigma_{m,j}^2 \left\|\left| B_m \right|\right\|^2)(1+\left\|\left| \widehat{K}_{m,j} \right|\right\|^2) \left\|\left| P_m \right|\right\|^3 \\
\lesssim\ &(1+\sigma_{m,j}^2 \left\|\left| B_m \right|\right\|^2) \left\|\left| P_m \right|\right\|^4
\end{aligned}$$

4. Lastly, the term $x_m(\tau_j)^\top P_{m,j} x_m(\tau_j)$ can be bounded as follows:

$$\begin{aligned}
x_m(\tau_j)^\top P_{m,j} x_m(\tau_j) &\leq \|x_m(\tau_j)\|_2^2 \left\|\left| P_{m,j} \right|\right\| \\
&\lesssim \|x_m(\tau_j)\|_2^2 \left\|\left| P_m \right|\right\| \\
&\lesssim \Psi_{B,m} \mathcal{J}_{m,0} \left\|\left| P_m \right|\right\|^4 \log \frac{1}{\delta}
\end{aligned}$$

Thus, by simnplifying the expressions, the covariance matrix can be bounded by the expression:

$$\begin{aligned}
\sum_{s=\tau_j}^{2\tau_j-1} x_m(s)^\top x_m(s) + u_m(s)^\top u_m(s) \lesssim\ &\tau_{j-1} \log \frac{1}{\delta} d \left(\left\|\left| P_m \right|\right\| + \sigma_{m,j}^2 \Psi_{B,m}^2 \left\|\left| P_m \right|\right\| \right) \\
&+ \Psi_{B,m} \mathcal{J}_{m,0} \left\|\left| P_m \right|\right\|^4 \log \frac{1}{\delta} + \sigma_m^2 \Psi_{B,m}^2 \left\|\left| P_m \right|\right\|^4 \log \frac{1}{\delta}
\end{aligned}$$

Thus, if $\tau_j \geq \mathcal{J}_{m,0} \left\|\left| P_m \right|\right\|^3$ and $\tau_j \geq \sigma_m^4 \Psi_{B,m}^4$, then, we can simplify the upper bound to:

$$\sum_{s=\tau_j}^{2\tau_j-1} x_m(s)^\top x_m(s) + u_m(s)^\top u_m(s) \lesssim \tau_j d \left\|\left| P_m \right|\right\| \log \frac{1}{\delta} + \sigma_m^2 \Psi_{B,m}^2 \left\|\left| P_m \right|\right\|^4 \log \frac{1}{\delta}$$

□

6.9.4.2 Lower bounding the concatenated covariance matrix

We show that the covariance matrix can be lower bounded by using Lemma E.4 in Simchowitz and Foster (2020). Firstly, note that the linear system on the complete state vector can be written as follows:

$$z_m(t+1) = \begin{bmatrix} I_{d_x} \\ \widehat{K}_{m,j} \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix} z_m(t) + \begin{bmatrix} I_{d_x} & 0 \\ \widehat{K}_{m,j} & \sigma_{m,j} I_{d_u} \end{bmatrix} \begin{bmatrix} \eta_m(t+1) \\ g_m(t+1) \end{bmatrix} \tag{6.36}$$

For this system, we can use the aforementioned result with the noise $\bar{\eta}_m(t) := \begin{bmatrix} I_{d_x} & 0 \\ \widehat{K}_{m,j} & \sigma_{m,j} I_{d_u} \end{bmatrix} \begin{bmatrix} \eta_m(t+1) \\ g_m(t+1) \end{bmatrix}$ and the upper bound on the covariance matrix stated in Lemma 6.18.

Lemma 6.19 (Corollary of Lemma 6.15)**.** *For all $m \in [M]$, if the epoch size per system is large enough such that:*

$$\tau_j \gtrsim \max\left(\left(d\log\left(\left(1+\frac{2\left\|P_m\right\|}{\sigma_m^2}\right)\left(d\Psi_{B,m}^2\left\|P_m\right\|^4\log\frac{1}{\delta}\right)\right)\right), \mathcal{J}_{m,0}\left\|P_m\right\|^3, \sigma_m^4\Psi_{B,m}^4\right)$$

then with probability at least $1-3M\delta$, the sample covariance matrix $\Sigma_{m,j}$ for system m can be bounded from below as follows: $\Sigma_{m,j} \succeq \frac{\sigma_{m,j}^2\tau_j}{\sigma_{m,j}^2+2\left\|\widehat{K}_{m,j}\right\|^2} I$.

Proof. To begin, we state Lemma E.4 from Simchowitz and Foster (2020) below:

Lemma 6.20 (Lemma E.4, Simchowitz and Foster (2020))**.** *Suppose that $z_t|\mathcal{F}_{t-1} \sim N(\bar{z}_t, C_t)$, where $z_t \in \mathbb{R}^d$ and $C_t \in \mathbb{R}^{d\times d}$ are $\mathcal{F}_{t-1}$-measurable and $C_t \succeq C \succeq 0$. Let $\mathcal{E}$ be any event for which $\bar{\Sigma}_T := \mathbb{E}\left[\Sigma\mathbb{1}(\mathcal{E})\right]$ satisfies $\mathrm{tr}(\bar{\Sigma}_T) \leq TJ$ for some $J \geq 0$. Then, for*

$$T \geq \frac{2000}{9}\left(2d\log\frac{100}{3} + d\log\frac{J}{\lambda_{\min}(C)}\right),$$

it holds that, for $\Sigma_0 := \frac{9T}{1600}\Sigma$

$$\mathbb{P}\left[\left\{\Sigma \not\succeq \frac{9T}{1600}\Sigma\right\} \cap \mathcal{E}\right] \leq 2\exp\left(-\frac{9}{2000(d+1)}T\right)$$

We will use this result for $z_t := z_m(t)$ and $\Sigma_T = \sum_{t=\tau_j}^{2\tau_j-1} z_m(t)z_m(t)^\top$ where $\Sigma_t = \begin{bmatrix} I_{d_x} & \widehat{K}_{m,j}^\top \\ \widehat{K}_{m,j} & \widehat{K}_{m,j}\widehat{K}_{m,j}^\top + \sigma_{m,j}^2 I_{d_u} \end{bmatrix}$. In order to apply the Lemma, we need:

(1) Lower bound on the smallest eigenvalue of the second moment of the noise vector $\bar{\eta}_m(t)$.

(2) Upper bound on the term $\lambda_{\max}\left(\mathbb{E}\left[\sum_{t=\tau_j}^{2\tau_j-1} Z_m(t)Z_m(t)^\top\right]\right)$ used in the self-normalized bound in the proof.

For (1), note that both $\eta_m(t)$ and $g_m(t)$ are iid Gaussian vectors. Therefore, for the lower bound, we simply need to lower bound the smallest eigenvalue of $\begin{bmatrix} I_{d_x} & \widehat{K}_{m,j}^\top \\ \widehat{K}_{m,j} & \widehat{K}_{m,j}\widehat{K}_{m,j}^\top + \sigma_{m,j}^2 I_{d_u} \end{bmatrix}$. By using Lemma F.6 of Dean et al. (2018), we can show that the least eigenvalue can be bounded as:

$$\lambda_{\min}\left(\begin{bmatrix} I_{d_x} & \widehat{K}_{m,j}^\top \\ \widehat{K}_{m,j} & \widehat{K}_{m,j}\widehat{K}_{m,j}^\top + \sigma_{m,j}^2 I_{d_u} \end{bmatrix}\right) \geq \sigma_{m,j}^2 \min\left(\frac{1}{2}, \frac{1}{\sigma_{m,j}^2 + 2\left\|\widehat{K}_{m,j}\right\|^2}\right)$$

For (2), we can use the same upper bound as shown in Lemma 6.18 for the expectation of the matrix $\Sigma_{m,j}$ as follows:

$$\operatorname{tr}\left(\bar{\Sigma}_{m,j}\right) \leq \tau_j \left(d \left\|\!\left| P_m \right|\!\right\| \log \frac{1}{\delta} + \sigma_m^2 \Psi_{B,m}^2 \left\|\!\left| P_m \right|\!\right\|^4 \log \frac{1}{\delta} \right)$$

when $\tau_j \geq \mathcal{J}_{m,0} \left\|\!\left| P_m \right|\!\right\|^3$ and $\tau_j \geq \sigma_m^4 \Psi_{B,m}^4$. We will now substitute the upper bound on $\left\|\!\left| \widehat{K}_{m,j} \right|\!\right\|^2$ from Theorem 6.6. Thus, using the upper bound on the value $\operatorname{tr}\left(\bar{\Sigma}_{m,j}\right)$, for $\tau_j \gtrsim d \log \frac{1}{\delta}$ and

$$\begin{aligned}
\tau_j &\geq \frac{2000}{9} \left(2d \log \frac{100}{3} + d \log \frac{\left(\sigma_{m,j}^2 + 2 \left\|\!\left| \widehat{K}_{m,j} \right|\!\right\|^2 \right) \left(d \Psi_{B,m}^2 \left\|\!\left| P_m \right|\!\right\|^4 \log \frac{1}{\delta} \right)}{\tau_j \sigma_{m,j}^2} \right) \\
&\gtrsim \frac{2000}{9} \left(2d \log \frac{100}{3} + d \log \left(\left(1 + \frac{2 \left\|\!\left| P_m \right|\!\right\|}{\sigma_m^2} \right) \left(d \Psi_{B,m}^2 \left\|\!\left| P_m \right|\!\right\|^4 \log \frac{1}{\delta} \right) \right) \right),
\end{aligned}$$

with probability at least $1 - M\delta$:

$$\lambda_{\min}\left(\Sigma_{m,j}\right) \gtrsim \frac{\tau_j \sigma_{m,j}^2}{\sigma_{m,j}^2 + 2 \left\|\!\left| \widehat{K}_{m,j} \right|\!\right\|^2}$$

which is the desired result. □

6.9.4.3 Final estimation error bound for joint system identification

As described before, the structural assumption in Assumption 6.1 allows us to use the same estimation error upper bound where we simply replace d^2 by $(d_x + d_u)d_u$, $\bar{\lambda}_m$ by $\lambda_{m,j}$ and $\underline{\lambda}_m$ by $\frac{\tau_j}{8+8\left\|\!\left| \widehat{K}_{m,j} \right|\!\right\|^2}$. Therefore, we get the final result as follows:

Theorem (Corollary of Theorem 6.2)**.** *Under Assumption 6.1, and for epochs j such that*

$$\tau_j \gtrsim \max\left(\left(d \log \left(\left(1 + \frac{2 \left\|\!\left| P_m \right|\!\right\|}{\sigma_m^2} \right) \left(d \Psi_{B,m}^2 \left\|\!\left| P_m \right|\!\right\|^4 \log \frac{1}{\delta} \right) \right) \right), \mathcal{J}_{m,0} \left\|\!\left| P_m \right|\!\right\|^3, \sigma_m^4 \Psi_{B,m}^4 \right),$$

the estimator in (6.19) *returns $(\widehat{A}_{m,j}, \widehat{B}_{m,j})$ for each system $m \in [M]$ and epoch j, such that with probability at least $1 - M\delta/8$, the following holds:*

$$\frac{1}{M} \sum_{m=1}^{M} \left(\left\| \widehat{A}_{m,j} - A_m \right\|_F^2 + \left\| \widehat{B}_{m,j} - B_m \right\|_F^2 \right) \lesssim \max_m \frac{\left(\sigma_{m,j}^2 + 2 \left\|\!\left| P_m \right|\!\right\| \right) \left(k \log \boldsymbol{\kappa} + \frac{dd_x k}{M} \log \frac{d\boldsymbol{\kappa}}{\delta} \right)}{\sigma_{m,j}^2 \tau_j}$$

where $\boldsymbol{\kappa} = \max_m d \log\left(\left(1 + \frac{2\|\!|P_m|\!\|}{\sigma_m^2}\right)\left(d\Psi_{B,m}^2 \|\!|P_m|\!\|^4 \log\frac{1}{\delta}\right)\right)$.

The result follows by upper bounding $\left\|\!\left|\widehat{K}_{m,j}\right|\!\right\|^2$ by $\|\!|P_m|\!\|$ using Theorem 6.6.

CHAPTER 7

Concluding Remarks

In Chapter 1, we outlined the goal of this thesis as studying the effect of linear/low-rank structures in the model of an environment on the sample efficiency of reinforcement learning. As stated in the beginning, we examine such structural assumptions in various problem settings and highlight the utility of such environment structure in efficient exploration, multi-task learning, and representation learning for MDPs. In particular, we presented sample complexity results in the following settings:

1. In Chapter 3, for contextual MDPs, we presented PAC mistake bounds for contextual mappings which (1) vary smoothly with the context and (2) are linear functions of the context. Our analysis showed that under the weaker assumption of smoothness, the sample complexity can be exponential in the context dimension. On the other hand, for linear structures, we proposed PAC-efficient and regret minimizing algorithms which are provably efficient, both statistically and computationally.

2. In Chapter 4, we studied the sample complexity of RL in complex environments when the true model can be expressed as a feature based linear combination of a known basis set of models. Our sample complexity bounds scale with the dimension of the feature representation and do not scale with the size of the environment. We also showed hardness results for feature selection in our linear model ensemble setting.

3. Next, in Chapter 5, we study a representation learning problem for low-rank MDPs. We proposed the algorithm MOFFLE, which is the first model-free representation learning and exploration algorithm for low-rank MDPs. The key contribution here is the novel representation learning objective which utilizes the low-rank structure in the underlying transition dynamics of the environment.

4. Lastly, we considered a multi task LQR problem where the system's transition matrices share a common linear basis. We first proposed a joint estimator for the transition matrices of LTIDS and showed finite time estimation error bounds for the joint estimator. We then

used this joint estimator in a certainty equivalence based controller and analyzed its regret behavior. Our results showed the improvement in estimation error and regret bounds when data is pooled across multiple systems under the structural assumption.

7.1 Discussion and Future work

In this section, we discuss the limitations and potential future work directions for this thesis.

Contextual MDPs In Chapter 3, we considered the problem of learning in a sequence of tabular MDPs where a context representation is given to the agent at the beginning of each episode. There are many future directions which we can consider in this setting, which we discuss below. Firstly, our PAC and regret guarantees have been established for (generalized) linear mappings. Similar results can be shown for contextual mappings which are non-linear but have a bounded structural complexity, like bounded Eluder dimension (Russo and Van Roy, 2013). Further, our algorithms show that the contextual representation can be efficiently used for sharing data across tasks, but only hold for tabular domains. Extending these ideas to non-tabular state spaces is an interesting problem, and we can start by looking at large but structured state spaces like the block MDP setting (Du et al., 2019a). Finally, in many applications, the contextual information is present but often latent to the agent. As such, it is important to study scenarios where the latent representation can be quickly identified from few-shot interaction data, in order to quickly adapt to the environment. Recent results like Kwon et al. (2021) have made progress in this direction.

Representation learning In Chapter 4 and Chapter 5, we considered two different feature selection/learning problems. In the former, we showed that selecting among features at different resolutions can be information-theoretically difficult in the absence of additional prior knowledge. In order to resolve this issue, we showed positive results when the agent knows the optimal value for the given MDP. It is an interesting problem to consider other possible algorithmic schemes based on weaker prior knowledge assumptions. In Chapter 5, we proposed the MOFFLE algorithm for representation learning in low-rank MDPs. Our work currently has two weaknesses: the algorithm is not provably computationally efficient and requires the reachability assumption for statistical efficiency (it effectively implies a small non-negative rank for the transition matrix). Hence, coming up with more efficient algorithms, removing reachability and studying the phenomenon empirically are all important threads for future work. Lastly, our learnt representation in MOFFLE can only be used for downstream planning for the same MDP. The key utility of representation learning is to transfer the representation to other tasks which share a similar structure. Thus, it will be interesting

to consider a multi-task or continual learning setting where a common feature ϕ^* is shared among MDPs and investigate if representation learning can be used for efficient learning.

Multi-task control In Chapter 6, we considered a multi-task adaptive control problem where the transition matrices of different systems shared a common linear basis. In the first part, we established system identification results for non-explosive systems and showed improved rates. However, our analysis showed estimation error bounds under the Frobenius norm and ignores any matrix structure which is present in the LTIDS setting. It will be important to explore a general estimator, and analysis thereof, such that estimation error bounds can be given for explosive, non-explosive as well as system matrices which have both sub-unit and explosive eigenvalues. In the second part, we proposed and analysed a multi-task controller where we observed that the improvement can only be seen for the components of the regret analysis which do not depend on the inherent random fluctuations of the systems. It will be interesting to see if another notion of regret (expected regret) can be studied, and show that the rates improve substantially for the multi-task case. Lastly, in the single task LQR problem, the problem of adaptive stabilization has been previously studied in Faradonbeh et al. (2018b) and can be used in the initial exploration phases of various algorithms. However, the problem of showing improved rates of stabilization for our multi-task setting is an open problem and is left for future work.

APPENDIX A

Basic Probabilistic Inequalities

In this chapter, we review the major probabilistic inequalities used in this thesis.

Theorem A.1 (Hoeffding's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be mean-zero independent real-valued random variables with $X_j \in [a_j, b_j]$ and $Y \coloneqq \sum_{j=1}^n X_j$. For any $\varepsilon \geq 0$, we have:*

$$\mathbb{P}\left[|Y| \geq \varepsilon\right] \leq 2\exp\left(\frac{-2n^2\varepsilon^2}{\sum_{j=1}^n (b_j - a_j)^2}\right). \tag{A.1}$$

As one can note, Hoeffding's bound uses the bounds on the random variables to give a high-probability concentration result. However, when the variances of the random variables are small, we can get a sharper bound as follows:

Theorem A.2 (Bernstein's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be mean-zero independent real-valued random variables with $|X_i| \leq c$ for all i and let $\sigma^2 \coloneqq \frac{1}{n}\sum_{i=1}^n \mathrm{Var}[X_i]$. Denote $Y \coloneqq \sum_{j=1}^n X_j$. For any $\varepsilon \geq 0$, we have:*

$$\mathbb{P}\left[|Y| \geq \varepsilon\right] \leq 2\exp\left(\frac{-2n\varepsilon^2}{2\sigma^2 + 2c\varepsilon/3}\right). \tag{A.2}$$

Therefore, for random variables with small variance, the Bernstein's inequality can be sharper than Hoeffding's.

The above results are for iid random variables. Below, we state the commonly used concentration result for martingales:

Theorem A.3 (Azuma-Hoeffding's inequality)**.** *Let $X_1, X_2, \ldots, X_n$ be martingale sequence with $|X_i - X_{i-1}| \leq c_i$ for all i. Then for any positive N any $\varepsilon \geq 0$, we have:*

$$\mathbb{P}\left[|X_N - X_0| \geq \varepsilon\right] \leq 2\exp\left(\frac{-2n\varepsilon^2}{2\sum_{i=1}^N c_i^2}\right). \tag{A.3}$$

Next, we state a concentration inequality for self-normalized martingales, which is Lemma 8 and Lemma 9 in the work of Abbasi-Yadkori et al. (2011). More details about self-normalized process can be found in the work of Victor et al. (2009).

Lemma A.4. *Let $\{\mathcal{F}_t\}_{t=0}^{\infty}$ be a filtration. Let $\{\eta_t\}_{t=1}^{\infty}$ be a real valued stochastic process such that η_t is $\mathcal{F}_t$ measurable and η_t is conditionally σ-sub-Gaussian for some $R > 0$, i.e.,*

$$\forall \lambda \in \mathbb{R}, \mathbb{E}\left[\exp(\lambda \eta_t) | \mathcal{F}_{t-1}\right] \leq e^{\frac{\lambda^2 \sigma^2}{2}}$$

Let $\{X_t\}_{t=1}^{\infty}$ be an $\mathbb{R}^d$-valued stochastic process such that X_t is $\mathcal{F}_t$ measurable. Assume that V is a $d \times d$ positive definite matrix. For any $t \geq 0$, define

$$\bar{V}_t = V + \sum_{s=1}^{t} X_s X_s^{\top}, \quad S_t = \sum_{s=1}^{t} \eta_{s+1} X_s.$$

Then with probability at least $1 - \delta$, for all $t \geq 0$ we have

$$\|S_t\|_{\bar{V}_t^{-1}}^2 \leq 2\sigma^2 \log\left(\frac{\det\left(\bar{V}_t\right)^{1/2} \det\left(V\right)^{-1/2}}{\delta}\right).$$

APPENDIX B

Missing Results for Chapter 5

B.1 FQI Planning Results

In this appendix, we provide various FQI (Fitted Q-iteration) planning results. In Appendix B.1.1, we provide the general framework of FQI algorithms. In Appendix B.1.2, we show the sample complexity of FQI-FULL-CLASS that handles the offline planning for a class of rewards. In Appendix B.1.3, we provide the sample complexity guarantee for planning for the elliptical reward class. In Appendix B.1.4, we discuss the sample complexity result of planning with the learned feature $\bar{\phi}$. We want to mention that we abuse some notations in this section. For example, $\mathcal{F}_h$, $\mathcal{G}_h$ would have different meanings from the main text. However, they should be clear within the context.

B.1.1 FQI planning algorithm

In this part, we present a general framework of FQI planner. Algorithm B.1 subsumes three different algorithms: FQI-FULL-CLASS, FQI-REPRESENTATION, and FQI-ELLIPTICAL. FQI-FULL-CLASS and FQI-REPRESENTATION will be used to plan for a finite deterministic reward class, while FQI-ELLIPTICAL is specialized in planning for the elliptical reward class. This leads to the different bounds of parameters in the Q-value function classes and different clipping thresholds of the state-value functions. In addition, for FQI-FULL-CLASS and FQI-ELLIPTICAL, we use all features in Φ to construct the Q-value function classes, while in FQI-REPRESENTATION we only utilize the the learnt representation $\bar{\phi}$. The details can be found below.

Algorithm B.1 FQI: Fitted Q-Iteration

input: (1) exploratory dataset $\{\mathcal{D}\}_{0:H-1}$ sampled from ρ_{h-3}^{+3} (or $\{\mathcal{D}^{\mathrm{o}}\}_{0:H-1}$ sampled from ρ_{h-2}^{+2}), (2) reward function $R = R_{0:H-1} : \mathcal{S} \times \mathcal{A} \to [0,1]$, (3) function class: (i) for FQI-FULL-CLASS, $\mathcal{F}_h(R_h) := \{R_h + \langle \phi_h, w_h \rangle : \|w_h\|_2 \le H\sqrt{d}, \phi_h \in \Phi_h\}, h \in [H]$; (ii) for FQI-REPRESENTATION, $\mathcal{F}_h(R_h) := \{R_h + \mathrm{clip}_{[0,H]}\left(\langle \bar{\phi}_h, w_h \rangle\right) : \|w_h\|_2 \le B\}, h \in [H]$; (iii) for FQI-ELLIPTICAL, $\mathcal{F}_h(R_h) := \{R_h + \langle \phi_h, w_h \rangle : \|w_h\|_2 \le \sqrt{d}, \phi_h \in \Phi_h\}, h \in [H]$.

Set $\hat{V}_H(s) = 0$.

for $h = H-1, \ldots, 0$ **do**

Pick n samples $\left\{(s_h^{(i)}, a_h^{(i)}, s_{h+1}^{(i)})\right\}_{i=1}^n$ from the exploratory dataset $\mathcal{D}_h$.

Solve least squares problem:

$$\hat{f}_{h,R_h} \leftarrow \operatorname*{argmin}_{f_h \in \mathcal{F}_h(R_h)} \mathcal{L}_{\mathcal{D}_h,R_h}(f_h, \hat{V}_{h+1}), \tag{B.1}$$

where $\mathcal{L}_{\mathcal{D}_h,R_h}(f_h, \hat{V}_{h+1}) := \sum_{i=1}^n \left(f_h(s_h^{(i)}, a_h^{(i)}) - R_h(s_h^{(i)}, a_h^{(i)}) - \hat{V}_{h+1}(s_{h+1}^{(i)})\right)^2$

Define $\hat{\pi}_h(s) = \mathrm{argmax}_a \hat{f}_{h,R_h}(s,a)$.

Define $\hat{V}_h(s) = \mathrm{clip}_{[0,H]}\left(\max_a \hat{f}_{h,R_h}(s,a)\right)$ for FQI-FULL-CLASS and FQI-REPRESENTATION, while define $\hat{V}_h(s) = \mathrm{clip}_{[0,1]}\left(\max_a \hat{f}_{h,R_h}(s,a)\right)$ for FQI-ELLIPTICAL.

return $\hat{\pi} = (\hat{\pi}_0, \ldots, \hat{\pi}_{H-1})$.

Unlike the regression problem in the main text, the objective here includes an additional reward function component. Therefore, we define a new loss function $\mathcal{L}_{D_h,R_h}$, and will use $\mathcal{L}_{\rho_{h-3}^{+3},R_h}$ to denote its population version. Notice that the function class $\mathcal{F}_h(R_h)$ in Algorithm B.1 also depends on the reward function R. If we pull out the reward term from $\hat{f}_{h,R_h}$, we can obtain an equivalent solution of the least squares problem (B.1) as below:

$$\hat{f}_{h,R_h} = R_h + \operatorname*{argmin}_{f_h \in \mathcal{F}_h(\mathbf{0})} \mathcal{L}_{\mathcal{D}_h}(f_h, \hat{V}_{h+1}),\ \mathcal{L}_{\mathcal{D}_h}(f_h, \hat{V}_{h+1}) := \sum_{i=1}^n \left(f_h(s_h^{(i)}, a_h^{(i)}) - \hat{V}_{h+1}(s_{h+1}^{(i)})\right)^2.$$

Intuitively, the reward function R_h only makes the current least squares solution offset the original (reward-independent) least squares solution by R_h.

B.1.2 Planning for a reward class with full representation class

In this part, we first establish the sample complexity of planning for a deterministic reward function R in Lemma B.1. We will choose FQI-FULL-CLASS as the planner, where the Q-value function class consists of linear function of all features in the feature class with reward appended. Specifically,

we have $\mathcal{F}_h(R_h) \coloneqq \{R_h + \langle \phi_h, w_h \rangle : \|w_h\|_2 \leq H\sqrt{d}, \phi_h \in \Phi_h\}, h \in [H]$. Equipped with this lemma, we also provide the sample complexity of planning for a finite deterministic reward class $\mathcal{R}$ in Corollary B.2.

Lemma B.1 (Planning for a known reward with full representation class)**.** *Assume that we have the exploratory dataset $\{\mathcal{D}\}_{0:H-1}$ (collected from ρ^{+3}_{h-3} and satisfies* (5.5) *for all $h \in [H]$). For a known deterministic reward function $R = R_{0:H-1} : \mathcal{S} \times \mathcal{A} \to [0,1]$ and $\delta \in (0,1)$, if we set*

$$n \geq \frac{512H^6d^2\kappa A}{\beta^2} \log\left(\frac{256H^6d^2\kappa A}{\beta^2}\right) + \frac{512H^6d^2\kappa A}{\beta^2} \log\left(\tfrac{2|\Phi|H}{\delta}\right),$$

then with probability at least $1-\delta$, the policy $\hat{\pi}$ returned by FQI-FULL-CLASS *satisfies*

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right] \geq \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right] - \beta.$$

Proof. For notation simplicity, we will drop the R_h subscript of $\hat{f}_{h,R_h}$ throughout the proof. Note that, this conflicts with the notation in later lemma statements (e.g. Lemma B.3 where we put back the subscript), and we reserve it for the reward-augmented function class for clarity in this proof. We first bound the difference in cumulative rewards between $\hat{\pi} := \hat{\pi}_{0:H-1}$ and the optimal policy π^* for the given reward function. Recall that $\hat{\pi}_0$ is greedy w.r.t. $\hat{f}_0$, which implies that $\hat{f}_0(x_0, \hat{\pi}_0(x_0)) \geq \hat{f}_0(x_0, \pi^*(x_0))$ for all x_0. Hence, we have

$$\begin{aligned}
V^* - V^{\hat{\pi}} &= \mathbb{E}_{\pi^*}\left[R_0(x_0,a_0) + V^*(x_1)\right] - \mathbb{E}_{\hat{\pi}}\left[R_0(x_0,a_0) + V^{\hat{\pi}}(x_1)\right] \\
&\leq \mathbb{E}_{\pi^*}\left[R_0(x_0,a_0) + V^*(x_1) - \hat{f}_0(x_0,a_0)\right] - \mathbb{E}_{\hat{\pi}}\left[R_0(x_0,a_0) + V^{\hat{\pi}}(x_1) - \hat{f}_0(x_0,a_0)\right] \\
&= \mathbb{E}_{\pi^*}\left[R_0(x_0,a_0) + V^*(x_1) - \hat{f}_0(x_0,a_0)\right] - \mathbb{E}_{\hat{\pi}}\left[R_0(x_0,a_0) + V^*(x_1) - \hat{f}_0(x_0,a_0)\right] \\
&\quad + \mathbb{E}_{\hat{\pi}}\left[V^*(x_1) - V^{\hat{\pi}}(x_1)\right] \\
&= \mathbb{E}_{\pi^*}\left[Q^*_0(x_0,a_0) - \hat{f}_0(x_0,a_0)\right] - \mathbb{E}_{\hat{\pi}}\left[Q^*_0(x_0,a_0) - \hat{f}_0(x_0,a_0)\right] \\
&\quad + \mathbb{E}_{\hat{\pi}}\left[V^*(x_1) - V^{\hat{\pi}}(x_1)\right].
\end{aligned}$$

Continuing unrolling to $h = H-1$, we get

$$V^* - V^{\hat{\pi}} \leq \sum_{h=0}^{H-1} \mathbb{E}_{\hat{\pi}_{0:h-1} \circ \pi^*}\left[Q^*_h(s_h,a_h) - \hat{f}_h(s_h,a_h)\right] - \sum_{h=0}^{H-1} \mathbb{E}_{\hat{\pi}_{0:h}}\left[Q^*_h(s_h,a_h) - \hat{f}_h(s_h,a_h)\right].$$

Now we bound each of these terms. The two terms only differ in the policies that generate the data and can be handled similarly. Therefore, in the following, we focus on just one of them.

For any function $V_{h+1} : \mathcal{S} \to \mathbb{R}$, we introduce Bellman backup operator $(\mathcal{T}_h V_{h+1})(s_h, a_h) := R_h(s_h, a_h) + \mathbb{E}\left[V_{h+1}(s_{h+1}) \mid s_h, a_h\right]$. Let's call the roll-in policy π and drop the dependence on h. This gives us

$$\begin{aligned}
&\left|\mathbb{E}_\pi\left[Q^*(s,a) - \hat{f}(s,a)\right]\right| = \left|\mathbb{E}_\pi\left[R(s,a) + \mathbb{E}\left[V^*(s') \mid s,a\right] - \hat{f}(s,a)\right]\right| \\
&\leq \mathbb{E}_\pi\left[\left|R(s,a) + \mathbb{E}\left[V^*(s') \mid s,a\right] - \hat{f}(s,a)\right|\right] \\
&\leq \mathbb{E}_\pi\left[\left|\mathbb{E}\left[V^*(s') \mid s,a\right] - \mathbb{E}\left[\hat{V}(s') \mid s,a\right]\right| + \left|R(s,a) + \mathbb{E}\left[\hat{V}(s') \mid s,a\right] - \hat{f}(s,a)\right|\right] \\
&\leq \mathbb{E}_\pi\left[\left|V^*(s') - \hat{V}(s')\right| + \left|(\mathcal{T}\hat{V})(s,a) - \hat{f}(s,a)\right|\right],
\end{aligned}$$

where the last inequality is Jensen's inequality.

From the definition of $\hat{V}(s')$, we have

$$\begin{aligned}
\mathbb{E}_\pi\left[\left|V^*(s') - \hat{V}(s')\right|\right] &\leq \mathbb{E}_\pi\left[\left|\max_a Q^*(s',a) - \max_{a'} \hat{f}(s',a')\right|\right] \\
&\leq \mathbb{E}_{\pi \circ \tilde{\pi}}\left[\left|Q^*(s',a') - \hat{f}(s',a')\right|\right].
\end{aligned}$$

In the last inequality, we define $\tilde{\pi}$ to be the greedy one between two actions, that is we set $\tilde{\pi}(s') = \mathrm{argmax}_{a'} \max\{Q^*(s',a'), \hat{f}(s',a')\}$. This expression has the same form as the initial one, while at the next timestep. Keep unrolling yields

$$\begin{aligned}
\mathbb{E}_\pi\left[Q^*(s_h,a_h) - \hat{f}_h(s_h,a_h)\right] &\leq \sum_{\tau=h}^{H-1} \max_{\pi_\tau} \mathbb{E}_{\pi_\tau}\left[\left|(\mathcal{T}_\tau \hat{V}_{\tau+1})(x_\tau,a_\tau) - \hat{f}_\tau(x_\tau,a_\tau)\right|\right] \\
&\leq \sum_{\tau=h}^{H-1} \max_{\pi_\tau} \sqrt{\mathbb{E}_{\pi_\tau}\left[\left[(\mathcal{T}_\tau \hat{V}_{\tau+1})(x_\tau,a_\tau) - \hat{f}_\tau(x_\tau,a_\tau)\right]^2\right]} \\
&\leq \sum_{\tau=h}^{H-1} \sqrt{\kappa A \mathbb{E}_{\rho_{\tau-3}^{+3}}\left[\left[(\mathcal{T}_\tau \hat{V}_{\tau+1})(x_\tau,a_\tau) - \hat{f}_\tau(x_\tau,a_\tau)\right]^2\right]}, \qquad \text{(B.2)}
\end{aligned}$$

where the last inequality is due to condition (5.5).

Further, we have that with probability at least $1-\delta$,

$$\begin{aligned}
&\mathbb{E}_{\rho_{\tau-3}^{+3}}\left[\left((\mathcal{T}_\tau \hat{V}_{\tau+1})(x_\tau,a_\tau)-\hat{f}_\tau(x_\tau,a_\tau)\right)^2\right] \\
&=\mathbb{E}_{\rho_{\tau-3}^{+3}}\Big[\left(R_\tau(x_\tau,a_\tau)+\hat{V}_{\tau+1}(x_{\tau+1})-\hat{f}_\tau(x_\tau,a_\tau)\right)^2 \\
&\qquad -\left(R_\tau(x_\tau,a_\tau)+\hat{V}_{\tau+1}(x_{\tau+1})-(\mathcal{T}_\tau\hat{V}_{\tau+1})(x_\tau,a_\tau)\right)^2\Big] \\
&=\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_\tau,R_\tau}(\hat{f}_\tau,\hat{V}_{\tau+1})\right]-\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_\tau,R_\tau}(\mathcal{T}_\tau\hat{V}_{\tau+1},\hat{V}_{\tau+1})\right] \\
&\le \frac{256H^2d^2\log(2n|\Phi|H/\delta)}{n}. \qquad \text{(Step (*), Lemma B.3)}
\end{aligned}$$

Plugging this back into the overall value performance difference, the bound is

$$V^*-V^{\hat{\pi}}\le H^2\sqrt{\kappa A}\sqrt{\frac{256H^2d^2\log(2n|\Phi|H/\delta)}{n}}.$$

Setting RHS to be less than β and reorganize, we get

$$n\ge\frac{256H^6d^2\kappa A\log(2n|\Phi|H/\delta)}{\beta^2}.$$

A sufficient condition for the inequality above is

$$n\ge\frac{512H^6d^2\kappa A}{\beta^2}\log\left(\frac{256H^6d^2\kappa A}{\beta^2}\right)+\frac{512H^6d^2\kappa A}{\beta^2}\log\left(\tfrac{2|\Phi|H}{\delta}\right),$$

which completes the proof. □

Corollary B.2 (Planning for a reward class with full representation class)**.** *Assume that we have the exploratory dataset $\{\mathcal{D}\}_{0:H-1}$ (collected from ρ_{h-3}^{+3} and satisfies* (5.5) *for all $h\in[H]$), and we are given a finite deterministic reward class $\mathcal{R}\subset(\mathcal{S}\times\mathcal{A}\to[0,1])$. For $\delta\in(0,1)$ and any reward function $R\in\mathcal{R}$, if we set*

$$n\ge\frac{512H^6d^2\kappa A}{\beta^2}\log\left(\frac{256H^6d^2\kappa A}{\beta^2}\right)+\frac{512H^6d^2\kappa A}{\beta^2}\log\left(\tfrac{2|\Phi||\mathcal{R}|H}{\delta}\right),$$

then with probability at least $1-\delta$, the policy $\hat{\pi}$ returned by FQI-FULL-CLASS *satisfies*

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=0}^{H-1}R_h(s_h,a_h)\right]\ge\max_\pi\mathbb{E}_\pi\left[\sum_{h=0}^{H-1}R_h(s_h,a_h)\right]-\beta.$$

Proof. For any fixed reward $R \in \mathcal{R}$, we apply Lemma B.1 and get that with probability $1-\delta'$,

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right] \geq \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right] - \beta,$$

if we set

$$n \geq \frac{512H^6d^2\kappa A}{\beta^2}\log\left(\frac{256H^6d^2\kappa A}{\beta^2}\right) + \frac{512H^6d^2\kappa A}{\beta^2}\log\left(\tfrac{2|\Phi|H}{\delta'}\right).$$

Union bounding over $R \in \mathcal{R}$ and setting $\delta = \delta'/|\mathcal{R}|$ gives us the desired result. □

Lemma B.3 (Deviation bound for Lemma B.1)**.** *Given a deterministic reward function $R(= R_{0:H-1} : \mathcal{S} \times \mathcal{A} \to [0,1])$ and a dataset $\{\mathcal{D}\}_{0:H-1}$ (collected from ρ^{+3}_{h-3}) or $\{\mathcal{D}^o\}_{0:H-1}$ (collected from ρ^{+2}_{h-2}), where $\mathcal{D}_h$ or $\mathcal{D}^o_h$ is $\{s_h^{(i)}, a_h^{(i)}, s_{h+1}^{(i)}\}_{i=1}^n$. With probability at least $1-\delta$, $\forall\, h \in [H], V_{h+1} \in \mathcal{V}_{h+1}(R_{h+1})$, we have*

$$\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\hat{f}_{h,R_h}, V_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\mathcal{T}_h V_{h+1}, V_{h+1})\right]\right| \leq \frac{256H^2d^2\log(2n|\Phi|H/\delta)}{n},$$

or

$$\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}^o_h,R_h}(\hat{f}_{h,R_h}, V_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}^o_h,R_h}(\mathcal{T}_h V_{h+1}, V_{h+1})\right]\right| \leq \frac{256H^2d^2\log(2n|\Phi|H/\delta)}{n}.$$

Here, $\mathcal{V}_{h+1}(R_{h+1}) := \{\mathrm{clip}_{[0,H]}(\max_a f_{h+1}(s_{h+1}, a)) : f_{h+1} \in \mathcal{F}_{h+1}(R_{h+1})\}$ for $h \in [H-1]$ and $\mathcal{V}_H = \{\mathbf{0}\}$ is the clipped state-value function class, and $\mathcal{F}_h(R_h) := \{R_h + \langle \phi_h, w_h\rangle : \|w_h\|_2 \leq H\sqrt{d}, \phi_h \in \Phi_h\}$ for $h \in [H-1]$ is the Q-value function class.

Recall the definition, we get $\hat{f}_{h,R_h} = R_h + \hat{f}_h$, where $\hat{f}_h = \mathrm{argmin}_{f_h \in \mathcal{F}_h(\mathbf{0})} \mathcal{L}_{\mathcal{D}_h}(f_h, V_{h+1})$, $\mathcal{F}_h(\mathbf{0}) := \{\langle \phi_h, w_h\rangle : \|w_h\|_2 \leq H\sqrt{d}, \phi_h \in \Phi_h\}$, and $\mathcal{L}_{\mathcal{D}_h}(f_h, V_{h+1}) := \sum_{i=1}^n (f_h(s_h^{(i)}, a_h^{(i)}) - V_{h+1}(s_{h+1}^{(i)}))^2$. Therefore we have the following: $\mathcal{L}_{\mathcal{D}_h,R_h}(\hat{f}_{h,R_h}, V_{h+1}) = \mathcal{L}_{\mathcal{D}_h}(\hat{f}_h, V_{h+1})$ and $\mathcal{L}_{\mathcal{D}_h,R_h}(\mathcal{T}_h V_{h+1}, V_{h+1}) = \mathcal{L}_{\mathcal{D}_h}(\mathcal{T}_h V_{h+1} - R_h, V_{h+1}) = \mathcal{L}_{\mathcal{D}_h}(\langle \phi^*_h, \theta^*_{V_{h+1}}\rangle, V_{h+1})$. We can similarly define the notation for $\mathcal{D}^o_h$.

Proof. We only present the proof for $\mathcal{D}_h$ and the proof for $\mathcal{D}^o_h$ follows the same steps by changing ρ^{+3}_{h-3} and $\mathcal{D}_h$ to ρ^{+2}_{h-2} and $\mathcal{D}^o_h$ respectively.

Firstly, we fix $h \in [H]$. Noticing the structure of $\mathcal{F}_h(\mathbf{0})$, we can associate any $f_h \in \mathcal{F}_h(\mathbf{0})$ with $\phi_h \in \Phi_h$ and w_h that satisfies $\|w_h\|_2 \leq H\sqrt{d}$. Therefore, we can equivalently write $\mathcal{L}_{D_h}(f_h, V_{h+1})$ as $\mathcal{L}_{D_h}(\phi_h, w_h, V_{h+1})$. Also noticing the structure of $\mathcal{V}_{h+1}(R_{h+1})$, we can directly apply Lemma B.8 with $\rho = \rho^{+3}_{h-3}$, $\Phi' = \Phi_h$, $\Phi'' = \Phi_{h+1}$, $B = H\sqrt{d}$, $L = H$, $\mathcal{R} = \{R\}$, and π' to be the greedy policy $\pi'(s_{h+1}) = \mathrm{argmax}_{a'}(R(s_{h+1}, a') + \langle \phi'(s_{h+1}, a'), \theta\rangle)$.

This implies that for all $\|w_h\|_2 \le H\sqrt{d}$, $\phi_h \in \Phi_h$, and $V_{h+1} \in \mathcal{V}_{h+1}(R_{h+1})$, with probability at least $1-\delta'$, we have

$$\begin{aligned}&\left|\mathcal{L}_{\rho^{+3}_{h-3}}(\phi_h, w_h, V_{h+1}) - \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right| \\ \le\ & 2\left(\mathcal{L}_{\mathcal{D}_h}(\phi_h, w_h, V_{h+1}) - \mathcal{L}_{\mathcal{D}_h}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right) + \frac{128H^2 d\log(2/\delta')}{n}.\end{aligned}$$

From the definition, we have $\hat{f}_h = \operatorname{argmin}_{f_h \in \mathcal{F}_h(\mathbf{0})} \mathcal{L}_{\mathcal{D}_h}(f_h, V_{h+1})$. Noticing the structure of $\mathcal{F}_h(\mathbf{0})$, we can write $\hat{f}_h = \langle \hat{\phi}_h, \hat{w}_h\rangle$, where $\hat{\phi}_h, \hat{w}_h = \operatorname{argmin}_{\phi_h \in \Phi_h, \|w_h\|_2 \le H\sqrt{d}} \mathcal{L}_{\mathcal{D}_h}(\phi_h, w_h, V_{h+1})$ (here we abuse the notation of $\hat{\phi}_h$, which is reserved for the learnt feature).

Since $\phi^*_h \in \Phi_h$ and $\|\theta^*_{V_{h+1}}\|_2 \le H\sqrt{d}$ from Lemma 5.1, we get

$$\left|\mathcal{L}_{\rho^{+3}_{h-3}}(\hat{\phi}_h, \hat{w}_h, V_{h+1}) - \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right| \le \frac{128H^2 d\log(2/\delta')}{n}.$$

Perform the same union bound as Lemma B.8, and additionally union bound over $h \in [H]$, and set $\delta = \delta' / \left(|\Phi|^2 |\overline{\mathcal{W}}_h| |\Theta_h| H\right)$, with probability at least $1-\delta$, we have

$$\left|\mathcal{L}_{\rho^{+3}_{h-3}}(\hat{\phi}_h, \hat{w}_h, V_{h+1}) - \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right| \le \frac{128H^2 d\log(2|\Phi|^2 H(2n)^{2d}/\delta)}{n}. \tag{B.3}$$

Finally, relaxing the rhs in the inequality above and noticing by definition $\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h}(\hat{f}_h, V_{h+1})\right] = \mathcal{L}_{\rho^{+3}_{h-3}}(\hat{\phi}_h, \hat{w}_h, V_{h+1})$ and $\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h, R_h}(\mathcal{T}_h V_{h+1}, V_{h+1})\right] = \mathcal{L}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})$, we complete the proof. □

B.1.3 Elliptical planner

In this part, we show the sample complexity of elliptical planner (FQI-ELLIPTICAL), which is specialized in planning for the elliptical reward class defined in Lemma B.4. The Q-value function class consists of linear function of all features in the feature class with reward appended. Specifically, we have $\mathcal{F}_h(R_h) := \{R_h + \langle \phi_h, w_h\rangle : \|w_h\|_2 \le \sqrt{d}, \phi_h \in \Phi_h\}, h \in [H]$. We still use the full representation class, but with a different bound on the norm of the parameters when compared with FQI-FULL-CLASS.

Lemma B.4 (Elliptical planner)**.** *Assume that we have the exploratory dataset $\{\mathcal{D}\}_{0:H-1}$ (collected from ρ^{+3}_{h-3} and satisfies* (5.5) *for all $h \in [H]$). For $\delta \in (0,1)$ and any deterministic elliptical reward function $R \in \mathcal{R}$, where $\mathcal{R} := \{R_{0:H-1} : R_{0:H-2} = \mathbf{0}, R_{H-1} \in \{\phi_{H-1}^\top \Gamma^{-1} \phi_{H-1} : \phi_{H-1} \in$*

$\Phi_{H-1}, \Gamma \in \mathbb{R}^{d\times d}, \lambda_{\min}(\Gamma) \geq 1\}\}$*, if we set*

$$n \geq \frac{584H^4d^3\kappa A}{\beta^2}\log\left(\frac{292H^4d^3\kappa A}{\beta^2}\right) + \frac{584H^4d^3\kappa A}{\beta^2}\log\left(\tfrac{2|\Phi|H}{\delta}\right),$$

then with probability at least $1-\delta$*, the policy* $\hat{\pi}$ *returned by* FQI-ELLIPTICAL *satisfies*

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right] \geq \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{h=0}^{H-1} R_h(s_h, a_h)\right] - \beta.$$

Remark: notice that the elliptical reward function only has a non-zero value at timestep $H-1$.

Proof. The proof follows the same steps in Lemma B.1. Since we consider a deterministic elliptical reward function class here, we apply Lemma B.5 instead of Lemma B.3 in Step (*). Then following a similar calculation gives us the result immediately. □

Lemma B.5 (Deviation bound for Lemma B.4)**.** *Consider the deterministic elliptical reward function classes* $\mathcal{R} \coloneqq \{R_{0:H-1} : R_{0:H-2} = \mathbf{0}, R_{H-1} \in \mathcal{R}_{H-1} \coloneqq \{\phi_{H-1}^\top \Gamma^{-1}\phi_{H-1} : \phi_{H-1} \in \Phi_{H-1}, \Gamma \in \mathbb{R}^{d\times d}, \lambda_{\min}(\Gamma) \geq 1\}\}$*, an exploratory dataset* $\mathcal{D}_h \coloneqq \{(s_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)})\}_{i=1}^n$ *collected from* ρ_{h-3}^{+3}*,* $h \in [H]$*. With probability at least* $1-\delta$*,* $\forall\, R \in \mathcal{R}, h \in [H-1], V_{h+1} \in \mathcal{V}_{h+1}(R_{h+1})$*, we have*

$$\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h, R_h}(\hat{f}_{h,R_h}, V_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h, R_h}(\mathcal{T}_h V_{h+1}, V_{h+1})\right]\right| \leq \frac{292d^3\log(2n|\Phi|H/\delta)}{n}.$$

Here, $\mathcal{V}_{h+1}(R_{h+1}) \coloneqq \{\mathrm{clip}_{[0,1]}(\max_a f_{h+1}(s_{h+1}, a)) : f_{h+1} \in \mathcal{F}_{h+1}(R_{h+1})\}$ *for* $h \in [H-1]$ *and* $\mathcal{V}_H = \{\mathbf{0}\}$ *is the clipped state-value function class, and* $\mathcal{F}_h(R_h) \coloneqq \{R_h + \langle \phi_h, w_h\rangle : \|w_h\|_2 \leq \sqrt{d}, \phi_h \in \Phi_h\}$ *for* $h \in [H-1]$ *is the reward dependent Q-value function class.*

Proof. First, from Lemma B.10, we know that there exists a γ-cover $\mathcal{C}_{\mathcal{R}_{H-1}}$ for the reward class $\mathcal{R}_{H-1}$. From the definition of $\mathcal{R}$, we know that $\mathcal{C}_{\mathcal{R}} \coloneqq \{R_{0:H-1} : R_{0:H-2} = \mathbf{0}, R_{H-1} \in \mathcal{C}_{\mathcal{R}_{H-1}}\}$ is a γ-cover of reward class $\mathcal{R}$ and $|\mathcal{C}_{\mathcal{R}}| = |\mathcal{C}_{\mathcal{R}_{H-1}}|$.

For any fixed $\tilde{R} \in \mathcal{C}_{\mathcal{R}}$, we can follow the same steps in Lemma B.3 to get a concentration result like (B.3). The only differences are that the norm of w_h is now bounded by $\sqrt{d}$ instead of $H\sqrt{d}$ and we clip to $[0,1]$. Therefore, for this fixed $\tilde{R}$, with probability at least $1-\delta'$, we have that $\forall h \in [H-1], \tilde{V}_{h+1} \in \mathcal{V}_{h+1}(\tilde{R}_{h+1})$,

$$\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h, \tilde{R}_h}(\hat{f}_{h,\tilde{R}_h}, \tilde{V}_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h, \tilde{R}_h}(\mathcal{T}_h V_{h+1}, V_{h+1})\right]\right| \leq \frac{128\log(|\Phi|^2|(2n)^{2d}H/\delta')}{n}.$$

Union bounding over all $\tilde{R} \in \mathcal{C}_{\mathcal{R}}$ and set $\delta = \delta'/|\mathcal{C}_{\mathcal{R}}|$, with probability at least $1-\delta$, we have

$$\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,\tilde{R}_h}(\hat{f}_{h,\tilde{R}_h}, V_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,\tilde{R}_h}(\mathcal{T}_h\tilde{V}_{h+1}, \tilde{V}_{h+1})\right]\right| \leq \frac{128d\log(|\Phi|^2|(2n)^{2d}|\mathcal{C}_{\mathcal{R}}|H/\delta)}{n}.$$

Notice that $\mathcal{C}_{\mathcal{R}}$ is a γ-cover of $\mathcal{R}$, for any $R \in \mathcal{R}$, there exists $\tilde{R} \in \mathcal{C}_{\mathcal{R}}$, so that $\|R_h - \tilde{R}_h\|_\infty \leq \gamma$. Therefore $\forall f_h \in \mathcal{F}_h(R_h)$ and $V_h \in \mathcal{V}_h(R_h)$, there exists some $\tilde{f}_h \in \mathcal{F}(\tilde{R}_h)$ and $\tilde{V}_h \in \mathcal{V}(\tilde{R}_h)$ that satisfies $\|\tilde{f}_h - f_h\|_\infty \leq \gamma$ and $\|\tilde{V}_h - V_h\|_\infty \leq \gamma$. Hence, for any $R \in \mathcal{R}$, with probability at least $1-\delta$,

$$\begin{aligned}
&\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\hat{f}_{h,R_h}, V_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\mathcal{T}_hV_{h+1}, V_{h+1})\right]\right| \\
\leq &\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,\tilde{R}_h}(\hat{f}_{h,\tilde{R}_h}, \tilde{V}_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,\tilde{R}_h}(\mathcal{T}_h\tilde{V}_{h+1}, \tilde{V}_{h+1})\right]\right| + 36\sqrt{d}\gamma \\
\leq &\frac{128d\log(|\Phi|^2|(2n)^{2d}|\mathcal{C}_{\mathcal{R}}|H/\delta)}{n} + 36\sqrt{d}\gamma \\
\leq &\frac{292d^3\log(2n|\Phi|H/\delta)}{n}.
\end{aligned}$$

The last inequality is obtained by choosing $\gamma = \frac{\sqrt{d}}{n}$ and noticing $|\mathcal{C}_{\mathcal{R}}| = |\Phi_{H-1}|(2\sqrt{d}/\gamma)^{d^2} \leq |\Phi|(2n)^{d^2}$. This completes the proof. □

B.1.4 Planning for a reward class with learnt representation function

In this part, we will show that the learnt feature $\bar{\phi}$ enables the downstream policy optimization for a finite deterministic reward class $\mathcal{R}$. The sample complexity is shown in Lemma B.6. We will choose FQI-REPRESENTATION as the planner, where the Q-value function class only consists of linear function of learnt feature $\bar{\phi}$ with reward appended. Specifically, we have $\mathcal{F}_h(R_h) \coloneqq \{R_h + \mathrm{clip}_{[0,H]}\left(\langle\bar{\phi}_h, w_h\rangle\right) : \|w_h\|_2 \leq B\}, h \in [H]$. In addition to constructing the function class with learnt feature itself, we also perform clipping in $\mathcal{F}_h(R_h)$. This clipping variant helps us avoid the poly(B) dependence in the sample complexity bound. Notice that clipped Q-value function classes also work for FQI-FULL-CLASS and FQI-ELLIPTICAL, and would save d factor. We only introduce this variant here because B is much larger than $H\sqrt{d}$ or d.

Lemma B.6 (Planning for a reward class with a learnt representation function)**.** *Assume that we have the exploratory dataset $\{\mathcal{D}\}_{0:H-1}$ (collected from ρ^{+3}_{h-3} and satisfies* (5.5) *for all $h \in [H]$), a learned feature $\bar{\phi}_h$ that satisfies the condition in* (5.6)*, and a finite deterministic reward class $\mathcal{R} \subset (\mathcal{S}\times\mathcal{A} \to [0,1])$. For $\delta \in (0,1)$ and any reward function $R \in \mathcal{R}$, if we set*

$$n \geq \frac{200H^6d\kappa A}{\beta^2}\log\left(\frac{100H^6d\kappa A}{\beta^2}\right) + \frac{200H^6d\kappa A}{\beta^2}\log\left(\tfrac{2|\mathcal{R}|B}{\delta}\right),$$

then with probability at least $1-\delta$*, the policy* $\hat{\pi}$ *returned by* FQI-REPRESENTATION *satisfies*

$$\mathbb{E}_{\hat{\pi}}\left[\sum_{h=0}^{H-1} R_h(s_h,a_h)\right] \geq \max_{\pi} \mathbb{E}_{\pi}\left[\sum_{h=0}^{H-1} R_h(s_h,a_h)\right] - \beta - 2H^2\sqrt{\kappa A \varepsilon_{\text{apx}}}.$$

Proof. Following similar steps in the proof of Lemma B.1 and replacing Lemma B.3 with Lemma B.7 in Step(*), with probability at least $1-\delta$, we have

$$\begin{aligned} V^* - V^{\hat{\pi}} \leq & H^2\sqrt{\kappa A}\sqrt{\frac{100H^2 d\log(2n|\mathcal{R}|B/\delta)}{n} + 3\varepsilon_{\text{apx}}} \\ \leq & H^2\sqrt{\kappa A}\sqrt{\frac{100H^2 d\log(2n|\mathcal{R}|B/\delta)}{n}} + 2H^2\sqrt{\kappa A\varepsilon_{\text{apx}}}. \end{aligned}$$

Setting rhs to be less than $\beta + 2H^2\sqrt{\kappa A\varepsilon_{\text{apx}}}$ and reorganize, we get the condition

$$n \geq \frac{100H^6 d\kappa A\log(2n|\mathcal{R}|B/\delta)}{\beta^2}.$$

A sufficient condition for the inequality above is

$$n \geq \frac{200H^6 d\kappa A}{\beta^2}\log\left(\frac{100H^6 d\kappa A}{\beta^2}\right) + \frac{200H^6 d\kappa A}{\beta^2}\log\left(\tfrac{2|\mathcal{R}|B}{\delta}\right),$$

which completes the proof. □

Lemma B.7 (Deviation bound for Lemma B.6)**.** *Assume that we have an exploratory dataset* $\mathcal{D}_h := \left\{(s_h^{(i)}, a_h^{(i)}, x_{h+1}^{(i)})\right\}_{i=1}^n$ *collected from* ρ_{h-3}^{+3}*,* $h \in [H]$*, a learned feature* $\bar{\phi}_h$ *that satisfies the condition in* (5.6)*, and a finite deterministic reward class* $\mathcal{R} \subset (\mathcal{S}\times\mathcal{A}\to[0,1])$*. Then, with probability at least* $1-\delta$*,* $\forall R\in\mathcal{R}, h\in[H], V_{h+1}\in\mathcal{V}_{h+1}(R_{h+1})$*, we have*

$$\left|\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\hat{f}_{h,R_h}, V_{h+1})\right] - \mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\mathcal{T}_h V_{h+1}, V_{h+1})\right]\right| \leq \frac{100H^2 d\log(2n|\mathcal{R}|B/\delta)}{n} + 3\varepsilon_{\text{apx}}.$$

Here $\mathcal{V}_{h+1}(R_{h+1}) := \{\text{clip}_{[0,H]}(\max_a f_{h+1}(s_{h+1},a)) : f_{h+1}\in\mathcal{F}_{h+1}(R_{h+1})\}$ *for* $h\in[H-1]$*, and* $\mathcal{V}_H = \{\mathbf{0}\}$ *is the clipped state-value function class and* $\mathcal{F}_h(R_h) := \{R_h + \text{clip}_{[0,H]}\left(\langle\bar{\phi}_h, w_h\rangle\right) : \|w_h\|_2 \leq B\}$ *for* $h\in[H-1]$ *is the reward dependent Q-value function class.*

Recall the definition, we have $\hat{f}_{h,R_h} = R_h + \hat{f}_h$, where $\hat{f}_h = \text{argmin}_{f_h\in\mathcal{F}_h(\mathbf{0})}\mathcal{L}_{\mathcal{D}_h}(f_h, V_{h+1})$, $\mathcal{L}_{\mathcal{D}_h}(f_h,V_{h+1}) := \sum_{i=1}^n\left(f_h(s_h^{(i)},a_h^{(i)}) - V_{h+1}(s_{h+1}^{(i)})\right)^2$, and $\mathcal{F}_h(\mathbf{0}) := \{\text{clip}_{[0,H]}\left(\langle\bar{\phi}_h, w_h\rangle\right) : \|w_h\|_2 \leq B\}$. Therefore we get the following: $\mathcal{L}_{\mathcal{D}_h,R_h}(\hat{f}_{h,R_h},V_{h+1}) = \mathcal{L}_{\mathcal{D}_h}(\hat{f}_h,V_{h+1})$ and $\mathcal{L}_{\mathcal{D}_h,R_h}(\mathcal{T}_h V_{h+1},V_{h+1}) = \mathcal{L}_{\mathcal{D}_h}(\mathcal{T}_h V_{h+1}-R_h,V_{h+1}) = \mathcal{L}_{\mathcal{D}_h}(\langle\phi_h^*,\theta_{V_{h+1}}^*\rangle, V_{h+1})$.

Proof. First, we show that condition (5.6) implies following condition for all $h \in [H]$: $V_{h+1} \in \{\mathrm{clip}_{[0,H]}\left(\max_a(R_{h+1}(s_{h+1},a) + \mathrm{clip}_{[0,H]}\left(\langle\phi_{h+1}(s_{h+1},a),\theta\rangle\right))\right) : \phi_{h+1} \in \Phi_{h+1}, \|\theta\|_2 \le B, R \in \mathcal{R}\}$. Let $\bar{w}_{V_{h+1}} = \mathrm{argmin}_{\|w_h\|_2\le B}\mathcal{L}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1})$ with $B \ge H\sqrt{d}$, we have

$$\mathbb{E}_{\rho_{h-3}^{+3}}\left[\left(\left\langle\bar{\phi}_h(s_h,a_h), \bar{w}_{V_{h+1}}\right\rangle - \mathbb{E}\left[V_{h+1}(s_{h+1})|s_h,a_h\right]\right)^2\right] \le \varepsilon_{\mathrm{apx}}.$$

This is due to the order of taking max and clipping doesn't matter:

$$\begin{aligned}
&\mathrm{clip}_{[0,H]}\left(\max_a\left(R_{h+1}(s_{h+1},a) + \mathrm{clip}_{[0,H]}\left(\langle\phi_{h+1}(s_{h+1},a),\theta\rangle\right)\right)\right)\\
=&\mathrm{clip}_{[0,H]}\left(\max_a\left(R_{h+1}(s_{h+1},a) + \langle\phi_{h+1}(s_{h+1},a),\theta\rangle\right)\right).
\end{aligned}$$

Then we follow the similar structure as Lemma B.8. We fix $R \in \mathcal{R}, h \in [H], \|w_h\|_2 \le B$, and $V_{h+1} \in \mathcal{V}_{h+1}(R_{h+1})$ (equivalently w_{h+1}). We also fix $\tilde{\theta}_{V_{h+1}}$, where $\tilde{\theta}_{V_{h+1}}$ satisfies $\|\tilde{\theta}_{V_{h+1}}\|_2 \le B$ and for all (s_h,a_h), $\left|\left\langle\phi_h^*(s_h,a_h), \tilde{\theta}_{V_{h+1}}\right\rangle - \mathbb{E}[V(s_{h+1})|s_h,a_h]\right| = \left|\left\langle\phi_h^*(s_h,a_h), \tilde{\theta}_{V_{h+1}} - \theta^*_{V_{h+1}}\right\rangle\right| \le \gamma$.

Then we show a high probability bound on the following deviation term: $|\mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\phi_h^*, \tilde{\theta}_{V_{h+1}}, V_{h+1}) - (\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\phi_h^*, \tilde{\theta}_{V_{h+1}}, V_{h+1}))|$.

Since we apply clipping here, we define $\mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\phi_h, w_h, V_{h+1}) \coloneqq \mathbb{E}_{\rho_{h-3}^{+3}}[(\mathrm{clip}_{[0,H]}\left(\langle\phi_h, w_h\rangle\right) - V_{h+1})^2]$ and $\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}$ as its empirical version.

Let $g_h(s_h,a_h) = \mathrm{clip}_{[0,H]}\left(\left\langle\bar{\phi}_h(s_h,a_h), w_h\right\rangle\right)$ and $\tilde{g}_h(s_h,a_h) = \mathrm{clip}_{[0,H]}\left(\left\langle\phi_h^*(s_h,a_h), \tilde{\theta}_{V_{h+1}}\right\rangle\right)$. For random variable variable $Y \coloneqq \left(g_h(s_h,a_h) - V_{h+1}(s_{h+1})\right)^2 - \left(\tilde{g}_h(s_h,a_h) - V_{h+1}(s_{h+1})\right)^2$, we have:

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}\left[\left(g_h(s_h,a_h) - V_{h+1}(s_{h+1})\right)^2 - \left(\tilde{g}_h(s_h,a_h) - V_{h+1}(s_{h+1})\right)^2\right]\\
&= \mathbb{E}\left[\left(g_h(s_h,a_h) + \tilde{g}_h(s_h,a_h) - 2V_{h+1}(s_{h+1})\right)\left(g_h(s_h,a_h) - \tilde{g}_h(s_h,a_h)\right)\right]\\
&= \mathbb{E}\left[\left(g_h(s_h,a_h) - \tilde{g}_h(s_h,a_h)\right)^2\right]\\
&\quad - 2\mathbb{E}\left[\left(\mathbb{E}[V_{h+1}(s_{h+1})|s_h,a_h)] - \tilde{g}_h(s_h,a_h)\right)\left(g_h(s_h,a_h) - \tilde{g}_h(s_h,a_h)\right)\right].
\end{aligned}$$

Here the expectation is taken over ρ_{h-3}^{+3}.

Noticing the approximation assumption of $\tilde{\theta}_{V_{h+1}}$, and $g_h(s_h,a_h)$ and $\tilde{g}_h(s_h,a_h)$ are bounded by $[0,H]$, we have

$$\begin{aligned}
\left|\mathbb{E}[Y] - \mathbb{E}\left[\left(g_h(s_h,a_h) - \tilde{g}_h(s_h,a_h)\right)^2\right]\right| &\le 2\left|\left[\mathbb{E}[V_{h+1}(s_{h+1})|s_h,a_h)] - \tilde{g}_h(s_h,a_h)\right]\right|\left\|g_h - \tilde{g}_h\right\|_\infty\\
&\le 2H\left\|\left\langle\phi_h^*, \theta^*_{V_{h+1}}\right\rangle - \left\langle\phi_h^*, \tilde{\theta}_{V_{h+1}}\right\rangle\right\|_\infty \le 2H\gamma.
\end{aligned}$$

Next, for the variance of the random variable, notice that $g_h(s_h,a_h)$ and $\tilde{g}_h(s_h,a_h)$ are bounded by

$[0, H]$, we have:

$$\begin{aligned}\mathbb{V}[Y] &\le \mathbb{E}\left[Y^2\right] = \mathbb{E}\left[\left(g_h(s_h,a_h) + \tilde{g}_h(s_h,a_h) - 2V_{h+1}(s_{h+1})\right)^2 \left(g_h(s_h,a_h) - \tilde{g}_h(s_h,a_h)\right)^2\right] \\ &\le 4H^2\mathbb{E}\left[\left(g_h(s_h,a_h) - \tilde{g}_h(s_h,a_h)\right)^2\right] \\ &\le 4H^2\left(\mathbb{E}\left[Y\right] + 2H\gamma\right).\end{aligned}$$

Noticing $Y \in [-4H^2, 4H^2]$ and applying Bernstein's inequality, with probability at least $1-\delta'$, we can bound the deviation term above as:

$$\begin{aligned}&\Big|\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1}) \\ &\quad - \Big(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\Big)\Big| \\ &\le \sqrt{\frac{2\mathbb{V}[Y]\log(2/\delta')}{n}} + \frac{4H^2\log(2/\delta')}{3n} \\ &\le \sqrt{\frac{(8H^2\mathbb{E}[Y] + 16H^3\gamma)\log(2/\delta')}{n}} + \frac{4H^2\log(2/\delta')}{3n} \\ &\le \sqrt{\frac{8H^2\mathbb{E}[Y]\log(2/\delta')}{n}} + \sqrt{\frac{16H^3\gamma\log(2/\delta')}{n}} + \frac{4H^2\log(2/\delta')}{3n} \\ &\le \frac{1}{2}\mathbb{E}[Y] + \frac{4H^2\log(2/\delta')}{n} + \frac{4H^2\log(2/\delta')}{n} + \frac{4H^2\log(2/\delta')}{3n},\end{aligned}$$

where we set $\gamma = \frac{H}{n}$.

Substituting the definition of Y into this equation and reorganize, we obtain

$$\begin{aligned}&\Big|\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1}) \\ &\quad - \Big(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\Big)\Big| \\ &\le \frac{1}{2}\Big(\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\Big) + \frac{10H^2\log(2/\delta')}{n}.\end{aligned}$$

Further, consider a finite point-wise cover of the function class $\mathcal{G}_h := \{g_h(s_h,a_h) = \langle\bar{\phi}_h(s_h,a_h), w_h\rangle : \|w_h\|_2 \le B\}$. Note that, with a ℓ_2-cover $\overline{\mathcal{W}}_h$ of $\mathcal{W}_h = \{\|w_h\|_2 \le B\}$ at scale γ, we have for all (s_h,a_h), there exists $\tilde{w}_h \in \overline{\mathcal{W}}_h$, $\|\langle\bar{\phi}_h, w_h - \tilde{w}_h\rangle\|_\infty \le \gamma$. Similarly, for any $V_{h+1} \in \mathcal{V}_{h+1}(R_{h+1})$, we know that $\mathbb{E}[V_{h+1}(s_{h+1})|s_h,a_h] = \left\langle\phi^*_h(s_h,a_h), \theta^*_{V_{h+1}}\right\rangle$ for some $\|\theta^*_{V_{h+1}}\|_2 \le B$. Thus, we choose $\overline{\mathcal{W}}_{h+1}$ as an ℓ_2-cover of the set $\{w_{h+1} \in \mathbb{R}^d : \|w_{h+1}\|_2 \le B\}$ at scale γ. For all $V_{h+1}(\cdot) = \mathrm{clip}_{[0,H]}\big(\max_a\left(R_{h+1}(\cdot,a) + \left\langle\bar{\phi}_{h+1}(\cdot,a), w_{h+1}\right\rangle\right)\big) \in \mathcal{V}_{h+1}(R_{h+1})$, there exists $\tilde{w}_{h+1} \in \overline{\mathcal{W}}_{h+1}$ such that $\|\tilde{w}_{h+1} - w_{h+1}\|_2 \le \gamma$. This implies that for $\tilde{V}_{h+1}(\cdot) = \mathrm{clip}_{[0,H]}\big(\max_a\left(R_{h+1}(\cdot,a) + \left\langle\bar{\phi}_{h+1}(\cdot,a), \tilde{w}_{h+1}\right\rangle\right)\big)$, we have $\|V_{h+1} - \tilde{V}_{h+1}\|_\infty \le \gamma$. Therefore, for $\tilde{\theta}_{V_{h+1}} = \theta^*_{\tilde{V}_{h+1}}$, we have $\left|\left\langle\phi^*_h(s_h,a_h), \tilde{\theta}_{V_{h+1}}\right\rangle - \mathbb{E}[V(s_{h+1})|s_h,a_h]\right| = \left|\mathbb{E}[\tilde{V}(s_{h+1})|s_h,a_h] -\right.$

$\mathbb{E}[V(s_{h+1})|s_h, a_h]\big| \le \gamma$. Via standard argument, we can set $|\overline{\mathcal{W}}_{h+1}| = \left(\frac{2B}{\gamma}\right)^d$ and $|\overline{\mathcal{W}}_h| = \left(\frac{2B}{\gamma}\right)^d$, which implies $|\overline{\mathcal{W}}_{h+1}| \le \left(\frac{2nB}{H}\right)^d$ and $|\overline{\mathcal{W}}_h| \le \left(\frac{2nB}{H}\right)^d$.

Thus, applying a union bound over elements in $\overline{\mathcal{W}}_h$, $\overline{\mathcal{W}}_{h+1}$, with probability $1 - |\overline{\mathcal{W}}_h||\overline{\mathcal{W}}_{h+1}|\delta'$, for all $w_h \in \overline{\mathcal{W}}_h$ and $V_{h+1} \in \mathcal{V}_{h+1}(R_{h+1})$, we have:

$$
\begin{aligned}
&\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1}) \\
\le\ & \left|\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right| \\
\le\ & \left|\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \tilde{w}_h, V_{h+1})\right| \\
&\quad + \left|\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \tilde{w}_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\right| \\
&\quad + \left|\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right| \\
\le\ & 2H\gamma + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\right) + \frac{10H^2\log(2/\delta')}{n} + 2H\gamma \\
\le\ & 4H\gamma + \frac{10H^2\log(2/\delta')}{n} + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\right) + 4H\gamma \\
=\ & 8H\gamma + \frac{10H^2\log(2/\delta')}{n} + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1})\right) \\
&\quad + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\right) \\
\le\ & 8H\gamma + \frac{10H^2\log(2/\delta')}{n} + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1})\right) \\
&\quad + 3\left(\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\right) + \frac{20H^2\log(2/\delta')}{n} \\
\le\ & 8H\gamma + \frac{30H^2\log(2/\delta')}{n} + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1})\right) \\
&\quad + 3\left(\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \bar{w}_{V_{h+1}}, V_{h+1})\right) \\
&\quad + 3\left(\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \bar{w}_{V_{h+1}}, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \tilde{\theta}_{V_{h+1}}, V_{h+1})\right) \\
\le\ & 8H\gamma + \frac{30H^2\log(2/\delta')}{n} + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1})\right) \\
&\quad + 6H\gamma + 3\left(\mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\bar{\phi}_h, \bar{w}_{V_{h+1}}, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\rho^{+3}_{h-3}}(\phi^*_h, \theta^*_{V_{h+1}}, V_{h+1})\right) + 6H\gamma \\
\le\ & 20H\gamma + \frac{30H^2\log(2/\delta')}{n} + 2\left(\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, w_h, V_{h+1}) - \mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h, \tilde{w}_{V_{h+1}}, V_{h+1})\right) + 3\varepsilon_{\mathrm{apx}}.
\end{aligned}
$$

In the second last inequality, the construction of $\overline{\mathcal{W}}_h$ tells us that there exists $\tilde{w}_{V_{h+1}} \in \overline{\mathcal{W}}_h$ satisfies

$\|\tilde{w}_{V_{h+1}} - \bar{w}_{V_{h+1}}\|_2 \leq \gamma$. In the last inequality, we notice that

$$\begin{aligned}
&\mathbb{E}_{\rho_{h-3}^{+3}}\left[\left(\mathrm{clip}_{[0,H]}\left(\left\langle\bar{\phi}_h(s_h,a_h),\bar{w}_{V_{h+1}}\right\rangle\right)-\mathbb{E}\left[V_{h+1}(s_{h+1})|s_h,a_h\right]\right)^2\right]\\
\leq\ &\mathbb{E}_{\rho_{h-3}^{+3}}\left[\left(\left\langle\bar{\phi}_h(s_h,a_h),\bar{w}_{V_{h+1}}\right\rangle-\mathbb{E}\left[V_{h+1}(s_{h+1})|s_h,a_h\right]\right)^2\right]\leq\varepsilon_{\mathrm{apx}}
\end{aligned}$$

From the definition, we have $\hat{f}_h = \mathrm{argmin}_{f_h\in\mathcal{F}_h(\mathbf{0})}\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(f_h,V_{h+1})$. Noticing the structure of $\mathcal{F}_h(\mathbf{0})$, we can write $\hat{f}_h = \mathrm{clip}_{[0,H]}\left(\langle\bar{\phi}_h,\hat{w}_h\rangle\right)$, where $\hat{w}_h = \mathrm{argmin}_{\|w_h\|_2\leq B}\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h,w_h,V_{h+1})$. This implies $\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h,\hat{w}_h,V_{h+1})-\mathcal{L}^{\mathrm{c}}_{\mathcal{D}_h}(\bar{\phi}_h,\tilde{w}_{V_{h+1}},V_{h+1})\leq 0$.

Union bounding over $h\in[H]$ and $R\in\mathcal{R}$, and set $\delta=\delta'/\left(|\overline{\mathcal{W}}_h||\overline{\mathcal{W}}_{h+1}||\mathcal{R}|H\right)$, we get that with probability at least $1-\delta$,

$$\left|\mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\bar{\phi}_h,\hat{w}_h,V_{h+1})-\mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\phi^*_h,\theta^*_{V_{h+1}},V_{h+1})\right|\leq\frac{100H^2d\log\frac{2nB|\mathcal{R}|}{\delta}}{n}+3\varepsilon_{\mathrm{apx}}.$$

Finally, noticing the property that $\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h}(\hat{f}_h,V_{h+1})\right] = \mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\bar{\phi}_h,\hat{w}_h,V_{h+1})$ and $\mathbb{E}\left[\mathcal{L}_{\mathcal{D}_h,R_h}(\mathcal{T}_hV_{h+1},V_{h+1})\right] = \mathcal{L}_{\rho_{h-3}^{+3}}(\phi^*_h,\theta^*_{V_{h+1}},V_{h+1}) = \mathcal{L}^{\mathrm{c}}_{\rho_{h-3}^{+3}}(\phi^*_h,\theta^*_{V_{h+1}},V_{h+1})$, we complete the proof. □

B.2 Auxiliary Lemmas for Chapter 5

B.2.1 Proof of Lemma 5.1

In this part, we provide the proof of Lemma 5.1 for completeness. This result is widely used throughout the paper.

Lemma (Restatement of Lemma 5.1)**.** *For a low-rank MDP $\mathcal{M}$ with embedding dimension d, for any function $f:\mathcal{S}\to[0,1]$, we have:*

$$\mathbb{E}\left[f(s_{h+1})|s_h,a_h\right]=\left\langle\phi^*_h(s_h,a_h),\theta^*_f\right\rangle$$

where $\theta^*_f\in\mathbb{R}^d$ and we have $\|\theta^*_f\|_2\leq\sqrt{d}$. A similar linear representation is true for $\mathbb{E}_{a\sim\pi_{h+1}}[f(s_{h+1},a)|s_h,a_h]$ where $f:\mathcal{S}\times\mathcal{A}\to[0,1]$ and a policy $\pi_{h+1}:\mathcal{S}\to\Delta(\mathcal{A})$.

Proof. For state-value function f, we have

$$\begin{aligned}\mathbb{E}\left[f(s_{h+1})|s_h, a_h\right] &= \int f(s_{h+1})T_h(s_{h+1}|s_h, a_h)d(s_{h+1}) \\ &= \int f(s_{h+1})\left\langle \phi_h^*(s_h, a_h), \mu_h^*(s_{h+1})\right\rangle d(s_{h+1}) \\ &= \left\langle \phi_h^*(s_h, a_h), \int f(s_{h+1})\mu_h^*(s_{h+1})d(s_{h+1})\right\rangle \\ &= \left\langle \phi_h^*(s_h, a_h), \theta_f^*\right\rangle,\end{aligned}$$

where $\theta_f^* \coloneqq \int f(s_{h+1})\mu_h^*(s_{h+1})d(s_{h+1})$ is a function of f. Additionally, we obtain $\|\theta_f^*\|_2 \leq \sqrt{d}$ from Definition 5.1.

For Q-value function f, we similarly have

$$\mathbb{E}_{a\sim\pi_{h+1}}\left[f(s_{h+1}, a)|s_h, a_h\right] = \left\langle \phi_h^*(s_h, a_h), \theta_f^*\right\rangle,$$

where $\theta_f^* \coloneqq \int\int f(s_{h+1}, a_{h+1})\pi(a_{h+1}|s_{h+1})\mu_h^*(s_{h+1})d(s_{h+1})d(a_{h+1})$ and $\|\theta_f^*\|_2 \leq \sqrt{d}$. □

B.2.2 Deviation bound for regression with squared loss

In this section, we derive a generalization error bound for squared loss for a class $\mathcal{F}$ which subsumes the discriminator classes $\mathcal{F}_h$ and $\mathcal{G}_h$ in the main text. When we apply Lemma B.8, $(s^{(i)}, a^{(i)}, s'^{(i)})$ tuples usually stands for $(s_h^{(i)}, a_h^{(i)}, s_{h+1}^{(i)})$ tuples, and function classes Φ and Φ' usually refers to Φ_h and Φ_{h+1}. In the proof of Lemma B.8, we abuse the notation of $\mathcal{F}$, $\mathcal{G}$, and Φ, and they are different from $\mathcal{F}_h$, $\mathcal{G}_h$, and Φ in the main text.

Lemma B.8. *For a dataset $\mathcal{D} \coloneqq \left\{(s^{(i)}, a^{(i)}, s'^{(i)})\right\}_{i=1}^n \sim \rho$, finite feature classes Φ and Φ' and finite reward function class $\mathcal{R} : \mathcal{S}\times\mathcal{A} \to [0, 1]$, we can show that, with probability at least $1-\delta$:*

$$\begin{aligned}&\left|\mathcal{L}_\rho(\phi, w, f) - \mathcal{L}_\rho(\phi^*, \theta_f^*, f) - \left(\mathcal{L}_\mathcal{D}(\phi, w, f) - \mathcal{L}_\mathcal{D}(\phi^*, \theta_f^*, f)\right)\right| \\ \leq &\frac{1}{2}\left(\mathcal{L}_\rho(\phi, w, f) - \mathcal{L}_\rho(\phi^*, \theta_f^*, f)\right) + \frac{32d(B + L\sqrt{d})^2\log(2n|\Phi||\Phi'||\mathcal{R}|/\delta)}{n}\end{aligned}$$

for all $\phi \in \Phi$, $\|w\|_2 \leq B$ and function $f : \mathcal{S} \to [0, 1]$ in class $\mathcal{F} \coloneqq \{f(s') = \mathrm{clip}_{[0,L]}(\mathbb{E}_{a'\sim\pi'(s')}[R(s', a') + \langle\phi'(s', a'), \theta\rangle]) : \phi' \in \Phi', \|\theta\|_2 \leq B, R \in \mathcal{R}\}$ for any policy π' over s'.

Proof. Consider a function $f \in \mathcal{F}$ such that $f(s') = \mathrm{clip}_{[0,L]}\big(\mathbb{E}[R(s', a) + \langle\phi'(s', a), \theta\rangle]\big)$. Applying Lemma 5.1, we know that for every such f, there exists some θ_f^*, s.t. $\mathbb{E}[f(s')|s, a] = \left\langle\phi^*(s, a), \theta_f^*\right\rangle$ and $\|\theta_f^*\| \leq L\sqrt{d}$.

To begin, we introduce notation for the discriminator class with a single reward function R as $\mathcal{F}(R) \coloneqq \{f(s') = \mathrm{clip}_{[0,L]}(\mathbb{E}_{a\sim\pi'(s')}\left[R(s',a') + \langle\phi'(s',a'),\theta\rangle\right]) : \phi' \in \Phi', \|\theta\|_2 \le B\}$. Then we fix $R \in \mathcal{R}, \phi \in \Phi, \|w\|_2 \le B$, $f \in \mathcal{F}(R)$, and $\tilde{\theta}_f$, where $\tilde{\theta}_f$ satisfies $\|\tilde{\theta}_f\|_2 \le L\sqrt{d}$ and for all (s,a),

$$\left|\left\langle\phi^*(s,a),\tilde{\theta}_f\right\rangle - \mathbb{E}[f(s'|s,a)]\right| = \left|\left\langle\phi^*(s,a),\tilde{\theta}_f\right\rangle - \langle\phi^*(s,a),\theta_f^*\rangle\right| \le \gamma.$$

We first give a high probability bound on the following deviation term:

$$\left|\mathcal{L}_\rho(\phi,w,f) - \mathcal{L}_\rho(\phi^*,\tilde{\theta}_f,f) - \left(\mathcal{L}_\mathcal{D}(\phi,w,f) - \mathcal{L}_\mathcal{D}(\phi^*,\tilde{\theta}_f,f)\right)\right|.$$

Let $g(s,a) = \langle\phi(s,a),w\rangle$ and $\tilde{g}(s,a) = \left\langle\phi^*(s,a),\tilde{\theta}_f\right\rangle$. For the random variable $Y \coloneqq (g(s,a) - f(s',a'))^2 - (\tilde{g}(s,a) - f(s',a'))^2$, we have:

$$\begin{aligned}
\mathbb{E}[Y] &= \mathbb{E}\left[(g(s,a) - f(s',a'))^2 - (\tilde{g}(s,a) - f(s',a'))^2\right] \\
&= \mathbb{E}\left[(g(s,a) + \tilde{g}(s,a) - 2f(s',a'))\,(g(s,a) - \tilde{g}(s,a))\right] \\
&= \mathbb{E}\left[\left(g(s,a) + \tilde{g}(s,a) - 2\left(\left\langle\phi^*,\theta_f^*\right\rangle + \mathrm{noise}(s,a,s',a')\right)\right)(g(s,a) - \tilde{g}(s,a))\right] \\
&= \mathbb{E}\left[\left(g(s,a) - \tilde{g}(s,a) - 2\left(\left\langle\phi^*,\theta_f^*\right\rangle - \tilde{g}(s,a) + \mathrm{noise}(s,a,s',a')\right)\right)(g(s,a) - \tilde{g}(s,a))\right] \\
&= \mathbb{E}\left[(g(s,a) - \tilde{g}(s,a))^2 - 2\left(\left\langle\phi^*,\theta_f^*\right\rangle - \tilde{g}(s,a)\right)(g(s,a) - \tilde{g}(s,a))\right].
\end{aligned}$$

Here the expectation is taken according to the distribution ρ. In the third equality, for sample (s,a,s',a'), we denote $f(s',a') - \mathbb{E}[(f(s',a'|s,a)]$ as $\mathrm{noise}(s,a,s',a')$. The last equality is due to the fact that

$$\mathbb{E}[\mathrm{noise}(s,a,s',a')\,(g(s,a) - \tilde{g}(s,a))] = \mathbb{E}_{s,a}[\mathbb{E}_{s',a'|s,a}[\mathrm{noise}(s,a,s',a')(g(s,a) - \tilde{g}(s,a)]] = 0.$$

Noticing the approximation assumption of $\tilde{\theta}_f$, and $g(s,a)$ and $\tilde{g}(s,a)$ is bounded by $[B,B]$ and $[-L\sqrt{d}, L\sqrt{d}]$ respectively, we have

$$\left|\mathbb{E}[Y] - \mathbb{E}\left[(g(s,a) - \tilde{g}(s,a))^2\right]\right| \le 2\left\|g - \tilde{g}\right\|_\infty \left\|\left\langle\phi^*,\theta_f^*\right\rangle - \tilde{g}\right\|_\infty \le 2(B + L\sqrt{d})\gamma. \tag{B.4}$$

Next, for the variance of the random variable, we have:

$$\begin{aligned}
\mathbb{V}[Y] \le \mathbb{E}\left[Y^2\right] &= \mathbb{E}\left[(g(s,a) + \tilde{g}(s,a) - 2f(s',a'))^2\,(g(s,a) - \tilde{g}(s,a))^2\right] \\
&\le 4(B + L\sqrt{d})^2\mathbb{E}\left[(g(s,a) - \tilde{g}(s,a))^2\right] \\
&\le 4(B + L\sqrt{d})^2\mathbb{E}[Y] + 8(B + L\sqrt{d})^3\gamma.
\end{aligned}$$

Noticing $Y \in [-(B + L\sqrt{d})^2, (B + L\sqrt{d})^2]$ and applying Bernstein's inequality, with probability

at least $1-\delta'$, we can bound the deviation term above as:

$$
\begin{aligned}
&\left|\mathcal{L}_\rho(\phi, w, f) - \mathcal{L}_\rho(\phi^*, \tilde{\theta}_f, f) - \Big(\mathcal{L}_\mathcal{D}(\phi, w, f) - \mathcal{L}_\mathcal{D}(\phi^*, \tilde{\theta}_f, f)\Big)\right| \\
&\leq \sqrt{\frac{2\mathbb{V}[Y]\log(2/\delta')}{n}} + \frac{4(B+L\sqrt{d})^2\log(2/\delta')}{3n} \\
&\leq \sqrt{\frac{\Big(8(B+L\sqrt{d})^2\mathbb{E}[Y] + 16(B+L\sqrt{d})^3\gamma\Big)\log(2/\delta')}{n}} + \frac{4(B+L\sqrt{d})^2\log(2/\delta')}{3n} \\
&\leq \sqrt{\frac{8(B+L\sqrt{d})^2\mathbb{E}[Y]\log(2/\delta')}{n}} + \sqrt{\frac{16(B+L\sqrt{d})^3\gamma\log(2/\delta')}{n}} + \frac{4(B+L\sqrt{d})^2\log(2/\delta')}{3n} \\
&\leq \sqrt{\frac{8(B+L\sqrt{d})^2\mathbb{E}[Y]\log(2/\delta')}{n}} + \frac{4(B+L\sqrt{d})^2\log(2/\delta')}{n} + \frac{4(B+L\sqrt{d})^2\log(2/\delta')}{3n},
\end{aligned}
$$

where in the last inequality is obtained by choosing $\gamma = \frac{(B+L\sqrt{d})}{n}$.

Further, consider a finite point-wise cover of the function class $\mathcal{G} \coloneqq \{g(s,a) = \langle\phi(s,a), w\rangle : \phi \in \Phi, \|w\|_2 \leq B\}$. Note that, with a ℓ_2-cover $\overline{\mathcal{W}}$ of $\mathcal{W} = \{\|w\|_2 \leq B\}$ at scale γ, we have for all (s,a) and $\phi \in \Phi$, there exists $\bar{w} \in \overline{\mathcal{W}}$, $|\langle\phi(s,a), w - \bar{w}\rangle| \leq \gamma$. Similarly, for any $f \in \mathcal{F}(R)$, we cover the linear term in $\mathcal{F}(R)$. We again choose Θ as an ℓ_2-cover of the set $\{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq B\}$ at scale γ. For all $f \in \mathcal{F}(R)$, we know that there exists some ϕ' so that $f(\cdot) = \mathrm{clip}_{[0,L]}\left(\mathbb{E}_{a'\sim\pi'}\left[R(\cdot,a') + \langle\phi'(\cdot,a'),\theta\rangle\right]\right) \in \mathcal{F}(R)$. Since there exists $\tilde{\theta} \in \Theta$ such that $\left\|\tilde{\theta} - \theta\right\|_2 \leq \gamma$, we know that we can pick $\tilde{f} = \mathrm{clip}_{[0,L]}\left(\mathbb{E}_{a'\sim\pi'}\left[R(\cdot,a') + \left\langle\phi'(\cdot,a'),\tilde{\theta}\right\rangle\right]\right)$ (which is in our covering set) and get $\left|\left\langle\phi^*(s,a), \tilde{\theta}_f - \theta^*_f\right\rangle\right| = \left|\mathbb{E}[f(s'|s,a)] - \mathbb{E}[\tilde{f}(s'|s,a)]\right| \leq \left\|\tilde{f} - f\right\|_\infty \leq \gamma$ (here, we use $\tilde{\theta}_f = \theta^*_{\tilde{f}}$). Via standard argument, we can set $|\Theta| = \left(\frac{2B}{\gamma}\right)^d$ and $|\overline{\mathcal{W}}| = \left(\frac{2B}{\gamma}\right)^d$, which implies $|\Theta| \leq (2n)^d$ and $|\overline{\mathcal{W}}| \leq (2n)^d$.

Thus, applying a union bound over elements in $\overline{\mathcal{W}}$, Θ, Φ, Φ' and $\mathcal{R}$, with probability $1-$

$|\Phi||\Phi'||\overline{\mathcal{W}}||\Theta||\mathcal{R}|\delta'$, for all $w \in \mathcal{W}$, $\phi \in \Phi$ and $f \in \mathcal{F}$, we have:

$$
\begin{aligned}
&\left|\mathcal{L}_\rho(\phi, w, f) - \mathcal{L}_\rho(\phi^*, \theta_f^*, f) - \left(\mathcal{L}_\mathcal{D}(\phi, w, f) - \mathcal{L}_\mathcal{D}(\phi^*, \theta_f^*, f)\right)\right| \\
\leq &\left|\mathcal{L}_\rho(\phi, \bar{w}, f) - \mathcal{L}_\rho(\phi^*, \tilde{\theta}_f, f) - \left(\mathcal{L}_\mathcal{D}(\phi, \bar{w}, f) - \mathcal{L}_\mathcal{D}(\phi^*, \tilde{\theta}_f, f)\right)\right| + 4(B + L\sqrt{d})\gamma \\
= &\left|\mathcal{L}_\rho(\phi, \bar{w}, f) - \mathcal{L}_\rho(\phi^*, \tilde{\theta}_f, f) - \left(\mathcal{L}_\mathcal{D}(\phi, \bar{w}, f) - \mathcal{L}_\mathcal{D}(\phi^*, \tilde{\theta}_f, f)\right)\right| + \frac{4(B + L\sqrt{d})^2}{n} \\
\leq &\sqrt{\frac{8(B + L\sqrt{d})^2 \mathbb{E}[Y_{\bar{w}}] \log \frac{2}{\delta'}}{n}} + \frac{4(B + L\sqrt{d})^2 \log \frac{2}{\delta'}}{n} + \frac{4(B + L\sqrt{d})^2 \log \frac{2}{\delta'}}{3n} + \frac{4(B + L\sqrt{d})^2}{n} \\
\leq &\frac{1}{2}\mathbb{E}[Y_{\bar{w}}] + \frac{4(B + L\sqrt{d})^2 \log(2/\delta')}{n} + \frac{(4 + 4 + 4/3)(B + L\sqrt{d})^2 \log(2/\delta')}{n} \\
\leq &\frac{1}{2}\mathbb{E}[Y_w] + \frac{2(B + L\sqrt{d})^2}{n} + \frac{4(B + L\sqrt{d})^2 \log(2/\delta')}{n} + \frac{(4 + 4 + 4/3)(B + L\sqrt{d})^2 \log(2/\delta')}{n} \\
\leq &\frac{1}{2}\left(\mathcal{L}_\rho(\phi, w, f) - \mathcal{L}_\rho(\phi^*, \theta_f^*, f)\right) + \frac{16(B + L\sqrt{d})^2 \log(2/\delta')}{n},
\end{aligned}
$$

where we add subscript to Y to distinguish $Y_{\bar{w}} \coloneqq \left(\langle \phi(s,a), \bar{w}\rangle - f(s',a')\right)^2 - \left(\tilde{g}(s,a) - f(s',a')\right)^2$ from $Y_w \coloneqq \left(\langle \phi(s,a), w\rangle - f(s',a')\right)^2 - \left(\tilde{g}(s,a) - f(s',a')\right)^2$.

Finally, setting $\delta = \delta' / \left(|\Phi||\Phi'||\overline{\mathcal{W}}||\Theta||\mathcal{R}|\right)$, we get $\log(2/\delta') \leq \log(2(2n)^{2d}|\Phi||\Phi'||\mathcal{R}|/\delta) \leq 2d \log(2n|\Phi||\Phi'||\mathcal{R}|/\delta)$. This completes the proof. □

B.2.3 Generalized elliptic potential lemma

Lemma B.9 (Generalized elliptic potential lemma, adapted from Proposition 1 of Carpentier et al. (2020))**.** *For any sequence of vectors $\theta_1^*, \theta_2^* \ldots, \theta_T^* \in \mathbb{R}^{d\times T}$ where $\|\theta_i^*\| \leq L\sqrt{d}$, for $\lambda \geq L^2 d$, we have:*

$$\sum_{t=1}^{T} \|\Sigma_t^{-1}\theta_{t+1}^*\|_2 \leq 2\sqrt{\frac{dT}{\lambda}}.$$

Proof. Proposition 1 from Carpentier et al. (2020) shows that for any bounded sequence of vectors $\theta_1^*, \theta_2^*, \ldots, \theta_T^* \in \mathbb{R}^{d\times T}$, we have:

$$\sum_{t=1}^{T} \|\Sigma_t^{-1}\theta_t^*\|_2 \leq \sqrt{\frac{dT}{\lambda}}.$$

Now, we have:

$$\Sigma_t = \Sigma_{t-1} + \theta_t^* \theta_t^{*\top} \preceq \Sigma_{t-1} + \lambda I_{d\times d} \preceq 2\Sigma_{t-1}$$

where we use the fact that $\|\theta_t^* \theta_t^{*\top}\|_2 \leq L^2 d \leq \lambda$. Using this relation, we can show the property that for any vector $x \in \mathbb{R}^d$, $4x^\top \Sigma_t^{-2} x \geq x^\top \Sigma_{t-1}^{-2} x$.

First noticing the above p.s.d. dominance inequality, we have $I_{d\times d}/2 \preceq \Sigma_t^{-1/2}\Sigma_{t-1}\Sigma_t^{-1/2}$. Therefore, all the eigenvalues of $\Sigma_t^{-1/2}\Sigma_{t-1}\Sigma_t^{-1/2}$ (thus $\Sigma_t^{-1}\Sigma_{t-1}$) are no less than 1/2. Applying SVD decomposition, we can get all eigenvalues of matrix $\Sigma_{t-1}\Sigma_t^{-2}\Sigma_{t-1} = (\Sigma_t^{-1}\Sigma_{t-1})^\top(\Sigma_t^{-1}\Sigma_{t-1})$ are no less than 1/4. Then consider any vector $y \in \mathbb{R}^d$, we have $4y^\top \Sigma_{t-1}\Sigma_t^{-2}\Sigma_{t-1} y \geq y^\top y$. Let $x = \Sigma_{y-1}^{-1} y$, we get this property.

Applying the above result, we finally have the following:

$$\sum_{t=1}^{T} \|\Sigma_t^{-1}\theta_{t+1}^*\|_2 \leq 2\sum_{t=1}^{T} \|\Sigma_t^{-1}\theta_t^*\|_2 \leq 2\sqrt{\frac{dT}{\lambda}}. \qquad \square$$

B.2.4 Covering lemma for the elliptical reward class

In this part, we provide the statistical complexity of the elliptical reward class. The result is used when we analyze the elliptical planner.

Lemma B.10 (Covering lemma for the elliptical reward class)**.** *For the elliptical reward class* $\mathcal{R}_h \coloneqq \{\phi_h^\top \Gamma^{-1}\phi_h : \phi_h \in \Phi_h, \Gamma \in \mathbb{R}^{d\times d}, \lambda_{\min}(\Gamma) \geq 1\}$*, where* $h \in [H]$*, there exists a* γ*-cover* $\mathcal{C}_{\mathcal{R}_h}$ *of size* $|\Phi_h|(2\sqrt{d}/\gamma)^{d^2}$.

Proof. Firstly, for any $\Gamma \in \mathbb{R}^{d\times d}$ with $\lambda_{\min}(\Gamma) \geq 1$, applying matrix norm inequality yields $\|\Gamma\|_{\mathrm{F}} \geq \sqrt{d}$. Further, we have

$$\|\Gamma^{-1}\|_{\mathrm{F}} \leq \frac{\|I_{d\times d}\|_{\mathrm{F}}}{\|\Gamma\|_{\mathrm{F}}} \leq \sqrt{d}.$$

Next, consider the matrix class $\bar{A} \coloneqq \{A \in \mathbb{R}^{d\times d} : \|A\|_{\mathrm{F}} \leq \sqrt{d}\}$. From the definition of the Frobenius norm, for any $A \in \bar{A}$ and any (i,j)-th element, we have $|A_{ij}| \leq \sqrt{d}$. Applying the standard covering argument for each of the d^2 elements, there exists a γ-cover of $\bar{A}$, whose size is upper bounded by $(2\sqrt{d}/\gamma)^{d^2}$. Denote this γ-cover as $\bar{A}_\gamma$. For any $\Gamma \in \mathbb{R}^{d\times d}$ with $\lambda_{\min}(\Gamma) \geq 1$, we can pick some $A \in \bar{A}_\gamma$ so that $\|\Gamma^{-1} - A_\Gamma\|_{\mathrm{F}} \leq \gamma$. Then for any $\phi_h \in \Phi_h$, we have

$$|\phi_h^\top \Gamma^{-1}\phi_h - \phi_h^\top A_\Gamma \phi_h| \leq \sup_{v:\|v\|_2\leq 1} |v^\top(\Gamma^{-1} - A_\Gamma)v| \leq \|\Gamma^{-1} - A_\Gamma\|_{\mathrm{F}} \leq \gamma.$$

This implies that $\mathcal{C}_{\mathcal{R}_h} \coloneqq \{\phi_h^\top A \phi_h : \phi_h \in \Phi_h, A \in \bar{A}_\gamma\}$ is a γ-cover of $\mathcal{R}_h$, which completes the proof. $\square$

BIBLIOGRAPHY

Yasin Abbasi-Yadkori and Gergely Neu. Online learning in mdps with side information. *arXiv preprint arXiv:1406.6812*, 2014. 51, 58

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 1–26. JMLR Workshop and Conference Proceedings, 2011. 31, 32, 152, 170, 175

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011. 76, 156, 159, 204

Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *Artificial Intelligence and Statistics*, pages 1–9, 2012. 58

Alekh Agarwal. Selective sampling algorithms for cost-sensitive multiclass prediction. In *International Conference on Machine Learning*, pages 1220–1228, 2013. 48, 49

Alekh Agarwal, Alexander Rakhlin, and Peter Bartlett. Matrix regularization techniques for online multitask learning. Technical Report UCB/EECS-2008-138, EECS Department, University of California, Berkeley, Oct 2008. URL http://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-138.html. 110

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E Schapire. Corralling a band of bandit algorithms. In *Conference on Learning Theory*, pages 12–38. PMLR, 2017. 100

Alekh Agarwal, Mikael Henaff, Sham Kakade, and Wen Sun. PC-PG: Policy cover directed exploration for provable policy gradient learning. In *Advances in Neural Information Processing Systems*, 2020a. 114

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in Neural Information Processing Systems*, 2020b. 114, 115, 116, 118, 127, 149, 150

Pierre Alquier, Massimiliano Pontil, et al. Regret bounds for lifelong learning. In *Artificial Intelligence and Statistics*, pages 261–269. PMLR, 2017. 169

Haitham B Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. Online multi-task learning for policy gradient methods. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1206–1214, 2014. 36

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853, 2005. 169

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 86

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 2008. 26, 122

Raman Arora, Teodor Vanislavov Marinov, and Mehryar Mohri. Corralling stochastic bandit algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 2116–2124. PMLR, 2021. 100

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002. 96

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020. 99, 126, 127

Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. On the sample complexity of reinforcement learning with a generative model. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1707–1714. Omnipress, 2012. 99

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org, 2017. 22

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995. 122

Sumanta Basu, Ali Shojaie, and George Michailidis. Network granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 16(1):417–453, 2015. 153

Marc Bellemare, Will Dabney, Robert Dadashi, Adrien Ali Taiga, Pablo Samuel Castro, Nicolas Le Roux, Dale Schuurmans, Tor Lattimore, and Clare Lyle. A geometric perspective on optimal representations for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 127

Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020. 1

Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716, 1952. 8

Carolin Benjamins, Theresa Eimer, Frederik Schubert, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Carl: A benchmark for contextual and adaptive reinforcement learning. *arXiv preprint arXiv:2110.02102*, 2021. 58, 59

Dimitri P Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996. 8, 12

Aurélien F Bibaut, Antoine Chambaz, and Mark J van der Laan. Rate-adaptive model selection over a collection of black-box contextual bandit algorithms. *arXiv preprint arXiv:2006.03632*, 2020. 100

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer school on machine learning*, pages 169–207. Springer, 2003. 2

Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996. 16

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002. 20, 37, 38

Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 122–131. AUAI Press, 2013. 56

Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion. In *Advances in Neural Information Processing Systems*, pages 8224–8234, 2018. 86

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018. 128

Apostolos N Burnetas and Michael N Katehakis. Optimal adaptive policies for markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, 1997. 21

MC Campi and PR Kumar. Optimal adaptive control of an lqg system. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 1, pages 349–353. IEEE, 1996. 31

Tongyi Cao and Akshay Krishnamurthy. Provably adaptive reinforcement learning in metric spaces. *Advances in Neural Information Processing Systems*, 33:9736–9744, 2020. 57

Alexandra Carpentier, Claire Vernade, and Yasin Abbasi-Yadkori. The elliptical potential lemma revisited. *arxiv:2010.10182*, 2020. 143, 222

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 169

Asaf Cassel, Alon Cohen, and Tomer Koren. Logarithmic regret for learning linear quadratic regulators efficiently. In *International Conference on Machine Learning*, pages 1328–1337. PMLR, 2020. 169

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019. 26, 119, 121

Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, and Wojciech Zaremba. Transfer from simulation to real world through learning deep inverse dynamics model. *arXiv preprint arXiv:1610.03518*, 2016. 85

Christoph Dann and Emma Brunskill. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826, 2015. 2, 23, 38, 41, 65

Christoph Dann, Gerhard Neumann, Jan Peters, et al. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014. 25

Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying pac and regret: uniform pac bounds for episodic reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5717–5727, 2017. 20, 23, 24

Christoph Dann, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. On oracle-efficient pac rl with rich observations. In *Advances in Neural Information Processing Systems*, pages 1422–1432, 2018. 93, 127

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *International Conference on Machine Learning*, pages 1507–1516. PMLR, 2019. 20, 22, 23, 51, 53, 58, 75

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4192–4201, 2018. 152, 155, 175, 197

Kenji Doya, Kazuyuki Samejima, Ken-ichi Katagiri, and Mitsuo Kawato. Multiple model-based reinforcement learning. *Neural computation*, 14(6):1347–1369, 2002. 99, 101

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, 2019a. 114, 116, 127, 201

Simon S Du, Sham M Kakade, Ruosong Wang, and Lin F Yang. Is a good representation sufficient for sample efficient reinforcement learning? In *International Conference on Learning Representations*, 2019b. 16, 17, 114

Simon S Du, Ruosong Wang, Mengdi Wang, and Lin F Yang. Continuous control with contexts, provably. *arXiv preprint arXiv:1910.13614*, 2019c. 170

Simon Shaolei Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. In *International Conference on Learning Representations*, 2020. 157, 170, 180

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008. 45

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite time identification in unstable linear systems. *Automatica*, 96:342–353, 2018a. 29, 153, 157, 159, 160, 181

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Finite-time adaptive stabilization of linear systems. *IEEE Transactions on Automatic Control*, 64(8):3498–3505, 2018b. 155, 171, 202

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. On adaptive linear–quadratic regulators. *Automatica*, 117:108982, 2020a. 160

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. On adaptive linear–quadratic regulators. *Automatica*, 117:108982, 2020b. 152

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Input perturbations for adaptive control and learning. *Automatica*, 117:108950, 2020c. 31, 33, 156

Mohamad Kazem Shirani Faradonbeh, Ambuj Tewari, and George Michailidis. Optimism-based adaptive regulation of linear-quadratic systems. *IEEE Transactions on Automatic Control*, 2020d. 31, 32, 33

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*, 2017. 101, 127, 128

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pages 162–169, 2004. 15

Claude-Nicolas Fiechter. Efficient reinforcement learning. In *Proceedings of the seventh annual conference on Computational learning theory*, pages 88–97, 1994. 23

Dylan J Foster, Akshay Krishnamurthy, and Haipeng Luo. Model selection for contextual bandits. *Advances in Neural Information Processing Systems*, 32:14741–14752, 2019. 97, 100

Dylan J Foster, Alexander Rakhlin, David Simchi-Levi, and Yunzong Xu. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. In *Advances in Neural Information Processing Systems*, 2020. 127

André Fujita, Joao R Sato, Humberto M Garay-Malpartida, Rui Yamaguchi, Satoru Miyano, Mari C Sogayar, and Carlos E Ferreira. Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC systems biology*, 1(1):1–11, 2007. 153

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017. 157

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. DeepMDP: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2019. 114, 127

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003. 15, 98

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019. 114, 127

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015. 3, 36, 37, 56, 57

Botao Hao, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Online sparse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, 2021. 114, 126

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007. 49, 84

Jiafan He, Dongruo Zhou, and Quanquan Gu. Uniform-pac bounds for reinforcement learning with linear function approximation. *arXiv preprint arXiv:2106.11612*, 2021. 53

Jiachen Hu, Xiaoyu Chen, Chi Jin, Lihong Li, and Liwei Wang. Near-optimal representation learning for linear bandits and linear rl. In *International Conference on Machine Learning*, pages 4349–4358. PMLR, 2021. 157, 170, 190, 191

Petros A Ioannou and Jing Sun. *Robust adaptive control*. Courier Corporation, 2012. 152

Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. On the convergence of stochastic iterative dynamic programming algorithms. *Neural computation*, 6(6):1185–1201, 1994. 2

Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017. 157

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010. 2, 21, 22, 51, 54, 55

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016. 25

Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 179–188, 2015. 95

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713. JMLR. org, 2017. 23, 91, 93, 106, 114, 127

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4868–4878, 2018. 13, 22, 24, 57

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020. 4, 16, 22, 99, 114, 117, 126, 170

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021. 26

Nicholas K Jong and Peter Stone. Model-based function approximation in reinforcement learning. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, 2007. 25

Katarina Juselius, Zorica Mladenovic, et al. *High inflation, hyperinflation and explosive roots: the case of Yugoslavia*. Citeseer, 2002. 30

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003. 20, 40, 60, 62

Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored mdps. In *IJCAI*, volume 16, pages 740–747, 1999. 14

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2):209–232, 2002. 2, 20, 39, 61, 103

Taylor Killian, George Konidaris, and Finale Doshi-Velez. Transfer learning across patient variations with hidden parameter Markov decision processes. *arXiv preprint arXiv:1612.00475*, 2016. 36

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 163

Pascal Klink, Hany Abdulsamad, Boris Belousov, and Jan Peters. Self-paced contextual reinforcement learning. In *Conference on Robot Learning*, pages 513–529. PMLR, 2020a. 58, 59

Pascal Klink, Carlo D'Eramo, Jan R Peters, and Joni Pajarinen. Self-paced deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020b. 60

Kenneth R Koedinger, Emma Brunskill, Ryan SJd Baker, Elizabeth A McLaughlin, and John Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013. 1

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems 29*, pages 1840–1848, 2016. 95

Miroslav Krstic, Petar V Kokotovic, and Ioannis Kanellakopoulos. *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995. 152

Vladimír Kučera. The discrete riccati equation of optimal control. *Kybernetika*, 8(5):430–447, 1972. 28

Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJJinbWRZ. 86

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. Rl for latent mdps: Regret guarantees and a lower bound. *arXiv preprint arXiv:2102.04939*, 2021. 201

Sahin Lale, Kamyar Azizzadenesheli, Babak Hassibi, and Anima Anandkumar. Explore more and improve regret in linear quadratic regulators. *arXiv preprint arXiv:2007.12291*, 2020. 31, 32

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020. 127

Tor Lattimore and Csaba Szepesvari. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, 2020. 114

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. 49, 51, 75, 126

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996. 47

A. Lazaric. Transfer in reinforcement learning: a framework and a survey. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State of the Art*. Springer, 2011. 56

Gilwoo Lee, Brian Hou, Aditya Mandalika, Jeongseok Lee, and Siddhartha S. Srinivasa. Bayesian policy optimization for model uncertainty. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJGvns0qK7. 86, 100, 101

Jonathan Lee, Aldo Pacchiano, Vidya Muthukumar, Weihao Kong, and Emma Brunskill. Online model selection for reinforcement learning with function approximation. In *International Conference on Artificial Intelligence and Statistics*, 2021. 126

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 5757–5766. PMLR, 2020. 58, 59

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *ISAIM*, 4:5, 2006. 14, 98

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010. 1

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 85

Zichuan Lin, Garrett Thomas, Guangwen Yang, and Tengyu Ma. Model-based adversarial meta-reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020. 123

Rui Lu, Gao Huang, and Simon S Du. On the power of multitask representation learning in linear mdp. *arXiv preprint arXiv:2106.08053*, 2021. 170

MM Mahmud, Majd Hawasly, Benjamin Rosman, and Subramanian Ramamoorthy. Clustering Markov decision processes for continual transfer. *arXiv preprint arXiv:1311.3959*, 2013. 56

Horia Mania, Stephen Tu, and B. Recht. Certainty equivalence is efficient for linear quadratic control. In *NeurIPS*, 2019. 31, 33, 34

Andreas Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research*, 7 (Jan):117–139, 2006. 169

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016. 169

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, 2020. 114, 116, 127

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. 1, 85, 114

Aditya Modi and Ambuj Tewari. No-regret exploration in contextual reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pages 829–838. PMLR, 2020. 3

Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. Markov decision processes with continuous side information. In *Algorithmic Learning Theory*, pages 597–618, 2018. 3, 103

Aditya Modi, Nan Jiang, Ambuj Tewari, and Satinder Singh. Sample complexity of reinforcement learning using linearly combined model ensembles. In *International Conference on Artificial Intelligence and Statistics*, pages 2010–2020. PMLR, 2020. 4, 114

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank mdps. *arXiv preprint arXiv:2102.07035*, 2021. 4

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018. 2

Igor Mordatch, Kendall Lowrey, Galen Andrew, Zoran Popovic, and Emanuel V Todorov. Interactive control of diverse complex characters with neural networks. In *Advances in Neural Information Processing Systems*, pages 3132–3140, 2015. 1

Igor Mordatch, Nikhil Mishra, Clemens Eppner, and Pieter Abbeel. Combining model-based policy search with online model learning for control of physical humanoids. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 242–248. IEEE, 2016. 85

Rémi Munos. Performance bounds in l_p-norm for approximate value iteration. *SIAM journal on control and optimization*, 46(2):541–561, 2007. 10, 11

Inbal Nahum-Shani, Shawna N Smith, Bonnie J Spring, Linda M Collins, Katie Witkiewitz, Ambuj Tewari, and Susan A Murphy. Just-in-time adaptive interventions (jitais) in mobile health: key components and design principles for ongoing health behavior support. *Annals of Behavioral Medicine*, 52(6):446–462, 2018. 1

Ian Osband and Benjamin Van Roy. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pages 1466–1474, 2014. 114, 126

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016. 54, 55

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011, 2013. 22

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29:4026–4034, 2016. 101

Aldo Pacchiano, Christoph Dann, Claudio Gentile, and Peter Bartlett. Regret bound balancing and elimination for model selection in bandits and rl. *arXiv:2012.13045*, 2020a. 100, 126

Aldo Pacchiano, My Phan, Yasin Abbasi Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. Model selection in contextual stochastic bandit problems. *Advances in Neural Information Processing Systems*, 33, 2020b. 100

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 752–759, 2008. 16

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019. 163

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, 2017. 114, 127, 128

Jason Pazis and Ronald Parr. Pac optimal exploration in continuous space markov decision processes. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013. 57

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 6

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=SyWvgP5el. 86, 100, 101

Samuel Ritter, Jane Wang, Zeb Kurth-Nelson, Siddhant Jayakumar, Charles Blundell, Razvan Pascanu, and Matthew Botvinick. Been there, done that: Meta-learning with episodic recall. In *International Conference on Machine Learning*, pages 4354–4363. PMLR, 2018. 59

Daniel Russo. Worst-case regret bounds for exploration via randomized value functions. In *Advances in Neural Information Processing Systems*, pages 14410–14420, 2019. 22, 53, 54, 79, 82

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pages 2256–2264, 2013. 201

Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In *International Conference on Machine Learning*, pages 5610–5618. PMLR, 2019. 153, 157, 159, 160, 184

Bruno Scherrer. Approximate policy iteration schemes: a comparison. In *International Conference on Machine Learning*, pages 1314–1322. PMLR, 2014. 26

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Mathematics of Operations Research*, 41(3):758–774, 2016. 12

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, 2020. 114, 127

Anil K Seth, Adam B Barrett, and Lionel Barnett. Granger causality analysis in neuroscience and neuroimaging. *Journal of Neuroscience*, 35(8):3293–3297, 2015. 153

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016. 85

Max Simchowitz and Dylan Foster. Naive exploration is optimal for online lqr. In *International Conference on Machine Learning*, pages 8937–8948. PMLR, 2020. 31, 33, 34, 152, 155, 165, 166, 168, 172, 173, 174, 175, 195, 196, 197

Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pages 439–473. PMLR, 2018. 153, 160, 175

Sean R Sinclair, Siddhartha Banerjee, and Christina Lee Yu. Adaptive discretization for episodic reinforcement learning in metric spaces. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–44, 2019. 57

Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38(3):287–308, 2000. 2

Satinder P Singh and Richard C Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994. 11

A Skripnikov and G Michailidis. Joint estimation of multiple network granger causal models. *Econometrics and Statistics*, 10:120–133, 2019a. 153

Andrey Skripnikov and George Michailidis. Regularized joint estimation of related vector autoregressive models. *Computational statistics & data analysis*, 139:164–177, 2019b. 153

Aleksandrs Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014. 41

Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9767–9779. PMLR, 2021. 59

Zhao Song and Wen Sun. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019. 57

James H Stock and Mark W Watson. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In *Handbook of macroeconomics*, volume 2, pages 415–525. Elsevier, 2016. 153

Alexander L Strehl and Michael L Littman. A theoretical analysis of model-based interval estimation. In *Proceedings of the 22nd international conference on Machine learning*, pages 856–863, 2005. 20, 38, 39

Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888, 2006. 20

Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009. 39

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, pages 2898–2933. PMLR, 2019. 14, 93, 114, 127

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991. 14

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in neural information processing systems*, pages 1038–1044, 1996. 13

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 1, 9

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000. 13

Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 993–1000, 2009. 16

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009. 56

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016. 25

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. 22

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 23–30. IEEE, 2017. 85, 86

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. 85

Nilesh Tripuraneni, Chi Jin, and Michael Jordan. Provable meta-learning of linear representations. In *International Conference on Machine Learning*, pages 10434–10443. PMLR, 2021. 170, 171

John N Tsitsiklis and Benjamin Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE transactions on automatic control*, 42(5):674–690, 1997. 16

Masatoshi Uehara, Masaaki Imaizumi, Nan Jiang, Nathan Kallus, Wen Sun, and Tengyang Xie. Finite sample analysis of minimax offline reinforcement learning: Completeness, fast rates and first-order efficiency. *arXiv preprint arXiv:2102.02981*, 2021. 26

Benjamin Van Roy and Shi Dong. Comments on the Du-Kakade-Wang-Yang lower bounds. *arXiv:1911.07910*, 2019. 114

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018. 178, 180

H Victor, De la Peña, Tze Leung Lai, and Qi-Man Shao. *Self-normalized processes: Limit theory and Statistical Applications*. Springer, 2009. 204

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019. 40, 149, 179

Thomas J Walsh, István Szita, Carlos Diuk, and Michael L Littman. Exploring compact reinforcement-learning representations with linear regression. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 591–598. AUAI Press, 2009. A corrected version is available as Technical Report DCS-tr-660, Department of Computer Science, Rutgers University, December, 2009. 43, 45, 68, 69

Ruosong Wang, S. Simon Du, F. Lin Yang, and Ruslan Salakhutdinov. On reward-free reinforcement learning with linear function approximation. In *Advances in Neural Information Processing Systems*, 2020a. 118, 120

Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33, 2020b. 114, 126

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992. 13

Gellert Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvari. On query-efficient planning in mdps under linear realizability of the optimal state-value function. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4355–4385. PMLR, 15–19 Aug 2021a. URL https://proceedings.mlr.press/v134/weisz21a.html. 16

Gellért Weisz, Csaba Szepesvári, and András György. Tensorplan and the few actions lower bound for planning in mdps under linear realizability of optimal value functions. *arXiv preprint arXiv:2110.02195*, 2021b. 16

Ward Whitt. Approximations of dynamic programs, i. *Mathematics of Operations Research*, 3(3): 231–243, 1978. 98

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021. 26

Lin Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004. PMLR, 2019. 4, 16, 99

Lin Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pages 10746–10756. PMLR, 2020. 114, 126

Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan. Bridging exploration and general function approximation in reinforcement learning: Provably efficient kernel and neural value iterations. *arXiv:2011.04622*, 2020. 114

Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*, pages 7304–7312. PMLR, 2019. 22

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. In *Advances in Neural Information Processing Systems*, 2020. 118, 120

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020. 114, 127

Lijun Zhang, Tianbao Yang, Rong Jin, Yichi Xiao, and Zhi-hua Zhou. Online stochastic linear optimization under one-bit feedback. In *International Conference on Machine Learning*, pages 392–401, 2016. 50, 70, 72, 73, 74

Dongruo Zhou, Quanquan Gu, and Csaba Szepesvari. Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR, 2021a. 99, 100

Dongruo Zhou, Jiafan He, and Quanquan Gu. Provably efficient reinforcement learning for discounted mdps with feature mapping. In *International Conference on Machine Learning*, pages 12793–12802. PMLR, 2021b. 99