Advancing Reinforcement Learning: Provably Efficient Algorithms for RL with Constraints and Function Approximation

Junxia Deng
University of Southern California
California, USA

Ceil Hong. Zhang
Massachusetts Institute of Technology
Massachusetts, USA
zhanghongceil@pku.org.cn

Abstract

Reinforcement learning has gained a surge of interest over the past years, fueled mainly by practical success and new applications in various domains. However, there is still a gap between our theoretical understanding of these RL techniques and their empirical success. In this thesis, we advance our understanding by studying reinforcement learning from a primarily theoretical point of view and designing provably efficient algorithms for two challenging settings of 1) RL with constraints and 2) RL with function approximation.

1) In standard RL, a learning agent seeks to optimize the overall reward. However, many key aspects of the desired behavior are more naturally expressed as constraints. First, we propose an algorithmic scheme that can handle RL tasks with general convex constraints improving upon prior works that are either limited to linear constraints or lack theoretical guarantees. Second, focusing on sample-efficient exploration, we develop the first provably efficient algorithm for tabular episodic constrained RL with the ability to handle convex constraints as well as the knapsack setting. Finally, motivated by recent advances in reward-free RL, we propose a simple meta-algorithm such that given any reward-free RL oracle, the constrained RL problems can be directly solved with negligible overheads in sample complexity.

2) Finding the minimal structural assumptions that empower sample-efficient learning is one of RL's most important research directions. This thesis advances our understanding of this fundamental question by introducing a new complexity measure—Bellman Eluder (BE) dimension. We show that the family of RL problems with low BE dimension is remarkably rich, which subsumes a vast majority of existing tractable RL problems. We further design a new optimization-based algorithm—GOLF, and provide regret and sample complexity results matching or improving the best existing results for several well-known subclasses of low BE dimension problems. Furthermore, moving towards a more challenging setting of partially observable RL, we study a new subclass of Partially Observable Markov Decision Processes (POMDPs) whose latent states can be decoded by the most recent history of a short length m. Our results show that a short-term memory suffices for reinforcement learning in these environments.

Contents

Chapter 1

Introduction

Reinforcement learning (RL) is a paradigm for sequential decision making, in which an agent learns to make decisions in an environment to accumulate reward. Many real-world settings rely on an agent's ability to make optimal sequential decisions. As a result, RL applies to many domains, including but not limited to education, healthcare, robotics, transportation, and robotics. Over the past years, we have witnessed the vast practical success of RL algorithms in various domains, along with much progress on the theoretical side. However, the theoretical understanding of challenges that underlie reinforcement learning remains somewhat limited. In this thesis, we advance our understanding by studying reinforcement learning from a primarily theoretical point of view and designing provably efficient algorithms for two challenging settings of *reinforcement learning with constraints* (Part I, Chapter 2-4) and *reinforcement learning with function approximation* (Part II, Chapter 5-6). In what follows, we describe each setting separately.

1.1 RL with Constraints (Part I)

Standard reinforcement learning (RL) approaches seek to maximize a scalar reward (Sutton and Barto, 1998, 2018; Schulman et al., 2015; Mnih et al., 2015), but in many settings this is insufficient,

because the desired properties of the agent behavior are better described using constraints. For example, an autonomous vehicle should not only get to the destination, but should also respect safety, fuel efficiency, and human comfort constraints along the way (Le et al., 2019); a robot should not only fulfill its task, but should also control its wear and tear, for example, by limiting the torque exerted on its motors (Tessler et al., 2019). Moreover, in many settings, we wish to satisfy such constraints already during *training* and not only during the *deployment*. For example, a power grid, an autonomous vehicle, or a real robotic hardware should avoid costly failures, where the hardware is damaged or humans are harmed, already during training (Leike et al., 2017; Ray et al., 2020). Constraints are also key in additional sequential decision-making applications, such as dynamic pricing with limited supply (e.g., Besbes and Zeevi, 2009; Babaioff et al., 2015), scheduling of resources on a computer cluster (Mao et al., 2016), and imitation learning, where the goal is to stay close to an expert behavior (Syed and Schapire, 2007; Ziebart et al., 2008; Sun et al., 2019b).

Most existsing work on constrained RL (Altman, 1999; Achiam et al., 2017; Tessler et al., 2019; Miryoosefi et al., 2019; Ray et al., 2020), either assume full knowledge of the model, lack theoretical guarantees, or are limited to *orthant* constraints. This naturally leads to the following question that we want to address.

Can we design provably efficient algorithms for RL with general convex constraints?

Our Contributions

- In Chapter 2, we propose an algorithmic scheme that can handle a wide class of constraints in RL tasks, specifically, any constraints that require expected values of some vector measurements (such as the use of an action) to lie in a convex set. This captures previously studied constraints (such as safety and proximity to an expert), but also enables new classes of constraints (such as diversity). Our approach comes with rigorous theoretical guarantees and only relies on the ability to approximately solve standard RL tasks. As a result, it can be easily adapted to work with any model-free or model-based RL algorithm. In our experiments, we show that it matches previous algorithms that enforce safety via constraints, but can also enforce new

properties that these algorithms cannot incorporate, such as diversity.

- in Chapter 3, We propose an algorithm for tabular episodic reinforcement learning (RL) with constraints. We provide a modular analysis with strong theoretical guarantees for two general settings. First is the convex-concave setting: maximization of a concave reward function subject to constraints that expected values of some vector quantities (such as the use of unsafe actions) lie in a convex set. Second is the knapsack setting: maximization of reward subject to the constraint that the total consumption of any of the specified resources does not exceed specified levels during the whole learning process. Previous work in constrained RL is limited to linear expectation constraints (a special case of convex-concave setting), or focuses on feasibility question, or on single-episode settings. Our experiments demonstrate that the proposed algorithm significantly outperforms these approaches in constrained episodic benchmarks.

- in Chapter 4, we bridge reward-free RL and constrained RL. Particularly, we propose a simple meta-algorithm such that given any reward-free RL oracle, the approachability and constrained RL problems can be directly solved with negligible overheads in sample complexity. Utilizing the existing reward-free RL solvers, our framework provides sharp sample complexity results for constrained RL in the tabular MDP setting, matching the best existing results up to a factor of horizon dependence; our framework directly extends to a setting of tabular two-player Markov games, and gives a new result for constrained RL with linear function approximation. Our approach isolates the challenges of constraint satisfaction, and leaves the remaining RL challenges such as learning dynamics and exploration to reward-free RL; therefore, it enables direct translation of any progress in reward-free RL to constrained RL.

1.2 RL with Function Approximation (Part II)

Modern Reinforcement Learning (RL) commonly engages practical problems with an enormous number of states, where *function approximation* must be deployed to approximate the true value function using functions from a prespecified function class. Function approximation, especially based

on deep neural networks, lies at the heart of the recent practical successes of RL in domains such as Atari (Mnih et al., 2015), Go (Silver et al., 2016), robotics (Kober et al., 2013), and dialogue systems (Li et al., 2016).

Despite its empirical success, RL with function approximation raises a new series of theoretical challenges when comparing to the classic tabular RL: (1) *generalization*, to generalize knowledge from the visited states to the unvisited states due to the enormous state space. (2) *limited expressiveness*, to handle the complicated issues where true value functions or intermediate steps computed in the algorithm can be functions outside the prespecified function class. (3) *exploration*, to address the tradeoff between exploration and exploitation when above challenges are present.

Consequently, most existing theoretical results on efficient RL with function approximation rely on relatively strong structural assumptions. For instance, many require that the MDP admits a linear approximation (Wang et al., 2019; Jin et al., 2020c; Zanette et al., 2020a), or that the model is precisely Linear Quadratic Regulator (LQR) (Anderson and Moore, 2007; Fazel et al., 2018; Dean et al., 2019). Most of these structural assumptions rarely hold in practical applications. This naturally leads to one of the most fundamental questions in RL.

What are the minimal structural assumptions that empower sample-efficient RL?

Our Contributions

- In Chapter 5, we advance our understanding of this fundamental question by introducing a new complexity measure—Bellman Eluder (BE) dimension. We show that the family of RL problems of low BE dimension is remarkably rich, which subsumes a vast majority of existing tractable RL problems including but not limited to tabular MDPs, linear MDPs, reactive POMDPs, low Bellman rank problems as well as low Eluder dimension problems. We further design a new optimization-based algorithm—GOLF, and reanalyzes a hypothesis elimination-based algorithm—OLIVE (proposed in Jiang et al., 2017). We prove that both algorithms learn the near-optimal policies of low BE dimension problems in a number of samples that is polynomial in all relevant parameters, but independent of the size of state-action space. Our

regret and sample complexity results match or improve the best existing results for several well-known subclasses of low BE dimension problems.

- In Chapter 6, we move towards a more challenging setting of *partially observable* RL as real-world sequential decision making commonly involves partial observability. Coping with partial observability in general is extremely challenging, as a number of worst-case statistical and computational barriers are known in learning Partially Observable Markov Decision Processes (POMDPs). Motivated by the problem structure in several physical applications, as well as a commonly used technique known as "frame stacking", we proposes to study a new subclass of POMDPs, whose *latent states can be decoded by the most recent history of a short length m*. We establish a set of upper and lower bounds on the sample complexity for learning near-optimal policies for this class of problems in both tabular and rich-observation settings (where the number of observations is enormous). In particular, in the rich-observation setting, we develop new algorithms using a novel "moment matching" approach with a sample complexity that scales exponentially with the short length m rather than the problem horizon, and is independent of the number of observations. Our results show that a short-term memory suffices for reinforcement learning in these environments.

1.3 Bibliographic Notes

The material presented in Chapter 2, has been published in NeurIPS 2019 (Miryoosefi et al., 2019) and is a joint work with Kiante Brantley, Hal Daume III, Miroslav Dudík, and Robert Schapire. The material presented in Chapter 3, has been published in NeurIPS 2020 (Brantley et al., 2020) and is a joint work with Kiante Brantley, Miroslav Dudik, Thodoris Lykouris, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. The material presented in Chapter 4 is a joint work with Chi Jin and is under submission (Miryoosefi and Jin, 2021). The material presented in Chapter 5, has been published in NeurIPS 2021 (Jin et al., 2021), and is a joint work with Qinghua Liu and Chi Jin. The material presented in Chapter 6 is under submission (Efroni et al., 2022) and is a joint work with Yonathan Efroni, Chi Jin, and Akshay Krishnamurthy.

Part I

Reinforcement Learning with Constraints

Chapter 2

Reinforcement Learning with Convex Constraints

2.1 Introduction

Reinforcement learning (RL) typically considers the problem of learning to optimize the behavior of an agent in an unknown environment against a single scalar reward function. For simple tasks, this can be sufficient, but for complex tasks, boiling down the learning goal into a single scalar reward can be challenging. Moreover, a scalar reward might not be a natural formalism for stating certain learning objectives, such as safety desires ("avoid dangerous situations") or exploration suggestions ("maintain a distribution over visited states that is as close to uniform as possible"). In these settings, it is much more natural to define the learning goal in terms of a vector of *measurements* over the behavior of the agent, and to learn a policy whose measurement vector is inside a target set (section 2.2).

We derive an algorithm, *approachability-based policy optimization* (APPROPO, pronounced like "apropos"), for solving such problems (section 2.3). Given a Markov decision process with vector-

valued measurements (section 2.2), and a target constraint set, ApproPO learns a stochastic policy whose expected measurements fall in that target set (akin to Blackwell approachability in single-turn games, Blackwell, 1956). We derive our algorithm from a game-theoretic perspective, leveraging recent results in online convex optimization. ApproPO is implemented as a *reduction* to any off-the-shelf reinforcement learning algorithm that can return an approximately optimal policy, and so can be used in conjunction with the algorithms that are the most appropriate for any given domain.

Our approach builds on prior work for reinforcement learning under constraints, such as the formalism of constrained Markov decision processes (CMDPs) introduced by Altman (1999). In CMDPs, the agent's goal is to maximize reward while satisfying some linear constraints over auxiliary costs (akin to our measurements). Altman (1999) gave an LP-based approach when the CMDP is fully known, and more recently, model-free approaches have been developed for CMDPs in high-dimensional settings. For instance, Achiam et al. (2017) constrained policy optimization (CPO) focuses on safe exploration and seeks to ensure approximate constraint satisfaction during the learning process. Tessler et al. (2019) reward constrained policy optimization (RCPO) follows a two-timescale primal-dual approach, giving guarantees for the convergence to a fixed point. Le et al. (2019) describe a batch off-policy algorithm with PAC-style guarantees for CMDPs using a similar game-theoretic formulation to ours.

While all of these works are only applicable to *orthant* constraints, our algorithm can work with arbitrary convex constraints. This enables ApproPO to incorporate previously studied constraint types, such as inequality constraints that represent safety or that keep the policy's behavior close to that of an expert (Syed and Schapire, 2007), as well as constraints like the aforementioned exploration suggestion, implemented as an entropy constraint on the policy's state visitation vector. The entropy of the visitation vector was recently studied as the objective by Hazan et al. (2018), who gave an algorithm capable of maximizing a concave function (e.g., entropy) over such vectors. However, it is not clear whether their approach can be adapted to the convex constraints setting studied here.

Our main contributions are: (1) a new algorithm, ApproPO, for solving reinforcement learning problems with arbitrary convex constraints; (2) a rigorous theoretical analysis that demonstrates

that it can achieve sublinear regret under mild assumptions (section 2.3); and (3) a preliminary experimental comparison with RCPO (Tessler et al., 2019), showing that our algorithm is competitive with RCPO on orthant constraints, while also handling a diversity constraint (section 2.4).

2.2 Setup and preliminaries: Defining the feasibility problem

We begin with a description of our learning setting. A *vector-valued Markov decision process* is a tuple $M = (\mathcal{S}, \mathcal{A}, \beta, P_s, P_z)$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions and β is the initial-state distribution. Each episode starts by drawing an initial state s_0 from the distribution β. Then in each step $i = 1, 2, \ldots$, the agent observes its current state s_i and takes action $a_i \in \mathcal{A}$ causing the environment to move to the next state $s_{i+1} \sim P_s(\cdot|s_i, a_i)$. The episode ends after a certain number of steps (called the horizon) or when a terminal state is reached. However, in our setting, instead of receiving a scalar reward, the agent observes a d-dimensional *measurement* vector $\mathbf{z}_i \in \mathbb{R}^d$, which, like s_{i+1}, is dependent on both the current state s_i and the action a_i, that is, $\mathbf{z}_i \sim P_z(\cdot|s_i, a_i)$. (Although not explicit in our setting, reward could be incorporated in the measurement vector.)

Typically, actions are selected according to a (stationary) policy π so that $a_i \sim \pi(s_i)$, where π maps states to distributions over actions. We assume we are working with policies from some candidate space Π. For simplicity of presentation, we assume this space is finite, though possibly extremely large. For instance, if $\mathcal{S}$ and $\mathcal{A}$ are finite, then Π might consist of all deterministic policies. (Our results hold also when Π is infinite with minor technical adjustments.)

Our aim is to control the MDP so that measurements satisfy some constraints. For any policy π, we define the *long-term measurement* $\overline{\mathbf{z}}(\pi)$ as the expected sum of discounted measurements:

$$\overline{\mathbf{z}}(\pi) \triangleq \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbf{z}_i \;\middle|\; \pi\right] \tag{2.1}$$

for some discount factor $\gamma \in [0, 1)$, and where expectation is over the random process described

above (including randomness inherent in π).

Later, we will also find it useful to consider *mixed policies* μ, which are distributions over finitely many stationary policies. The space of all such mixed policies over Π is denoted $\Delta(\Pi)$. To execute a mixed policy μ, before taking any actions, a single policy π is randomly selected according to μ; then all actions henceforth are chosen from π, for the entire episode. The long-term measurement of a mixed policy $\overline{\mathbf{z}}(\mu)$ is defined accordingly:

$$\overline{\mathbf{z}}(\mu) \triangleq \mathbb{E}_{\pi\sim\mu}\left[\overline{\mathbf{z}}(\pi)\right] = \sum_{\pi} \mu(\pi)\overline{\mathbf{z}}(\pi). \tag{2.2}$$

Our learning problem, called the *feasibility problem*, is specified by a convex *target set* $\mathcal{C}$. The goal is to find a mixed policy μ whose long-term measurements lie in the set $\mathcal{C}$:

$$\text{Feasibility Problem:} \quad \text{Find } \mu \in \Delta(\Pi) \text{ such that } \overline{\mathbf{z}}(\mu) \in \mathcal{C}. \tag{2.3}$$

For instance, in our experiments (section 2.4) we consider a grid-world environment where the measurements include the distance traveled, an indicator of hitting a rock, and indicators of visiting various locations on the grid. The feasibility goal is to achieve at most a certain trajectory length while keeping the probability of hitting the rock below a threshold for safety reasons, and maintaining a distribution over visited states close to the uniform distribution to enable exploration. We can potentially also handle settings where the goal is to maximize one measurement (e.g., "reward") subject to others by performing a binary search over the maximum attainable value of the reward (see subsection 2.3.4).

2.3 Approach, algorithm, and analysis

Before giving details of our approach, we overview the main ideas, which, to a large degree, follow the work of Abernethy et al. (2011), who considered the problem of solving two-player games; we extend these results to solve our feasibility problem (2.3).

Although feasibility is our main focus, we actually solve the stronger problem of finding a mixed policy μ that minimizes the Euclidean distance between $\overline{\mathbf{z}}(\mu)$ and $\mathcal{C}$, meaning the Euclidean distance between $\overline{\mathbf{z}}(\mu)$ and its closest point in $\mathcal{C}$. That is, we want to solve

$$\min_{\mu \in \Delta(\Pi)} \operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}) \tag{2.4}$$

where dist denotes the Euclidean distance between a point and a set.

Our main idea is to take a game-theoretic approach, formulating this problem as a game and solving it. Specifically, suppose we can express the distance function in Eq. (2.4) as a maximization of the form

$$\operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}) = \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\mu) \tag{2.5}$$

for some convex, compact set Λ.[1] Then Eq. (2.4) becomes

$$\min_{\mu \in \Delta(\Pi)} \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\mu). \tag{2.6}$$

This min-max form immediately evokes interpretation as a two-person zero-sum game: the first player chooses a mixed policy μ, the second player responds with a vector $\boldsymbol{\lambda}$, and $\boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\mu)$ is the amount that the first player is then required to pay to the second player. Assuming this game satisfies certain conditions, the final payout under the optimal play, called the *value* of the game, is the same even when the order of the players is reversed:

$$\max_{\boldsymbol{\lambda} \in \Lambda} \min_{\mu \in \Delta(\Pi)} \boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\mu). \tag{2.7}$$

Note that the policy μ we are seeking is the *solution* of this game, that is, the policy realizing the minimum in Eq. (2.6). Therefore, to find that policy, we can apply general techniques for solving a game, namely, to let a no-regret learning algorithm play the game repeatedly against a best-response

[1] Note that the distance between a point and a set is defined as a minimization of the distance function over all points in the set $\mathcal{C}$, but here we require that it be rewritten as a maximization of a linear function over some other set Λ. We will show how to achieve this in subsection 2.3.2.

player. When played in this way, it can be shown that the averages of their plays converge to the solution of the game (details in subsection 2.3.1).

In our case, we can use a no-regret algorithm for the $\boldsymbol{\lambda}$-player, and best response for the μ-player. Importantly, in our context, computing best response turns out to be an especially convenient task. Given $\boldsymbol{\lambda}$, best response means finding the mixed policy μ minimizing $\boldsymbol{\lambda} \cdot \bar{\mathbf{z}}(\mu)$. As we show below, this can be solved by treating the problem as a standard reinforcement learning task where in each step i, the agent accrues a scalar reward $r_i = -\boldsymbol{\lambda} \cdot \mathbf{z}_i$. We refer to any algorithm for solving the problem of scalar reward maximization as the *best-response oracle*. During the run of our algorithm, we invoke this oracle for different vectors $\boldsymbol{\lambda}$ corresponding to different definitions of a scalar reward. Although the oracle is only capable of solving RL tasks with a scalar reward, our algorithm can leverage this capability to solve the multi-dimensional feasibility (or distance minimization) problem.

In the remainder of this section, we provide the details of our approach, leading to our main algorithm and its analysis, and conclude with a discussion of steps for making a practical implementation. We begin by discussing game-playing techniques in general, which we then apply to our setting.

2.3.1 Solving zero-sum games using online learning

At the core of our approach, we use the general technique of Freund and Schapire (1999) for solving a game by repeatedly playing a no-regret online learning algorithm against best response.

For this purpose, we first briefly review the framework of online convex optimization, which we will soon use for one of the players: At time $t = 1, \ldots, T$, the learner makes a decision $\boldsymbol{\lambda}_t \in \Lambda$, the environment reveals a convex loss function $\ell_t : \Lambda \to \mathbb{R}$, and the learner incurs loss $\ell_t(\boldsymbol{\lambda}_t)$. The learner seeks to achieve small *regret*, the gap between its loss and the best in hindsight:

$$\text{Regret}_T \triangleq \left[\sum_{t=1}^{T} \ell_t(\boldsymbol{\lambda}_t)\right] - \min_{\boldsymbol{\lambda} \in \Lambda} \left[\sum_{t=1}^{T} \ell_t(\boldsymbol{\lambda})\right]. \tag{2.8}$$

An online learning algorithm is *no-regret* if $\text{Regret}_T = o(T)$, meaning its average loss approaches the best in hindsight. An example of such an algorithm is *online gradient descent (OGD)* of Zinkevich

(2003) (see Appendix A.1). If the Euclidean diameter of Λ is at most D, and $\|\nabla \ell_t(\boldsymbol{\lambda})\| \leq G$ for any t and $\boldsymbol{\lambda} \in \Lambda$, then the regret of OGD is at most $DG\sqrt{T}$.

Now consider a two-player zero-sum game in which two players select, respectively, $\boldsymbol{\lambda} \in \Lambda$ and $\mathbf{u} \in \mathcal{U}$, resulting in a payout of $g(\boldsymbol{\lambda}, \mathbf{u})$ from the $\mathbf{u}$-player to the $\boldsymbol{\lambda}$-player. The $\boldsymbol{\lambda}$-player wants to maximize this quantity and the $\mathbf{u}$-player wants to minimize it. Assuming g is concave in $\boldsymbol{\lambda}$ and convex in $\mathbf{u}$, if both spaces Λ and $\mathcal{U}$ are convex and compact, then the minimax theorem (von Neumann, 1928; Sion, 1958) implies that

$$\max_{\boldsymbol{\lambda} \in \Lambda} \min_{\mathbf{u} \in \mathcal{U}} g(\boldsymbol{\lambda}, \mathbf{u}) = \min_{\mathbf{u} \in \mathcal{U}} \max_{\boldsymbol{\lambda} \in \Lambda} g(\boldsymbol{\lambda}, \mathbf{u}). \tag{2.9}$$

This means that the $\boldsymbol{\lambda}$-player has an "optimal" strategy which realizes the maximum on the left and guarantees payoff of at least the *value* of the game, i.e., the value given by this expression; a similar statement holds for the $\mathbf{u}$-player.

We can solve this game (find these optimal strategies) by playing it repeatedly. We use a no-regret online learner as the $\boldsymbol{\lambda}$-player. At each time $t = 1, \ldots, T$, the learner chooses $\boldsymbol{\lambda}_t \in \Lambda$. In response, the $\mathbf{u}$-player, who in this setting is permitted knowledge of $\boldsymbol{\lambda}_t$, selects $\mathbf{u}_t$ to minimize the payout, that is, $\mathbf{u}_t = \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} g(\boldsymbol{\lambda}_t, \mathbf{u})$. This is called *best response*. The online learning algorithm is then updated by setting its loss function to be $\ell_t(\boldsymbol{\lambda}) = -g(\boldsymbol{\lambda}, \mathbf{u}_t)$. (See Algorithm 1.) As stated in Theorem 2.3.1, $\overline{\boldsymbol{\lambda}}$ and $\overline{\mathbf{u}}$, the averages of the players' decisions, converge to the solution of the game (see Appendix A.2 for the proof).

Algorithm 1 Solving a game with repeated play

input concave-convex function $g : \Lambda \times \mathcal{U} \to \mathbb{R}$, online learning algorithm LEARNER
for $t = 1$ **to** T **do**
  LEARNER makes a decision $\boldsymbol{\lambda}_t \in \Lambda$
  $\mathbf{u}_t \leftarrow \operatorname{argmin}_{\mathbf{u} \in \mathcal{U}} g(\boldsymbol{\lambda}_t, \mathbf{u})$
  LEARNER observes loss function $\ell_t(\boldsymbol{\lambda}) = -g(\boldsymbol{\lambda}, \mathbf{u}_t)$
return $\overline{\boldsymbol{\lambda}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\lambda}_t$ and $\overline{\mathbf{u}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{u}_t$

Theorem 2.3.1. *Let v be the value of the game in Eq. (2.9) and let* Regret_T *be the regret of the*

$\boldsymbol{\lambda}$-*player. Then for* $\overline{\boldsymbol{\lambda}}$ *and* $\overline{\mathbf{u}}$ *we have*

$$\min_{\mathbf{u}\in\mathcal{U}} g(\overline{\boldsymbol{\lambda}}, \mathbf{u}) \;\geq\; v - \delta \quad \textit{and} \quad \max_{\boldsymbol{\lambda}\in\Lambda} g(\boldsymbol{\lambda}, \overline{\mathbf{u}}) \;\leq\; v + \delta, \quad \textit{where } \delta = \tfrac{1}{T}\mathrm{Regret}_T. \tag{2.10}$$

2.3.2 Algorithm and main result

We can now apply this game-playing framework to the approach outlined at the beginning of this section. First, we show how to write distance as a maximization, as in Eq. (2.5). For now, we assume that our target set $\mathcal{C}$ is a *convex cone*, that is, closed under summation and also multiplication by non-negative scalars (we will remove this assumption in subsection 2.3.3). With this assumption, we can apply the following lemma (Lemma 13 of Abernethy et al., 2011), in which distance to a convex cone $\mathcal{C} \subseteq \mathbb{R}^d$ is written as a maximization over a dual convex cone $\mathcal{C}^\circ$ called the *polar cone*:

$$\mathcal{C}^\circ \triangleq \{\boldsymbol{\lambda} :\; \boldsymbol{\lambda}\cdot\mathbf{x} \leq 0 \text{ for all } \mathbf{x}\in\mathcal{C}\}. \tag{2.11}$$

Lemma 2.3.2. *For a convex cone* $\mathcal{C} \subseteq \mathbb{R}^d$ *and any point* $\mathbf{x} \in \mathbb{R}^d$

$$\mathrm{dist}(\mathbf{x}, \mathcal{C}) = \max_{\boldsymbol{\lambda}\in\mathcal{C}^\circ\cap\mathcal{B}} \boldsymbol{\lambda}\cdot\mathbf{x}, \tag{2.12}$$

where $\mathcal{B} \triangleq \{\mathbf{x} :\; \|\mathbf{x}\| \leq 1\}$ *is the Euclidean ball of radius 1 at the origin.*

Thus, Eq. (2.5) is immediately achieved by setting $\Lambda = \mathcal{C}^\circ \cap \mathcal{B}$, so the distance minimization problem (2.4) can be cast as the min-max problem (2.6). This is a special case of the zero-sum game (2.9), with $\mathcal{U} = \{\overline{\mathbf{z}}(\mu) :\; \mu \in \Delta(\Pi)\}$ and $g(\boldsymbol{\lambda}, \mathbf{u}) = \boldsymbol{\lambda}\cdot\mathbf{u}$, which can be solved with Algorithm 1. Note that the set $\mathcal{U}$ is convex and compact, because it is a linear transformation of a convex and compact set $\Delta(\Pi)$.

We will see below that the best responses $\mathbf{u}_t$ in Algorithm 1 can be expressed as $\overline{\mathbf{z}}(\pi_t)$ for some

$\pi_t \in \Pi$, and so Algorithm 1 returns

$$\overline{\mathbf{u}} = \frac{1}{T}\sum_{t=1}^{T} \overline{\mathbf{z}}(\pi_t) = \overline{\mathbf{z}}\left(\frac{1}{T}\sum_{t=1}^{T} \pi_t\right),$$

which is exactly the long-term measurement vector of the mixed policy $\bar{\mu} = \frac{1}{T}\sum_{t=1}^{T} \pi_t$. For this mixed policy, Theorem 2.3.1 immediately implies

$$\operatorname{dist}(\overline{\mathbf{z}}(\bar{\mu}), \mathcal{C}) \leq \min_{\mu \in \Delta(\Pi)} \operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}) + \tfrac{1}{T}\mathrm{Regret}_T. \tag{2.13}$$

If the problem is feasible, then $\min_{\mu\in\Delta(\Pi)} \operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}) = 0$, and since $\mathrm{Regret}_T = o(T)$, our long-term measurement $\overline{\mathbf{z}}(\bar{\mu})$ converges to the target set and solves the feasibility problem (2.3). It remains to specify how to implement the no-regret learner for the $\boldsymbol{\lambda}$-player and best response for the $\mathbf{u}$-player. We discuss these next, beginning with the latter.

The best-response player, for a given $\boldsymbol{\lambda}$, aims to minimize $\boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\mu)$ over mixed policies μ, but since this objective is linear in the mixture weights $\mu(\pi)$ (see Eq. 2.2), it suffices to minimize $\boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\pi)$ over stationary policies $\pi \in \Pi$. The key point, as already mentioned, is that this is the same as finding a policy that maximizes long-term reward in a standard reinforcement learning task if we define the scalar reward to be $r_i = -\boldsymbol{\lambda} \cdot \mathbf{z}_i$. This is because the reward of a policy π is given by

$$R(\pi) \triangleq \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r_i \;\middle|\; \pi\right] = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i (-\boldsymbol{\lambda} \cdot \mathbf{z}_i) \;\middle|\; \pi\right] = -\boldsymbol{\lambda} \cdot \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbf{z}_i \;\middle|\; \pi\right] = -\boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\pi). \tag{2.14}$$

Therefore, maximizing $R(\pi)$, as in standard RL, is equivalent to minimizing $\boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\pi)$.

Thus, best response can be implemented using any one of the many well-studied RL algorithms that maximize a scalar reward. We refer to such an RL algorithm as the *best-response oracle*. For robustness, we allow this oracle to return an approximately optimal policy.

Best-response oracle: BestResponse($\boldsymbol{\lambda}$).

> Given $\boldsymbol{\lambda} \in \mathbb{R}^d$, return a policy $\pi \in \Pi$ that satisfies $R(\pi) \geq \max_{\pi' \in \Pi} R(\pi') - \epsilon_0$, where $R(\pi)$ is the long-term reward of policy π with scalar reward defined as $r = -\boldsymbol{\lambda} \cdot \mathbf{z}$.

For the $\boldsymbol{\lambda}$-player, we do our analysis using online gradient descent (Zinkevich, 2003), an effective no-regret learner. For its update, OGD needs the gradient of the loss functions $\ell_t(\boldsymbol{\lambda}) = -\boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\pi_t)$, which is just $-\overline{\mathbf{z}}(\pi_t)$. With access to the MDP, $\overline{\mathbf{z}}(\pi)$ can be estimated simply by generating multiple trajectories using π and averaging the observed measurements. We formalize this by assuming access to an *estimation oracle* for estimating $\overline{\mathbf{z}}(\pi)$.

Estimation oracle: $\textsc{Est}(\pi)$.

Given policy π, return $\hat{\mathbf{z}}$ satisfying $\|\hat{\mathbf{z}} - \overline{\mathbf{z}}(\pi)\| \leq \epsilon_1$.

OGD also requires projection to the set $\Lambda = \mathcal{C}^\circ \cap \mathcal{B}$. In fact, if we can simply project onto the target set $\mathcal{C}$, which is more natural, then it is possible to also project onto Λ. Consider an arbitrary $\mathbf{x}$ and denote its projection onto $\mathcal{C}$ as $\Gamma_{\mathcal{C}}(\mathbf{x})$. Then the projection of $\mathbf{x}$ onto the polar cone $\mathcal{C}^\circ$ is $\Gamma_{\mathcal{C}^\circ}(\mathbf{x}) = \mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x})$ (Ingram and Marsh, 1991). Given the projection $\Gamma_{\mathcal{C}^\circ}(\mathbf{x})$ and further projecting onto $\mathcal{B}$, we obtain $\Gamma_\Lambda(\mathbf{x}) = (\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x}))/\max\{1, \|\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x})\|\}$ (because Dykstra's projection algorithm converges to this point after two steps, Boyle and Dykstra, 1986). Therefore, it suffices to require access to a *projection oracle* for $\mathcal{C}$:

Projection oracle: $\Gamma_{\mathcal{C}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{x}' \in \mathcal{C}} \|\mathbf{x} - \mathbf{x}'\|$.

Algorithm 2 ApproPO

input projection oracle $\Gamma_{\mathcal{C}}(\cdot)$ for target set $\mathcal{C}$ which is a convex cone,
best-response oracle $\textsc{BestResponse}(\cdot)$, estimation oracle $\textsc{Est}(\cdot)$,
step size η, number of iterations T
define $\Lambda \triangleq \mathcal{C}^\circ \cap \mathcal{B}$, and its projection operator $\Gamma_\Lambda(\mathbf{x}) \triangleq (\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x}))/\max\{1, \|\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x})\|\}$
initialize $\boldsymbol{\lambda}_1$ arbitrarily in Λ
for $t = 1$ **to** T **do**
  Compute an approximately optimal policy for standard RL with scalar reward $r = -\boldsymbol{\lambda}_t \cdot \mathbf{z}$:
    $\pi_t \leftarrow \textsc{BestResponse}(\boldsymbol{\lambda}_t)$
  Call the estimation oracle to approximate long-term measurement for π_t:
    $\hat{\mathbf{z}}_t \leftarrow \textsc{Est}(\pi_t)$
  Update $\boldsymbol{\lambda}_t$ using online gradient descent with the loss function $\ell_t(\boldsymbol{\lambda}) = -\boldsymbol{\lambda} \cdot \hat{\mathbf{z}}_t$:
    $\boldsymbol{\lambda}_{t+1} \leftarrow \Gamma_\Lambda\big(\boldsymbol{\lambda}_t + \eta \hat{\mathbf{z}}_t\big)$
return $\bar{\mu}$, a uniform mixture over $\pi_1, \ldots, \pi_T$

Pulling these ideas together and plugging into Algorithm 1, we obtain our main algorithm, called ApproPO (Algorithm 2), for *approachability-based policy optimization*. The algorithm provably

yields a mixed policy that approximately minimizes distance to the set $\mathcal{C}$, as shown in Theorem 2.3.3 (proved in Appendix A.3).

Theorem 2.3.3. *Assume that $\mathcal{C}$ is a convex cone and for all measurements we have $\|\mathbf{z}\| \leq B$. Suppose we run Algorithm 2 for T rounds with $\eta = \left(\frac{B}{1-\gamma} + \epsilon_1\right)^{-1} T^{-1/2}$. Then*

$$\operatorname{dist}(\overline{\mathbf{z}}(\bar{\mu}), \mathcal{C}) \leq \min_{\mu \in \Delta(\Pi)} \operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}) + \left(\tfrac{B}{1-\gamma} + \epsilon_1\right) T^{-1/2} + \epsilon_0 + 2\epsilon_1, \tag{2.15}$$

where $\bar{\mu}$ is the mixed policy returned by the algorithm.

When the goal is to solve the feasibility problem (2.3) rather than the stronger distance minimization (2.4), we can make use of a weaker reinforcement learning oracle, which only needs to find a policy that is "good enough" in the sense of providing long-term reward above some threshold:

Positive-response oracle: PosResponse($\boldsymbol{\lambda}$).

> Given $\boldsymbol{\lambda} \in \mathbb{R}^d$, return $\pi \in \Pi$ that satisfies $R(\pi) \geq -\epsilon_0$ if $\max_{\pi' \in \Pi} R(\pi') \geq 0$ (and arbitrary π otherwise), where $R(\pi)$ is the long-term reward of π with scalar reward $r = -\boldsymbol{\lambda} \cdot \mathbf{z}$.

When the problem is feasible, it can be shown that there must exist $\pi \in \Pi$ with $R(\pi) \geq 0$, and furthermore, that $\ell_t(\boldsymbol{\lambda}_t) \geq -(\epsilon_0 + \epsilon_1)$ (from Lemma A.3.1 in Appendix A.3). This means, if the goal is feasibility, we can modify Algorithm 2, replacing BestResponse with PosResponse, and adding a test at the end of each iteration to report infeasibility if $\ell_t(\boldsymbol{\lambda}_t) < -(\epsilon_0 + \epsilon_1)$. The pseudocode is provided in Algorithm 8 in Appendix A.4 along with the proof of the following convergence bound:

Theorem 2.3.4. *Assume that $\mathcal{C}$ is a convex cone and for all measurements we have $\|\mathbf{z}\| \leq B$. Suppose we run Algorithm 8 for T rounds with $\eta = \left(\frac{B}{1-\gamma} + \epsilon_1\right)^{-1} T^{-1/2}$. Then either the algorithm reports infeasibility or returns $\bar{\mu}$ such that*

$$\operatorname{dist}(\overline{\mathbf{z}}(\bar{\mu}), \mathcal{C}) \leq \left(\tfrac{B}{1-\gamma} + \epsilon_1\right) T^{-1/2} + \epsilon_0 + 2\epsilon_1. \tag{2.16}$$

2.3.3 Removing the cone assumption

Our results so far have assumed the target set $\mathcal{C}$ is a convex cone. If instead $\mathcal{C}$ is an arbitrary convex, compact set, we can use the technique of Abernethy et al. (2011) and apply our algorithm to a specific convex cone $\tilde{\mathcal{C}}$ constructed from $\mathcal{C}$ to obtain a solution with provable guarantees.

In more detail, given a compact, convex target set $\mathcal{C} \subseteq \mathbb{R}^d$, we augment every vector in $\mathcal{C}$ with a new coordinate held fixed to some value $\kappa > 0$, and then let $\tilde{\mathcal{C}}$ be its conic hull. Thus,

$$\tilde{\mathcal{C}} = \text{cone}(\mathcal{C} \times \{\kappa\}), \qquad \text{where cone}(\mathcal{X}) = \{\alpha \mathbf{x} \mid \mathbf{x} \in \mathcal{X}, \alpha \geq 0\}. \tag{2.17}$$

Given our original vector-valued MDP $M = (\mathcal{S}, \mathcal{A}, \beta, P_s, P_z)$, we define a new MDP $M' = (\mathcal{S}, \mathcal{A}, \beta, P_s, P'_{z'})$ with $(d+1)$-dimensional measurement $\mathbf{z}' \in \mathbb{R}^{d+1}$, defined (and generated) by

$$\mathbf{z}'_i = \mathbf{z}_i \oplus \langle (1-\gamma)\kappa \rangle, \qquad \mathbf{z}_i \sim P_z(\cdot \mid s_i, a_i) \tag{2.18}$$

where $\oplus$ denotes vector concatenation. Writing long-term measurement for M and M' as $\overline{\mathbf{z}}$ and $\overline{\mathbf{z}}'$ respectively, $\overline{\mathbf{z}}'(\pi) = \overline{\mathbf{z}}(\pi) \oplus \langle \kappa \rangle$, for any policy $\pi \in \Pi$, and similarly for any mixed policy μ.

The main idea is to apply the algorithms described above to the modified MDP M' using the cone $\tilde{\mathcal{C}}$ as target set. For an appropriate choice of $\kappa > 0$, we show that the resulting mixed policy will approximately minimize distance to $\mathcal{C}$ for the original MDP M. This is a consequence of the following lemma, an extension of Lemma 14 of Abernethy et al. (2011), which shows that distances are largely preserved in a controllable way under this construction. The proof is in Appendix A.5.

Lemma 2.3.5. *Consider a compact, convex set $\mathcal{C}$ in $\mathbb{R}^d$ and $\mathbf{x} \in \mathbb{R}^d$. For any $\delta > 0$, let $\tilde{\mathcal{C}} = \text{cone}(\mathcal{C} \times \{\kappa\})$, where $\kappa = \frac{\max_{\mathbf{x} \in \mathcal{C}} \|\mathbf{x}\|}{\sqrt{2\delta}}$. Then $\text{dist}(\mathbf{x}, \mathcal{C}) \leq (1+\delta)\text{dist}(\mathbf{x} \oplus \langle \kappa \rangle, \tilde{\mathcal{C}})$.*

Corollary 2.3.6. *Assume that $\mathcal{C}$ is a convex, compact set and for all measurements we have $\|\mathbf{z}\| \leq B$. Then by putting $\eta = \left(\frac{B+\kappa}{1-\gamma} + \epsilon_1\right)^{-1} T^{-1/2}$ and running Algorithm 2 for T rounds with M'*

as the MDP and $\tilde{\mathcal{C}}$ as the target set, the mixed policy $\bar{\mu}$ returned by the algorithm satisfies

$$\operatorname{dist}(\overline{\mathbf{z}}(\bar{\mu}), \mathcal{C}) \leq (1+\delta)\left(\min_{\mu\in\Delta(\Pi)} \operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}) + \left(\tfrac{B+\kappa}{1-\gamma} + \epsilon_1\right)T^{-1/2} + \epsilon_0 + 2\epsilon_1\right), \tag{2.19}$$

where $\kappa = \frac{\max_{\mathbf{x}\in\mathcal{C}} \|\mathbf{x}\|}{\sqrt{2\delta}}$ for an arbitrary $\delta > 0$. Similarly for Algorithm 8, we either have

$$\operatorname{dist}(\overline{\mathbf{z}}(\bar{\mu}), \mathcal{C}) \leq (1+\delta)\left(\left(\tfrac{B+\kappa}{1-\gamma} + \epsilon_1\right)T^{-1/2} + \epsilon_0 + 2\epsilon_1\right) \tag{2.20}$$

or the algorithm reports infeasibility.

2.3.4 Practical implementation of the positive response and estimation oracles

We next briefly describe a few techniques for the practical implementation of our algorithm.

As discussed in subsection 2.3.2, when our aim is to solve a feasibility problem, we only need access to a positive response oracle. In episodic environments, it is straightforward to use any standard iterative RL approach as a positive response oracle: As the RL algorithm runs, we track its accrued rewards, and when the trailing average of the last n trajectory-level rewards goes above some level $-\epsilon$, we return the current policy (possibly specified implicitly as a Q-function).[2] Furthermore, the average of the measurement vectors $\mathbf{z}$ collected over the last n trajectories can serve as the estimate $\hat{\mathbf{z}}_t$ of the long-term measurement required by the algorithm, side-stepping the need for an additional estimation oracle.

The hyperparameters ϵ and n influence the oracle quality; specifically, assuming that the rewards are bounded and the overall number of trajectories until the oracle terminates is at most polynomial in n, we have $\epsilon_0 = \epsilon - O(\sqrt{(\log n)/n})$ and $\epsilon_1 = O(\sqrt{(\log n)/n})$. In principle, we could use Theorem 2.3.4 to select a value T at which to stop; in practice, we run until the running average of the measurements $\hat{\mathbf{z}}_t$ gets within a small distance of the target set $\mathcal{C}$. If the RL algorithm runs for too long without achieving

[2] This assumes that the last n trajectories accurately estimate the performance of the final iterate. If that is not the case, the oracle can instead return the mixture of the policies corresponding to the last n iterates.

non-negative rewards, we stop and declare that the underlying problem is "empirically infeasible." (Actual infeasibility would hold if it is truly not possible to reach non-negative expected reward.)

An important mechanism to further speed up our algorithm is to maintain a "cache" of all the policies returned by the positive response oracle so far. Each of the cached policies π is stored with the estimate of its expected measurement vector $\hat{\mathbf{z}}(\pi) \approx \bar{\mathbf{z}}(\pi)$, based on its last n iterations (as above). In each outer-loop iteration of our algorithm, we first check if the cache contains a policy that already achieves a reward at least $-\epsilon$ under the new $\boldsymbol{\lambda}$; this can be determined from the cached $\hat{\mathbf{z}}(\pi)$ since the reward is just a linear function of the measurement vector. If such a policy is found, we return it, alongside $\hat{\mathbf{z}}(\pi)$, instead of calling the oracle. Otherwise, we pick the policy from the cache with the largest reward (below $-\epsilon$ by assumption) and use it to warm-start the RL algorithm implementing the oracle. The cache can be initialized with a few random policies (as we do in our experiments), effectively implementing randomized weight initialization.

The cache interacts well with a straightforward binary-search scheme that can be used when the goal is to maximize some reward (possibly subject to additional constraints), rather than only satisfy a set of constraints. The feasibility problems corresponding to iterates of binary search only differ in the constraint values, but use the same measurements, so the same cache can be reused across all iterations.

Running time. Note that APPROPO spends the bulk of its running time executing the best-response oracle. It additionally performs updates of $\boldsymbol{\lambda}$, but these tend to be orders of magnitude cheaper than any per-episode (or per-transition) updates within the oracle. For example, in our experiments (section 2.4), the dimension of $\boldsymbol{\lambda}$ is either 2 or 66 (without or with the diversity constraint, respectively), whereas the policies π trained by the oracle are two-layer networks described by 8,704 floating-point numbers.

2.4 Experiments

We next evaluate the performance of APPROPO and demonstrate its ability to handle a variety of constraints. For simplicity, we focus on the feasibility version (Algorithm 8 in Appendix A.4).

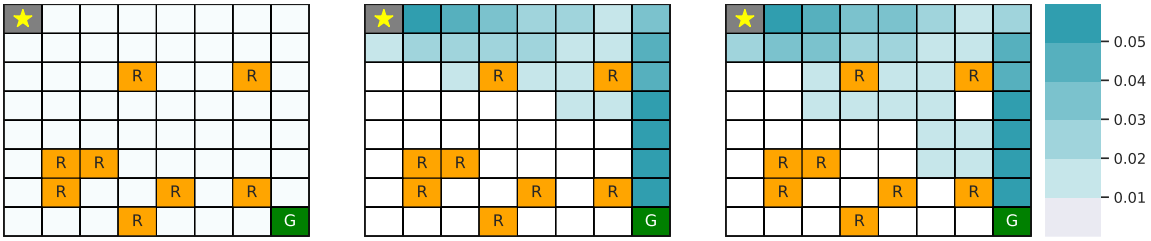


Figure 2.1: *Left:* The Mars rover environment. The agent starts in top-left and needs to reach the goal in bottom-right while avoiding rocks. *Middle, Right:* Visitation probabilities of ApproPO (middle) and ApproPO with a diversity constraints (right) at 12k samples. Both plots based on a single run.

We compare ApproPO with the RCPO approach of Tessler et al. (2019), which adapts policy gradient, specifically, asynchronous actor-critic (A2C) (Mnih et al., 2016), to find a fixed point of the Lagrangian of the constrained policy optimization problem. RCPO maintains and updates a vector of Lagrange multipliers, which is then used to derive a reward for A2C. The vector of Lagrange multipliers serves a similar role as our $\boldsymbol{\lambda}$, and the overall structure of RCPO is similar to ApproPO, so RCPO is a natural baseline for a comparison. Unlike ApproPO, RCPO only allows orthant constraints and it seeks to maximize reward, whereas ApproPO solves the feasibility problem.

For a fair comparison, ApproPO uses A2C as a positive-response oracle, with the same hyperparameters as used in RCPO. Online learning in the outer loop of ApproPO was implemented via online gradient descent with momentum. Both RCPO and ApproPO have an outer-loop learning rate parameter, which we tuned over a grid of values 10^{-i} with integer i (see Appendix A.6 for the details). Here, we report the results with the best learning rate for each method.

We ran our experiments on a small version of the *Mars rover* grid-world environment, used previously for the evaluation of RCPO (Tessler et al., 2019). In this environment, depicted in Figure 2.1 (left), the agent must move from the starting position to the goal without crashing into rocks. The episode terminates when a rock or the goal is reached, or after 300 steps. The environment is stochastic: with probability $\delta = 0.05$ the agent's action is perturbed to a random action. The agent receives small negative reward each time step and zero for terminating, with $\gamma = 0.99$. We used the same safety constraint as Tessler et al. (2019): ensure that the (discounted) probability of hitting a rock is at most a fixed threshold (set to 0.2). RCPO seeks to maximize reward subject to this constraint.

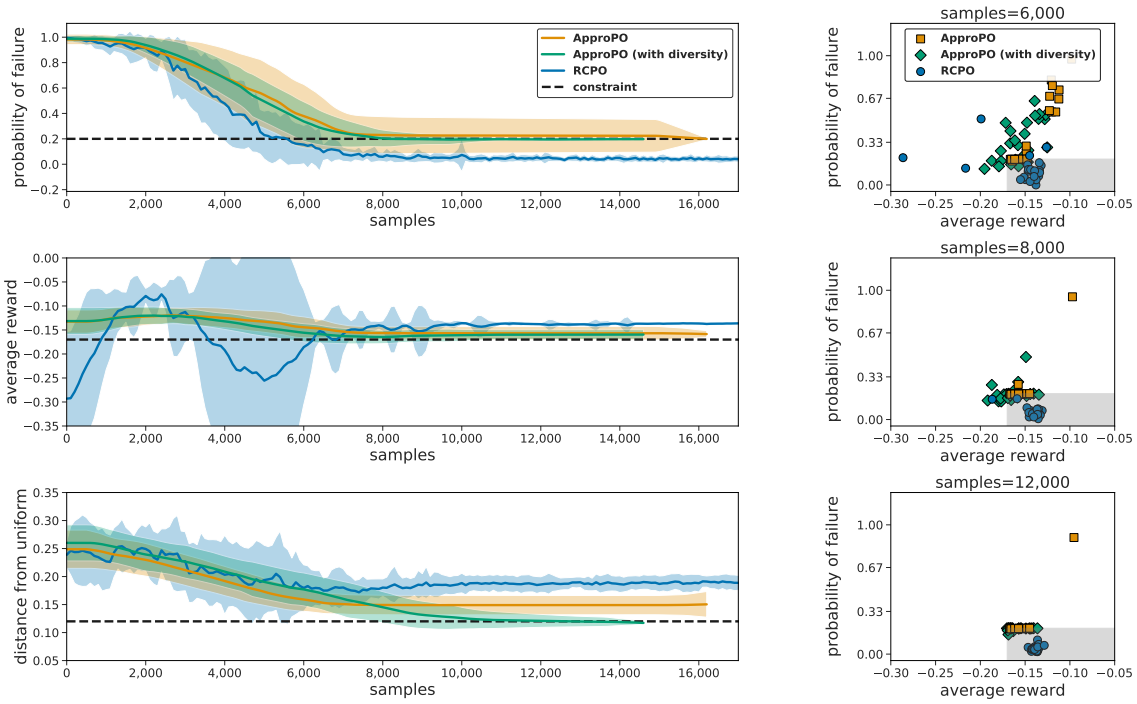


Figure 2.2: *Left:* The performance of the algorithms as a function of the number of samples (steps in the environment); showing average and standard deviation over 25 runs. The vertical axes correspond to the three constraints, with thresholds shown as a dashed line; for reward (middle) this is a lower bound; for the others it is an upper bound. *Right:* Each point in the scatter plot represents the reward and the probability of failure obtained by the policy learnt by the algorithm at the specified number of samples. The grey region is the target set. Different points represent different random runs.

ApproPO solves a feasibility problem with the same safety constraint, and an additional constraint requiring that the reward be at least -0.17 (this is slightly lower than the final reward achieved by RCPO). We also experimented with including the exploration suggestion as a "diversity constraint," requiring that the Euclidean distance between our visitation probability vector (across the cells of the grid) and the uniform distribution over the upper-right triangle cells of the grid (excluding rocks) be at most 0.12.[3]

In Figure 2.2 (left), we show how the probability of failure, the average reward, and the distance to the uniform distribution over upper triangle vary as a function of the number of samples seen by each algorithm. Both variants of our algorithm are able to satisfy the safety constraints and reach similar reward as RCPO with a similar number of samples (around 8k samples). Furthermore, including the diversity constraint, which RCPO is not capable of enforcing, allowed our method to reach a more diverse policy as depicted in both Figure 2.2 (bottom-left) and Figure 2.1 (right).

2.5 Conclusion

In this paper, we introduced ApproPO, an algorithm for solving reinforcement learning problems with arbitrary convex constraints. ApproPO can combine any no-regret online learner with any standard RL algorithm that optimizes a scalar reward. Theoretically, we showed that for the specific case of online gradient descent, ApproPO learns to approach the constraint set at a rate of $1/\sqrt{T}$, with an additive non-vanishing term that measures the optimality gap of the reinforcement learner. Experimentally, we demonstrated that ApproPO can be applied with well-known RL algorithms for discrete domains (like actor-critic), and achieves similar performance as RCPO (Tessler et al., 2019), while being able to satisfy additional types of constraints. In sum, this yields a theoretically justified, practical algorithm for solving the approachability problem in reinforcement learning.

[3] This number ensures that ApproPO without the diversity constraint does not satisfy it automatically.

Chapter 3

Constrained Episodic RL in Concave-Convex and Knapsack Settings

3.1 Introduction

Standard reinforcement learning (RL) approaches seek to maximize a scalar reward (Sutton and Barto, 1998, 2018; Schulman et al., 2015; Mnih et al., 2015), but in many settings this is insufficient, because the desired properties of the agent behavior are better described using constraints. For example, an autonomous vehicle should not only get to the destination, but should also respect safety, fuel efficiency, and human comfort constraints along the way (Le et al., 2019); a robot should not only fulfill its task, but should also control its wear and tear, for example, by limiting the torque exerted on its motors (Tessler et al., 2019). Moreover, in many settings, we wish to satisfy such constraints already during *training* and not only during the *deployment*. For example, a power grid, an autonomous vehicle, or a real robotic hardware should avoid costly failures, where the hardware

is damaged or humans are harmed, already during training (Leike et al., 2017; Ray et al., 2020). Constraints are also key in additional sequential decision-making applications, such as dynamic pricing with limited supply (e.g., Besbes and Zeevi, 2009; Babaioff et al., 2015), scheduling of resources on a computer cluster (Mao et al., 2016), and imitation learning, where the goal is to stay close to an expert behavior (Syed and Schapire, 2007; Ziebart et al., 2008; Sun et al., 2019b).

In this chapter we study *constrained episodic reinforcement learning*, which encompasses all of these applications. An important characteristic of our approach, distinguishing it from previous work (e.g., Altman, 1999; Achiam et al., 2017; Tessler et al., 2019; Miryoosefi et al., 2019; Ray et al., 2020), is our focus on *efficient exploration*, leading to reduced sample complexity. Notably, the modularity of our approach enables extensions to more complex settings such as (i) maximizing concave objectives under convex constraints, and (ii) reinforcement learning under hard constraints, where the learner has to stop when some constraint is violated (e.g., a car runs out of gas). For these extensions, which we refer to as *concave-convex setting* and *knapsack setting*, we provide the first regret guarantees in the episodic setting (see related work below for a detailed comparison). Moreover, our guarantees are *anytime*, meaning that the constraint violations are bounded at any point during learning, even if the learning process is interrupted. This is important for those applications where the system continues to learn after it is deployed.

Our approach relies on the principle of *optimism under uncertainty* to efficiently explore. Our learning algorithms optimize their actions with respect to a model based on the empirical statistics, while optimistically overestimating rewards and underestimating the resource consumption (i.e., overestimating the distance from the constraint). This idea was previously introduced in multi-armed bandits (Agrawal and Devanur, 2014); extending it to episodic reinforcement learning poses additional challenges since the policy space is exponential in the episode horizon. Circumventing these challenges, we provide a modular way to analyze this approach in the basic setting where both rewards and constraints are linear (Section 3.3) and then transfer this result to the more complicated concave-convex and knapsack settings (Sections 3.4 and 3.5). We empirically compare our approach with the only previous works that can handle convex constraints and show that our algorithmic

innovations lead to significant empirical improvements (Section 3.6).

Related work. Sample-efficient exploration in constrained episodic reinforcement learning has only recently started to receive attention. Most previous works on episodic reinforcement learning focus on unconstrained settings (Jaksch et al., 2010; Azar et al., 2017; Dann et al., 2017). A notable exception is the work of Cheung (2019) and Tarbouriech and Lazaric (2019). Both of these works consider vectorial feedback and aggregate reward functions, and provide theoretical guarantees for the reinforcement learning setting with a single episode, but require a strong reachability or communication assumption, which is not needed in the episodic setting studied here. Also, compared to Cheung (2019), our results for the knapsack setting allow for a significantly smaller budget, as we illustrate in Section 3.5. Moreover, our approach is based on a tighter bonus, which leads to a superior empirical performance (see Section 3.6). Recently, there have also been several concurrent and independent works on sample-efficient exploration for reinforcement learning with constraints (Singh et al., 2020; Efroni et al., 2020; Qiu et al., 2020; Ding et al., 2021; Zheng and Ratliff, 2020). Unlike our work, all of these approaches focus on linear reward objective and linear constraints and do not handle the concave-convex and knapsack settings that we consider.

Constrained reinforcement learning has also been studied in settings that do not focus on sample-efficient exploration (Achiam et al., 2017; Tessler et al., 2019; Miryoosefi et al., 2019). Among these, only Miryoosefi et al. (2019) handle convex constraints, albeit without a reward objective (they solve the feasibility problem). Since these works do not focus on sample-efficient exploration, their performance drastically deteriorates when the task requires exploration (as we show in Section 3.6).

Sample-efficient exploration under constraints has been studied in multi-armed bandits, starting with a line of work on dynamic pricing with limited supply (Besbes and Zeevi, 2009, 2011; Babaioff et al., 2015; Wang et al., 2014). A general setting for bandits with global knapsack constraints (*bandits with knapsacks*) was defined and solved by Badanidiyuru et al. (2018) (see also Ch. 10 of Slivkins, 2019). Within this literature, the closest to ours is the work of Agrawal and Devanur (2014), who study bandits with concave objectives and convex constraints. Our work is directly inspired by theirs and lifts their techniques to the more general episodic reinforcement learning

setting.

3.2 Model and preliminaries

In episodic reinforcement learning, a learner repeatedly interacts with an environment across K episodes. The environment includes the state space $\mathcal{S}$, the action space $\mathcal{A}$, the episode horizon H, and the initial state s_0.[1] To capture constrained settings, the environment includes a set $\mathcal{D}$ of d resources where each $i \in \mathcal{D}$ has a capacity constraint $\xi(i) \in \mathbb{R}^+$. The above are fixed and known to the learner.

Constrained Markov decision process. We work with MDPs that have resource consumption in addition to rewards. Formally, a *constrained* MDP (CMDP) is a triple $\mathcal{M} = (p, r, \mathbf{c})$ that describes transition probabilities $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, rewards $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$, and resource consumption $\mathbf{c} : \mathcal{S} \times \mathcal{A} \to [0, 1]^d$. For convenience, we denote $c(s, a, i) = c_i(s, a)$. We allow stochastic rewards and consumptions, in which case r and $\mathbf{c}$ refer to the conditional expectations, conditioned on s and a (our definitions and algorithms are based on this conditional expectation rather than the full conditional distribution).

We use the above definition to describe two kinds of CMDPs. The *true* CMDP $\mathcal{M}^\star = (p^\star, r^\star, \mathbf{c}^\star)$ is fixed but *unknown* to the learner. Selecting action a at state s results in rewards and consumptions drawn from (possibly correlated) distributions with means $r^\star(s, a)$ and $\mathbf{c}^\star(s, a)$ and supports in $[0, 1]$ and $[0, 1]^d$ respectively. Next states are generated from transition probabilities $p^\star(s, a)$. The second kind of CMDP arises in our algorithm, which is model-based and at episode k uses a CMDP $\mathcal{M}^{(k)}$.

Episodic reinforcement learning protocol. At episode $k \in [K]$, the learner commits to a policy $\pi_k = (\pi_{k,h})_{h=1}^H$ where $\pi_{k,h} : \mathcal{S} \to \Delta(\mathcal{A})$ specifies how to select actions at step h for every state. The learner starts from state $s_{k,1} = s_0$. At step $h = 1, \ldots, H$, she selects an action $a_{k,h} \sim \pi_{k,h}(s_{k,h})$. The learner earns reward $r_{k,h}$ and suffers consumption $\mathbf{c}_{k,h}$, both drawn from the true CMDP $\mathcal{M}^\star$

[1] A fixed and known initial state is without loss of generality. In general, there is a fixed but unknown distribution ρ from which the initial state is drawn before each episode. We modify the MDP by adding a new state s_0 as initial state, such that the next state is sampled from ρ for any action. Then ρ is "included" within the transition probabilities. The extra state s_0 does not contribute any reward and does not consume any resources.

on state-action pair $(s_{k,h}, a_{k,h})$ as described above, and transitions to state $s_{k,h+1} \sim p^\star(s_{k,h}, a_{k,h})$.

Objectives. In the basic setting (Section 3.3), the learner wishes to maximize reward while respecting the consumption constraints in expectation by competing favorably against the following benchmark:

$$\max_{\pi} \mathbb{E}^{\pi,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h, a_h\big)\Big] \qquad \text{s.t.} \qquad \forall i \in \mathcal{D} : \mathbb{E}^{\pi,p^\star}\Big[\sum_{h=1}^{H} c^\star\big(s_h, a_h, i\big)\Big] \le \xi(i), \tag{3.1}$$

where $\mathbb{E}^{\pi,p}$ denotes the expectation over the run of policy π according to transitions p, and s_h, a_h are the induced random state-action pairs. We denote by $\pi^\star$ the policy that maximizes this objective.

For the basic setting, we track two performance measures: *reward regret* compares the learner's total reward to the benchmark and *consumption regret* bounds excess in resource consumption:

$$\begin{aligned} \textsc{RewReg}(k) &:= \mathbb{E}^{\pi^\star,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h, a_h\big)\Big] - \frac{1}{k}\sum_{t=1}^{k} \mathbb{E}^{\pi_t,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h, a_h\big)\Big], \\ \textsc{ConsReg}(k) &:= \max_{i\in\mathcal{D}} \Big(\frac{1}{k}\sum_{t=1}^{k} \mathbb{E}^{\pi_t,p^\star}\Big[\sum_{h=1}^{H} c^\star\big(s_h, a_h, i\big)\Big] - \xi(i)\Big). \end{aligned} \tag{3.2}$$

Our guarantees are *anytime*, i.e., they hold at any episode k and not only after the last episode.

We also consider two extensions. In Section 3.4, we consider a concave reward objective and convex consumption constraints. In Section 3.5, we require consumption constraints to be satisfied with high probability under a cumulative budget across all K episodes, rather than in expectation in a single episode.

Tabular MDPs. We assume that the state space $\mathcal{S}$ and the action space $\mathcal{A}$ are finite (tabular setting). We construct standard empirical estimates separately for each state-action pair (s, a), using the learner's observations up to and not including a given episode k. Eqs. (3.3) define sample

counts, empirical transition probabilities, empirical rewards, and empirical resource consumption.[2]

$$
\begin{aligned}
N_k(s,a) &= \max\left\{1, \sum_{t\in[k-1],\, h\in[H]} \mathbf{1}\{s_{t,h}=s, a_{t,h}=a\}\right\},\\
\widehat{p}_k(s'|s,a) &= \frac{1}{N_k(s,a)} \sum_{t\in[k-1],\, h\in[H]} \mathbf{1}\{s_{t,h}=s, a_{t,h}=a, s_{t,h+1}=s'\},\\
\widehat{r}_k(s,a) &= \frac{1}{N_k(s,a)} \sum_{t\in[k-1],\, h\in[H]} r_{t,h}\cdot\mathbf{1}\{s_{t,h}=s, a_{t,h}=a\},\\
\widehat{c}_k(s,a,i) &= \frac{1}{N_k(s,a)} \sum_{t\in[k-1],\, h\in[H]} c_{t,h,i}\cdot\mathbf{1}\{s_{t,h}=s, a_{t,h}=a\} \quad \forall i\in\mathcal{D}.
\end{aligned}
\tag{3.3}
$$

Preliminaries for theoretical analysis. The *Q-function* is a standard object in RL that tracks the learner's expected performance if she starts from state $s \in \mathcal{S}$ at step h, selects action $a \in \mathcal{A}$, and then follows a policy π under a model with transitions p for the remainder of the episode. We parameterize it by the *objective function* $m : \mathcal{S}\times\mathcal{A} \to [0,1]$, which can be either a reward, i.e., $m(s,a) = r(s,a)$, or consumption of some resource $i \in \mathcal{D}$, i.e., $m(s,a) = c(s,a,i)$. (For the unconstrained setting, the objective is the reward.) The performance of the policy in a particular step h is evaluated by the value function V which corresponds to the expected Q-function of the selected action (where the expectation is taken over the possibly randomized action selection of π). The Q and value functions can be both recursively defined by dynamic programming:

$$
\begin{aligned}
Q_m^{\pi,p}(s,a,h) &= m(s,a) + \sum_{s'\in\mathcal{S}} p(s'|s,a) V_m^{\pi,p}(s',h+1),\\
V_m^{\pi,p}(s,h) &= \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q_m^{\pi,p}(s,a,h)\right] \quad \text{and} \quad V_m^{\pi,p}(s,H+1) = 0.
\end{aligned}
$$

By slight abuse of notation, for $m \in \{r\}\cup\{c_i\}_{i\in\mathcal{D}}$, we denote by $m^\star \in \{r^\star\}\cup\{c_i^\star\}_{i\in\mathcal{D}}$ the corresponding objectives with respect to the rewards and consumptions of the true CMDP $\mathcal{M}^\star$. For objectives $m^\star$ and transitions $p^\star$, the above are the *Bellman equations* of the system (Bellman, 1957).

[2]The max operator in Eq. Eq. (3.3) is to avoid dividing by 0.

Estimating the Q-function based on the model parameters p and m rather than the ground truth parameters $p^\star$ and $m^\star$ introduces errors. These errors are localized across stages by the notion of *Bellman error* which contrasts the performance of policy π starting from stage h under the model parameters to a benchmark that behaves according to the model parameters starting from the next stage $h+1$ but uses the true parameters of the system in stage h. More formally, for objective m:

$$\textsc{Bell}_m^{\pi,p}(s,a,h) = Q_m^{\pi,p}(s,a,h) - \Big(m^\star(s,a) + \sum_{s'\in\mathcal{S}} p^\star(s'|s,a) V_m^{\pi,p}(s',h+1)\Big). \tag{3.4}$$

Note that when the cMDP is $\mathcal{M}^\star$ ($m = m^\star$, $p = p^\star$), there is no mismatch and $\textsc{Bell}_{m^\star}^{\pi,p^\star} = 0$.

3.3 Warm-up algorithm and analysis in the basic setting

In this section, we introduce a simple algorithm that allows to simultaneously bound reward and consumption regrets for the basic setting introduced in the previous section. Even in this basic setting, we provide the first sample-efficient guarantees in constrained episodic reinforcement learning.[3] The modular analysis of the guarantees also allows us to subsequently extend (in Sections 3.4 and 3.5) the algorithm and guarantees to the more general concave-convex and knapsack settings.

Our algorithm. At episode k, we construct an estimated cMDP $\mathcal{M}^{(k)} = \big(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)}\big)$ based on the observations collected so far. The estimates are *bonus-enhanced* (formalized below) to encourage more targeted exploration. Our algorithm ConRL selects a policy π_k by solving the following constrained optimization problem which we refer to as $\textsc{BasicConPlanner}(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)})$:

$$\max_\pi \mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^H r^{(k)}\big(s_h,a_h\big)\Big] \qquad \text{s.t.} \qquad \forall i\in\mathcal{D}: \mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^H c^{(k)}\big(s_h,a_h,i\big)\Big] \le \xi(i).$$

The above optimization problem is similar to the objective Eq. (3.1) but uses the estimated model instead of the (unknown to the learner) true model. We also note that this optimization problem can be optimally solved as it is a linear program on the occupation measures (Puterman, 2014),

[3] We refer the reader to the related work (in Section 3.1) for discussion on concurrent and independent papers. Unlike our results, these papers do not extend to either concave-convex or knapsack settings.

i.e., setting as variables the probability of each state-action pair and imposing flow conservation constraints with respect to the transitions. This program is described in Appendix B.1.1.

Bonus-enhanced model. A standard approach to implement the principle of optimism under uncertainty is to introduce, at each episode k, a *bonus term* $\widehat{b}_k(s,a)$ that favors under-explored actions. Specifically, we add this bonus to the empirical rewards Eq. (3.3), and subtract it from the consumptions Eq. (3.3): $r^{(k)}(s,a) = \widehat{r}_k(s,a) + \widehat{b}_k(s,a)$ and $c^{(k)}(s,a,i) = \widehat{c}_k(s,a,i) - \widehat{b}_k(s,a)$ for each resource i.

Similar to unconstrained analogues (Azar et al., 2017; Dann et al., 2017), we define the bonus as:

$$\widehat{b}_k(s,a) = \min\left\{2H,\ H\sqrt{\frac{2\ln\big(8SAH(d+1)k^2/\delta\big)}{N_k(s,a)}}\right\}, \tag{3.5}$$

where $\delta > 0$ is the desired failure probability of the algorithm and $N_k(s,a)$ is the number of times (s,a) pair is visited, c.f. Eq. (3.3), $S = |\mathcal{S}|$, and $A = |\mathcal{A}|$. Thus, under-explored actions have a larger bonus, and therefore appear more appealing to the planner. For estimated transition probabilities, we just use the empirical averages Eq. (3.3): $p^{(k)}(s'|s,a) = \widehat{p}(s'|s,a)$.

Valid bonus and Bellman-error decomposition. For a bonus-enhanced model to achieve effective exploration, the resulting bonuses need to be *valid*, i.e., they should ensure that the estimated rewards overestimate the true rewards and the estimated consumptions underestimate the true consumptions.

Definition 3.3.1. A bonus $b_k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is valid if, $\forall s \in \mathcal{S}, a \in \mathcal{A}, h \in [H], m \in \{r\} \cup \{c_i\}_{i\in\mathcal{D}}$:

$$\left|\Big(\widehat{m}_k(s,a) - m^\star(s,a)\Big) + \sum_{s'\in\mathcal{S}}\Big(\widehat{p}_k(s'|s,a) - p^\star(s'|s,a)\Big)V_{m^\star}^{\pi^\star,p^\star}(s',h+1)\right| \le b_k(s,a).$$

By classical concentration bounds (Appendix B.2.1), the bonus $\widehat{b}_k$ of Eq. Eq. (3.5) satisfies this condition:

Lemma 3.3.2. *With probability $1-\delta$, the bonus $\widehat{b}_k(s,a)$ is valid for all episodes k simultaneously.*

Our algorithm solves the BasicConPlanner optimization problem based on a bonus-enhanced model. When the bonuses are valid, we can upper bound the per-episode regret by the expected sum of Bellman errors across steps. This is the first part in classical unconstrained analyses and the following proposition extends this decomposition to constrained episodic reinforcement learning. The proof uses the so-called simulation lemma (Kearns and Singh, 2002) and is provided in Appendix B.2.3.

Proposition 3.3.3. *If $\widehat{b}_k(s,a)$ is valid for all episodes k simultaneously then the per-episode reward and consumption regrets can be upper bounded by the expected sum of Bellman errors Eq. (3.4):*

$$\begin{aligned} &\mathbb{E}^{\pi^\star,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h,a_h\big)\Big] - \mathbb{E}^{\pi_k,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h,a_h\big)\Big] \le \mathbb{E}^{\pi_k}\Big[\sum_{h=1}^{H}\Big|\textsc{Bell}^{\pi_k,p^{(k)}}_{r^{(k)}}\big(s_h,a_h,h\big)\Big|\Big] \\ &\forall i\in\mathcal{D}: \qquad \mathbb{E}^{\pi_k,p^\star}\Big[\sum_{h=1}^{H} c^\star\big(s_h,a_h,i\big)\Big] - \xi(i) \le \mathbb{E}^{\pi_k}\Big[\sum_{h=1}^{H}\Big|\textsc{Bell}^{\pi_k,p^{(k)}}_{c_i^{(k)}}\big(s_h,a_h,h\big)\Big|\Big]. \end{aligned} \tag{3.6}$$

Final guarantee. One difficulty with directly bounding the Bellman error is that the value function is not independent of the draws forming $r^{(k)}(s,a)$, $\mathbf{c}^{(k)}(s,a)$, and $p^{(k)}(s'|s,a)$. Hence we cannot apply Hoeffding inequality directly. While Azar et al. (2017) propose a trick to get an $\mathcal{O}(\sqrt{S})$ bound on Bellman error in unconstrained settings, the trick relies on the crucial property of Bellman optimality: for an unconstrained MDP, its optimal policy $\pi^\star$ satisfies the condition, $V^{\pi^\star}_{r^\star}(s,h) \ge V^{\pi}_{r^\star}(s,h)$ for all s,h,π (i.e., $\pi^\star$ is optimal at any state). However, when constraints exist, the optimal policy does not satisfy the Bellman optimality property. Indeed, we can only guarantee optimality with respect to the initial state distribution, i.e., $V^{\pi^\star}_{r^\star}(s_0,1) \ge V^{\pi}_{r^\star}(s_0,1)$ for any π, but not everywhere else. This illustrates a fundamental difference between constrained MDPs and unconstrained MDPs. Thus we cannot directly apply the trick from Azar et al. (2017). Instead we follow an alternative approach of bounding the value function via an ϵ-net over the possible values. This analysis leads to a guarantee that is weaker by a factor of $\sqrt{S}$ than the unconstrained results. The proof is provided in Appendix B.2.6.

Theorem 3.3.4. *There exists an absolute constant $c\in\mathbb{R}^+$ such that, with probability at least $1-3\delta$,*

reward and consumption regrets are both upper bounded by:

$$\frac{c}{\sqrt{k}} \cdot H^{2.5} S\sqrt{A} \cdot \sqrt{\ln(k) \ln\big(SAH(d+1)k/\delta\big)} + \frac{c}{k} \cdot S^{3/2} AH^3 \sqrt{\ln\big(2SAH(d+1)k/\delta\big)}.$$

Comparison to single-episode results. In single-episode setting, Cheung (2019) achieves $\sqrt{S}$ dependency under the further assumption that the transitions are sparse, i.e., $\|p^\star(s,a)\|_0 \ll S$ for all (s,a). We do not make such assumptions on the sparsity of the MDP and we note that the regret bound of Cheung (2019) scales linearly in S when $\|p^\star(s,a)\|_0 = \Theta(S)$. Also, the single-episode setting requires a strong reachability assumption, not present in the episodic setting.

Remark 3.3.5. The aforementioned regret bound can be turned into a PAC bound of $\tilde{\mathcal{O}}\big(\frac{S^2AH^5}{\epsilon^2}\big)$ by taking the uniform mixture of policies $\pi_1, \pi_2, \ldots, \pi_k$.

3.4 Concave-convex setting

We now extend the algorithm and guarantees derived for the basic setting to when the objective is concave function of the accumulated reward and the constraints are expressed as a convex function of the cumulative consumptions. Our approach is modular, seamlessly building on the basic setting.

Setting and objective. Formally, there is a concave reward-objective function $f : \mathbb{R} \to \mathbb{R}$ and a convex consumption-objective function $g : \mathbb{R}^d \to \mathbb{R}$; the only assumption is that these functions are L-Lipschitz for some constant L, i.e., $|f(x) - f(y)| \le L|x-y|$ for any $x, y \in \mathbb{R}$, and $|g(x) - g(y)| \le L\|x-y\|_1$ for any $x, y \in \mathbb{R}^d$. Analogous to Eq. (3.1), the learner wishes to compete against the following benchmark which can be viewed as a reinforcement learning variant of the benchmark used by Agrawal and Devanur (2014) in multi-armed bandits:

$$\max_{\pi} f\Big(\mathbb{E}^{\pi,p^\star}\Big[\sum_{h=1}^{H} r^\star(s_h,a_h)\Big]\Big) \quad \text{s.t.} \quad g\Big(\mathbb{E}^{\pi,p^\star}\Big[\sum_{h=1}^{H} \mathbf{c}^\star(s_h,a_h)\Big]\Big) \le 0. \tag{3.7}$$

The reward and consumption regrets are therefore adapted to:

$$\textsc{ConvexRewReg}(k) \coloneqq f\Big(\mathbb{E}^{\pi^\star,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h,a_h\big)\Big]\Big) - f\Big(\frac{1}{k}\sum_{t=1}^{k}\mathbb{E}^{\pi_t,p^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h,a_h\big)\Big]\Big),$$

$$\textsc{ConvexConsReg}(k) \coloneqq g\Big(\frac{1}{k}\sum_{t=1}^{k}\mathbb{E}^{\pi_t,p^\star}\Big[\sum_{h=1}^{H} \mathbf{c}^\star\big(s_h,a_h\big)\Big]\Big).$$

Our algorithm. As in the basic setting, we wish to create a bonus-enhanced model and optimize over it. To model the transition probabilites, we use empirical estimates $p^{(k)} = \widehat{p}_k$ of Eq. Eq. (3.3) as before. However, since reward and consumption objectives are no longer monotone in the accumulated rewards and consumption respectively, it does not make sense to simply add or subtract $\widehat{b}_k$ (defined in Eq. 3.5) as we did before. Instead we compute the policy π_k of episode k together with the model by solving the following optimization problem which we call ConvexConPlanner:

$$\max_{\pi}\max_{r^{(k)}\in\left[\widehat{r}_k\pm\widehat{b}_k\right]} f\Big(\mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} r^{(k)}\big(s_h,a_h\big)\Big]\Big) \text{ s.t. } \min_{\mathbf{c}^{(k)}\in\left[\widehat{\mathbf{c}}_k\pm\widehat{b}_k\cdot\mathbf{1}\right]} g\Big(\mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} \mathbf{c}^{(k)}\big(s_h,a_h\big)\Big]\Big) \le 0.$$

The above problem is convex in the occupation measures,[4] i.e., the probability $\rho(s,a,h)$ that the learner is at state-action-step (s,a,h) — c.f. Appendix B.1.2 for further discussion.

$$\begin{aligned}\max_{\rho}\max_{r\in\left[\widehat{r}_k\pm\widehat{b}_k\right]} f\Big(\sum_{s,a,h}\rho(s,a,h)r(s,a)\Big) \quad &\text{s.t.} \min_{\mathbf{c}\in\left[\widehat{\mathbf{c}}_k\pm\widehat{b}_k\cdot\mathbf{1}\right]} g\Big(\sum_{s,a,h}\rho(s,a,h)\mathbf{c}(s,a)\Big) \le 0\\ \forall s', h: \quad &\sum_{a}\rho(s',a,h+1) = \sum_{s,a}\rho(s,a,h)\widehat{p}_k(s'|s,a)\\ \forall s,a,h: \quad &0\le\rho(s,a,h)\le 1 \quad \text{and} \quad \sum_{s,a}\rho(s,a,h) = 1.\end{aligned}$$

Guarantee for concave-convex setting. To extend the guarantee of the basic setting to the concave-convex setting, we face an additional challenge: it is not immediately clear that the optimal policy $\pi^\star$ is feasible for the ConvexConPlanner program because ConvexConPlanner is

[4] Under mild assumptions, this program can be solved in polynomial time similar to its bandit analogue of Lemma 4.3 in (Agrawal and Devanur, 2014). We note that in the basic setting, it reduces to just a linear program.

defined with respect to the empirical transition probabilities $p^{(k)}$.[5] Moreover, when $H > 1$, it is not straightforward to show that objective in the used model is always greater than the one in the true model as the used model transitions $p^{(k)}(s, a)$ can lead to different states than the ones encountered in the true model.[6] We deal with both of these issues by introducing a novel application of the mean-value theorem to show that $\pi^\star$ is indeed a feasible solution of that program and create a similar regret decomposition to Proposition 3.3.3 (see Proposition B.3.1 and more discussion in Appendix B.3.1); this allows us to plug in the results developed for the basic setting. The full proof is provided in Appendix B.3.

Theorem 3.4.1. *Let L be the Lipschitz constant for f and g and let* RewReg *and* ConsReg *be the reward and consumption regrets for the basic setting (Theorem 3.3.4) with the failure probability δ. With probability $1 - \delta$, our algorithm in the concave-convex setting has reward and consumption regret upper bounded by $L \cdot$ RewReg and $Ld \cdot$ ConsReg respectively.*

The linear dependence on d in the consumption regret above comes from the fact that we assume g is Lipschitz under ℓ_1 norm.

3.5 Knapsack setting

Our last technical section extends the algorithm and guarantee of the basic setting to scenarios where the constraints are hard which is in accordance with most of the literature on *bandits with knapsacks*. The goal here is to achieve aggregate reward regret that is sublinear in the time horizon (in our case, the number of episodes K), while also respecting budget constraints for as small budgets as possible. We derive guarantees in terms of *reward regret*, as defined previously, and then argue that our guarantee extends to the seemingly stronger benchmark of the best dynamic policy.

Setting and objective. Each resource $i \in \mathcal{D}$ has an aggregate budget B_i that the learner should not exceed over K episodes. Unlike the basic setting, where we track the consumption regret, here we

[5] Note that in multi-armed bandit concave-convex setting (Agrawal and Devanur, 2014), proving feasibility of the best arm is straightforward as there are no transitions.

[6] Again, this is not an issue in multi-armed bandits.

view this as a hard constraint. As in most works on bandits with knapsacks, the algorithm is allowed to use a "null action" for an episode, i.e., an action that yields a zero reward and consumption when selected at the beginning of an episode. The learner wishes to maximize her aggregate reward while respecting these hard constraints. We reduce this problem to a specific variant of the basic problem Eq. (3.1) with $\xi(i) = \frac{B_i}{K}$. We modify the solution to Eq. (3.1) to take the null action if any constraint is violated and call the resulting benchmark $\pi^\star$. Note that $\pi^\star$ satisfies constraints in expectation. At the end of this section, we explain how our algorithm also competes against a benchmark that is required to respect constraints deterministically (i.e., with probability one across all episodes).

Our algorithm. In the basic setting of Section 3.3, we showed a reward regret guarantee and a consumption regret guarantee, proving that the average constraint violation is $\mathcal{O}(1/\sqrt{K})$. Now we seek a stronger guarantee: the learned policy needs to satisfy budget constraints with high probability. Our algorithm optimizes a mathematical program KnapsackConPlanner Eq. (3.8) that strengthens the consumption constraints:

$$\max_{\pi} \mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} r^{(k)}\big(s_h,a_h\big)\Big] \quad \text{s.t.} \quad \forall i \in \mathcal{D} : \mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} c^{(k)}\big(s_h,a_h,i\big)\Big] \leq \frac{(1-\epsilon)B_i}{K}. \tag{3.8}$$

In the above, $p^{(k)}$, $r^{(k)}$, $\mathbf{c}^{(k)}$ are exactly as in the basic setting and $\epsilon > 0$ is instantiated in the theorem below. Note that the program Eq. (3.8) is feasible thanks to the existence of the null action. The following mixture policy induces a feasible solution: with probability $1-\epsilon$, we play the optimal policy $\pi^\star$ for the entire episode; with probability ϵ, we play the null action for the entire episode. Note that the above program can again be cast as a linear program in the occupancy measure space — c.f. Appendix B.1.3 for further discussion.

Guarantee for knapsack setting. The guarantee of the basic setting on this tighter mathematical program seamlessly transfers to a reward guarantee that does not violate the hard constraints.

Theorem 3.5.1. *Assume that* $\min_i B_i \leq KH$*, i.e., constraints are non-vacuous. Let* AggReg(δ) *be a bound on the aggregate (across episodes) reward or consumption regret for the soft-constraint*

setting (Theorem 3.3.4) with the failure probability δ. Let $\epsilon = \frac{\text{AggReg}(\delta)}{\min_i B_i}$. If $\min_i B_i > \text{AggReg}(\delta)$ then, with probability $1-\delta$, the reward regret in the hard-constraint setting is at most $\frac{2H\text{AggReg}(\delta)}{\min_i B_i}$ and constraints are not violated.

The above theorem implies that the aggregate reward regret is sublinear in K as long as $\min_i B_i \gg H\text{AggReg}(\delta)$. The analysis in the above main theorem (provided in Appendix B.4) is *modular* in the sense that it leverages the ConRL's performance to solve Eq. (3.8) in a black-box manner. Smaller $\text{AggReg}(\delta)$ from the basic soft-constraint setting immediately translates to smaller reward regret and smaller budget regime (i.e., $\min_i B_i$ can be smaller). In particular, using the $\text{AggReg}(\delta)$ bound of Theorem 3.3.4, the reward regret is sublinear as long as $\min_i B_i = \Omega(\sqrt{K})$.

In contrast, previous work of Cheung (2019) can only deal with larger budget regime, i.e., $\min_i B_i = \Omega(K^{2/3})$. Although the guarantees are not directly comparable as the latter is for the single-episode setting, which requires further reachability assumptions, the budget we can handle is significantly smaller and in the next section we show that our algorithm has superior empirical performance in episodic settings even when such assumptions are granted.

Dynamic policy benchmark. The common benchmark used in bandits with knapsacks is not the best stationary policy $\pi^\star$ that respects constraints in expectation but rather the best *dynamic* policy (i.e., a policy that makes decisions based on the history) that never violates hard constraints *deterministically*. In Appendix B.4, we show that the optimal dynamic policy (formally defined there) has reward less than policy $\pi^\star$ (informally, this is because $\pi^\star$ respects constraints in expectation while the dynamic policy has to satisfy constraints deterministically) and therefore the guarantee of Theorem 3.5.1 also applies against the optimal dynamic policy.

3.6 Empirical comparison to other concave-convex approaches

In this section, we evaluate the performance of ConRL against previous approaches.[7] Although our ConPlanner (see Appendix B.1) can be solved exactly using linear programming (**?**), in our

[7] Code is available at `https://github.com/miryoosefi/ConRL`

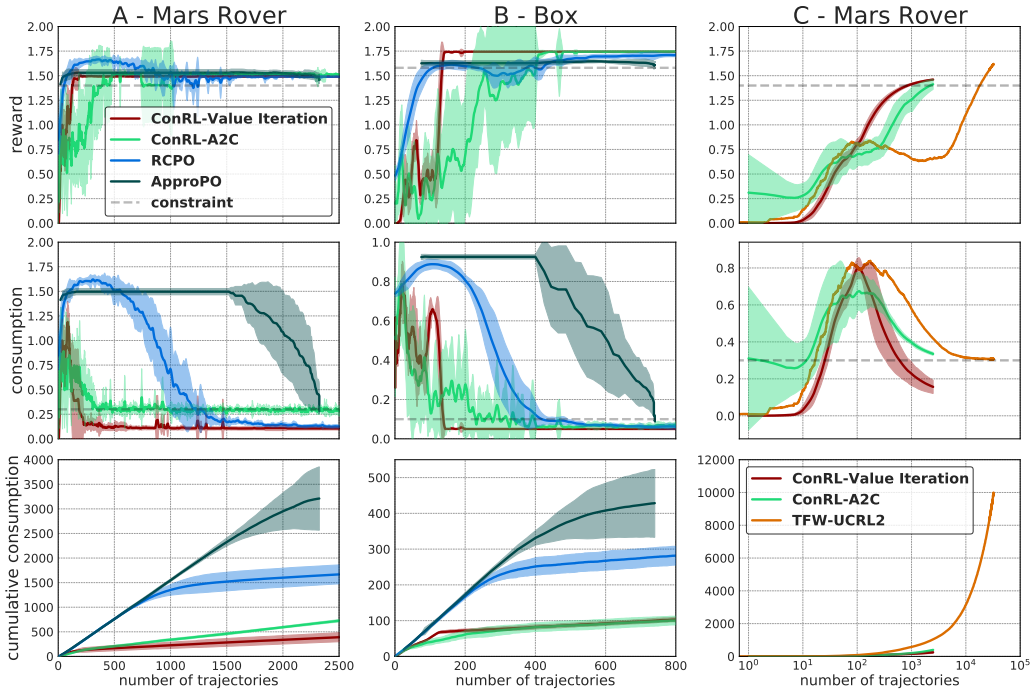


Figure 3.1: The performance of the algorithms as a function of the number of sample trajectories (trajectory = 30 samples); showing average and standard deviation over 10 runs. Dashed line in the second row is the upper bound on the consumption (for all algorithms), the dashed line in the first row is a lower bound on the reward (only required by ApproPO).

experiments, it suffices to use Lagrangian heuristic, denoted as LagrConPlanner (see Appendix B.5.1). This Lagrangian heuristic only needs a planner for the *unconstrained* RL task. We consider two unconstrained RL algorithms as planners: value iteration and a model-based Advantage Actor-Critic (A2C) (Mnih et al., 2016) (based on fictitious samples drawn from the model provided as an input). The resulting variants of LagrConPlanner are denoted ConRL-Value Iteration and ConRL-A2C. We run our experiments on two grid-world environments *Mars rover* (Tessler et al., 2019) and *Box* (Leike et al., 2017).[8]

Mars rover. The agent must move from the initial position to the goal without crashing into rocks. If the agent reaches the goal or crashes into a rock it will stay in that cell for the remainder of the episode. Reward is 1 when the agent reaches the goal and $1/H$ afterwards. Consumption is 1 when the agent crashes into a rock and $1/H$ afterwards. The episode horizon H is 30 and the agent's action is perturbed with probability 0.1 to a random action.

Box. The agent must move a box from the initial position to the goal while avoiding corners (cells adjacent to at least two walls). If the agent reaches the goal it stays in that cell for the remainder of the episode. Reward is 1 when agent reaches the goal for the first time and $1/H$ afterwards; consumption is $1/H$ whenever the box is in a corner. Horizon H is 30 and the agent's action is perturbed with probability 0.1 to a random action.

We compare ConRL to previous constrained approaches (derived for either episodic or single-episode settings) in Figure 3.1. We keep track of three metrics: episode-level reward and consumption (the first two rows) and cumulative consumption (the third row). Episode-level metrics are based on the most recent episode in the first two columns, i.e., we plot $\mathbb{E}^{\pi_k}[\sum_{h=1}^{H} r_h^\star]$ and $\mathbb{E}^{\pi_k}[\sum_{h=1}^{H} c_h^\star]$. In the third column, we plot the average across episodes so far, i.e., $\frac{1}{k}\sum_{t=1}^{k} \mathbb{E}^{\pi_t}[\sum_{h=1}^{H} r_h^\star]$ and $\frac{1}{k}\sum_{t=1}^{k} \mathbb{E}^{\pi_t}[\sum_{h=1}^{H} c_h^\star]$, and we use the log scale for the x-axis. The cumulative consumption is $\sum_{t=1}^{k}\sum_{h=1}^{H} c_{t,h}$ in all columns. See Appendix B.5 for further details about experiments.

Episodic setting. We first compare our algorithms to two episodic RL approaches: ApproPO

[8] We are not aware of any benchmarks for convex/knapsack constraints. For transparency, we compare against prior works handling concave-convex or knapsack settings on established benchmarks for the linear case.

(Miryoosefi et al., 2019) and RCPO (Tessler et al., 2019). We note that none of the previous approaches in this setting address sample-efficient exploration. In addition, most of them are limited to linear constraints, with the exception of ApproPO (Miryoosefi et al., 2019), which can handle general convex constraints.[9] Both ApproPO and RCPO (used as a baseline by Miryoosefi et al., 2019) maintain and update a weight vector λ, used to derive reward for an unconstrained RL algorithm, which we instantiate as A2C. ApproPO focuses on the feasibility problem, so it requires to specify a lower bound on the reward, which we set to 0.3 for Mars rover and 0.1 for Box. In the first two columns of Figure 3.1 we see that both versions of ConRL are able to solve the constrained RL task with a much smaller number of trajectories (see top two rows), and their overall consumption levels are substantially lower (the final row) than those of the previous approaches.

Single-episode setting. Closest to our work is TFW-UCRL2 (Cheung, 2019), which is based on UCRL (Jaksch et al., 2010). However, that approach focuses on the single-episode setting and requires a strong reachability assumption. By connecting terminal states of our MDP to the intial state, we reduce our episodic setting to single-episode setting in which we can compare ConRL against TFW-UCRL2. Results for Mars rover are depicted in last column of Figure 3.1.[10] Again, both versions of ConRL find the solution with a much smaller number of trajectories (note the log scale on the x-axis) and their overall consumption levels are much lower than those of TFW-UCRL2. This suggests that TFW-UCRL2 might be impractical in (at least some) episodic settings.

3.7 Conclusions

In this chapter we study two types of constraints in the framework of constrained tabular episodic reinforcement learning: concave rewards and convex constraints, and knapsacks constraints. Our algorithms achieve near-optimal regret in both settings, and experimentally we show that our approach outperforms prior works on constrained reinforcement learning.

Regarding future work, it would be interesting to extend our framework to continuous state and

[9] In addition to that, trust region methods like CPO (Achiam et al., 2017) address a more restrictive setting and require constraint satisfaction at each iteration; for this reason, they are not included in the experiments.

[10] Due to a larger state space, it was computationally infeasible to run TFW-UCRL2 in the Box environment.

action spaces. Potential directions include extensions to Lipschitz MDPs (Song and Sun, 2019) and MDPs with linear parameterization (Jin et al., 2020c) where optimism-based exploration algorithms exist under the classic reinforcement learning setting without constraints.

Chapter 4

A Simple Reward-free Approach to Constrained Reinforcement Learning

4.1 Introduction

In a wide range of modern reinforcement learning (RL) applications, it is not sufficient for the learning agents to only maximize a scalar reward. More importantly, they must satisfy various *constraints*. For instance, such constraints can be the physical limit of power consumption or torque in motors for robotics tasks (Tessler et al., 2019); the budget for computation and the frequency of actions for real-time strategy games (Vinyals et al., 2019); and the requirement for safety, fuel efficiency and human comfort for autonomous drive (Le et al., 2019). In addition, constraints are also crucial in tasks such as dynamic pricing with limited supply (Besbes and Zeevi, 2009; Babaioff et al., 2015), scheduling of resources on a computer cluster (Mao et al., 2016), imitation learning (Syed and Schapire, 2007; Ziebart et al., 2008; Sun et al., 2019b), as well as RL with fairness (Jabbari

et al., 2017).

These huge demand in practice gives rise to a subfield—constrained RL, which focuses on designing efficient algorithms to find near-optimal policies for RL problems under linear or general convex constraints. Most constrained RL works directly combine the existing techniques such as value iteration and optimism from unconstrained literature, with new techniques specifically designed to deal with linear constraints (Efroni et al., 2020; Ding et al., 2021; Qiu et al., 2020) or general convex constraints (Brantley et al., 2020; Yu et al., 2021). The end product is a single new complex algorithm which is tasked to solve all the challenges of learning dynamics, exploration, planning as well as constraints satisfaction simultaneously. Thus, these algorithms need to be re-analyzed from scratch, and it is highly nontrivial to translate the progress in the unconstrained RL to the constrained setting.

On the other hand, reward-free RL—proposed in Jin et al. (2020b)—is a framework for the unconstrained setting, which learns the transition dynamics without using the reward. The framework has two phases: in the exploration phase, the agent first collects trajectories from a Markov decision process (MDP) and learns the dynamics without a pre-specified reward function. After exploration, the agent is tasked with computing near-optimal policies under the MDP for a collection of given reward functions. This framework is particularly suitable when there are multiple reward functions of interest, and has been developed recently to attack various settings including tabular MDPs (Jin et al., 2020b; Zhang et al., 2020a), linear MDPs (Wang et al., 2020a; Zanette et al., 2020b), and tabular Markov games (Liu et al., 2020).

Contribution. In this chapter, we propose a simple approach to solve constrained RL problems by bridging the reward-free RL literature and constrained RL literature. Our approach isolates the challenges of constraint satisfaction, and leaves the remaining RL challenges such as learning dynamics and exploration to reward-free RL. This allows us to design a new algorithm which purely focuses on addressing the constraints. Formally, we design a meta-algorithm for RL problems with general convex constraints. Our meta-algorithm takes a reward-free RL solver, and can be used to directly solve the approachability problem, as well as the constrained MDP problems using very

Table 4.1: Sample complexity for algorithms to solve reward-free RL for VMDP (Definition 4.2.1), approachability (Definition 4.2.3) and CMDP with general convex constraints (Definition 4.2.4).[1]

	Algorithm	**Reward-free**	**Approachability**	**CMDP**
Tabular	Wu et al. (2020)	$\tilde{\mathcal{O}}(\min\{d,S\}H^4SA/\epsilon^2)$	-	-
	Brantley et al. (2020)	-	-	$\tilde{\mathcal{O}}(d^2H^3S^2A/\epsilon^2)$
	Yu et al. (2021)	-	$\tilde{\mathcal{O}}(\min\{d,S\}H^3SA/\epsilon^2)$	$\tilde{\mathcal{O}}(\min\{d,S\}H^3SA/\epsilon^2)$
	This work	$\tilde{\mathcal{O}}(\min\{d,S\}H^4SA/\epsilon^2)$	$\tilde{\mathcal{O}}(\min\{d,S\}H^4SA/\epsilon^2)$	$\tilde{\mathcal{O}}(\min\{d,S\}H^4SA/\epsilon^2)$
Linear	**This work**	$\tilde{\mathcal{O}}(d_{\text{lin}}^3H^6/\epsilon^2)$	$\tilde{\mathcal{O}}(d_{\text{lin}}^3H^6/\epsilon^2)$	$\tilde{\mathcal{O}}(d_{\text{lin}}^3H^6/\epsilon^2)$

small amount of samples in addition to what is required for reward-free RL.

Our framework enables direct translation of any progress in reward-free RL to constrained RL. Leveraging recent advances in reward-free RL, our meta-algorithm directly implies sample-efficient guarantees of constrained RL in the settings of tabular MDP, linear MDP, as well as tabular two-player Markov games. In particular,

- *Tabular setting*: Our work achieves sample complexity of $\tilde{\mathcal{O}}(\min\{d,S\}H^4SA/\epsilon^2)$ for all three tasks of reward-free RL for Vector-valued MDPs (VMDP), approachability, and RL with general convex constraints. Here d is the dimension of VMDP or the number of constraints, S, A are the number of states and actions, H is the horizon, and ϵ is the error tolerance. It matches the best existing results up to a factor of H.

- *Linear setting*: Our work provides new sample complexity of $\tilde{\mathcal{O}}(d_{\text{lin}}^3H^6/\epsilon^2)$ for all three tasks above for linear MDPs. To our best knowledge, this result is the first sample-efficient result for approachability and also constrained RL with general convex constraints in the linear function approximation setting.

- *Two-player setting*: Our work extends to the setting of tabular two-player vector-valued Markov games and achieves low regret of $\alpha(T) = \mathcal{O}(\epsilon/2 + \sqrt{H^2\iota/T})$ at the cost of this $\mathcal{O}(\epsilon)$ bias in regret as well as additional samples for preprocessing.

4.1.1 Related work

In this section, we review the related works on three tasks studied in this chapter—reward-free RL, approachability, and constrained RL.

Reward-free RL. Reward-free exploration has been formalized by Jin et al. (2020b) for the tabular setting. Furthermore, Jin et al. (2020b) proposed an algorithm which has sample complexity $\tilde{\mathcal{O}}(\text{poly}(H)S^2A/\epsilon^2)$ outputting ϵ-optimal policy for arbitrary number of reward functions. More recently, Zhang et al. (2020a); Liu et al. (2020) propose algorithm VI-Zero with sharp sample complexity of $\tilde{\mathcal{O}}(\text{poly}(H)\log(N)SA/\epsilon^2)$ capable of handling N fixed reward functions. Wang et al. (2020a); Zanette et al. (2020b) further provide reward-free learning results in the setting of linear function approximation, in particular, Wang et al. (2020a) guarantees to find the near-optimal policies for an arbitrary number of (linear) reward functions within a sample complexity of $\tilde{\mathcal{O}}(\text{poly}(H)d_{\text{lin}}^3/\epsilon^2)$. All results mentioned above are for scalar-valued MDPs. For the vector-valued MDPs (VMDPs), very recent work of Wu et al. (2020) designs a reward-free algorithm with sample complexity guarantee $\tilde{\mathcal{O}}(\text{poly}(H)\min\{d,S\}SA/\epsilon^2)$ in the tabular setting. Compared to Wu et al. (2020), our reward-free algorithms for VMDP is adapted from the VI-Zero algorithm presented in Liu et al. (2020); While achieving the same sample complexity, it allows arbitrary planning algorithms in the planning phase.

Approachability and Constrained RL Approachability and Constrained RL are two related tasks involving constraints. Inspired by Blackwell approachability (Blackwell, 1956), recent work of Miryoosefi et al. (2019) introduces approachability task for VMDPs. However, the proposed algorithm does not have polynomial sample complexity guarantees. More recently, Yu et al. (2021) gave a new algorithm for approachability for both VMDPs and vector-valued Markov games (VMGs). Yu et al. (2021) provides regret bounds for the proposed algorithm resulting in sample complexity guarantees of $\tilde{\mathcal{O}}(\text{poly}(H)\min\{d,S\}SA/\epsilon^2)$ for approachability in VMDPs and $\tilde{\mathcal{O}}(\text{poly}(H)\min\{d,S\}SAB/\epsilon^2)$

[1]The presented sample complexities are all under the L_2 normalization conditions as studied in this work. We comment that the results of (Wu et al., 2020; Brantley et al., 2020; Yu et al., 2021) are originally presented under L_1/L_∞ normalization conditions. While the results in Wu et al. (2020) can be directly adapted to our setting as stated in the table, the other two results Brantley et al. (2020); Yu et al. (2021) will be no better than the displayed results after adaptation.

for approachability in VMGs.

Sample-efficient exploration in constrained reinforcement learning has been recently studied in a recent line of work by Brantley et al. (2020); Qiu et al. (2020); Efroni et al. (2020); Ding et al. (2021); Singh et al. (2020). All these works are also limited to linear constraints except Brantley et al. (2020) which extends their approach to general convex constraints achieving sample complexity of $\tilde{\mathcal{O}}(\text{poly}(H)d^2S^2A/\epsilon^2)$. However, Brantley et al. (2020) requires solving a large-scale convex optimization sub-problem. The best result for constrained RL with general convex constraints can be achieved by the approachability-based algorithm in Yu et al. (2021) obtaining sample complexity of $\tilde{\mathcal{O}}(\text{poly}(H)\min\{d, S\}SA/\epsilon^2)$. Technically, our meta-algorithm is based on the Fenchel's duality, which is similar to Yu et al. (2021). In contrast, Yu et al. (2021) does not use reward-free RL, and is thus different from our results in terms of algorithmic approaches. Consequently, Yu et al. (2021) does not reveal the deep connections between reward-free RL and constrained RL, which is one of the main contribution of this work. In addition, Yu et al. (2021) does not address the function approximation setting.

Finally, we note that among all results mentioned above, only Ding et al. (2021) has considered models beyond tabular setting in the context of constrained RL. The model studied in Ding et al. (2021) is known as linear mixture MDPs which is different and incomparable to the linear MDP models considered in this work. We further comment that Ding et al. (2021) can only handle linear constraints for CMDP, while our results is capable of solving CMDPs with general convex constraints.

4.2 Preliminaries and problem setup

We consider an episodic *vector-valued Markov decision process* (VMDP) specified by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, \mathbf{r})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, H is the length of each episode, $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ is the collection of *unknown* transition probabilities with $\mathbb{P}_h(s' \mid s, a)$ equal to the probability of transiting to s' after taking action a in state s at the h^{th} step, and $\mathbf{r} = \{\mathbf{r}_h : \mathcal{S} \times \mathcal{A} \to \mathcal{B}(1)\}_{h=1}^H$ is a collection of *unknown* d-dimensional return functions, where $\mathcal{B}(r)$

is the d-dimensional Euclidean ball of radius r centered at the origin.

Interaction protocol. In each episode, agent starts at a *fixed* initial state s_1. Then, at each step $h \in [H]$, the agent observes the current state s_h, takes action a_h, receives stochastic sample of the return vector $\mathbf{r}_h(s_h, a_h)$, and it causes the environment to transit to $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, a_h)$. We assume that stochastic samples of the return function are also in $\mathcal{B}(1)$, almost surely.

Policy and value function. A policy π of an agent is a collection of H functions $\{\pi_h : \mathcal{S} \to \Delta(\mathcal{A})\}_{h=1}^{H}$ that map states to distribution over actions. The agent following policy π, picks action $a_h \sim \pi_h(s_h)$ at the h^{th} step. We denote $\mathbf{V}_h^\pi : \mathcal{S} \to \mathcal{B}(H)$ as the value function at step h for policy π, defined as

$$\mathbf{V}_h^\pi(s) := \mathbb{E}_\pi \left[\textstyle\sum_{h'=h}^{H} \mathbf{r}_{h'}(s_{h'}, a_{h'}) \mid s_h = s \right].$$

Similarly, we denote $\mathbf{Q}_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathcal{B}(H)$ as the Q-value function at step h for policy π, where

$$\mathbf{Q}_h^\pi(s, a) := \mathbb{E}_\pi \left[\textstyle\sum_{h'=h}^{H} \mathbf{r}_{h'}(s_{h'}, a_{h'}) \mid s_h = s, a_h = a \right].$$

Scalarized MDP. For a VMDP $\mathcal{M}$ and $\theta \in \mathcal{B}(1)$, we define scalar-valued MDP $\mathcal{M}_\theta = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r_\theta)$, where $r_\theta = \{\langle \theta, \mathbf{r}_h \rangle : \mathcal{S} \times \mathcal{A} \to [-1, 1]\}_{h=1}^{H}$. We denote $V_h^\pi(\cdot; \theta) : \mathcal{S} \to [-H, H]$ as the scalarized value function at step h for policy π, defined as

$$V_h^\pi(s; \theta) := \mathbb{E}_\pi \left[\sum_{h'=h}^{H} \langle \theta, \mathbf{r}_{h'}(s_{h'}, a_{h'}) \rangle \mid s_h = s \right] = \langle \theta, \mathbf{V}_h^\pi(s) \rangle.$$

Similarly, we denote $Q_h^\pi(\cdot; \theta) : \mathcal{S} \times \mathcal{A} \to [-H, H]$ as the scalarized Q-value function at step h for policy π, where

$$Q_h^\pi(s, a; \theta) := \mathbb{E}_\pi \left[\textstyle\sum_{h'=h}^{H} \langle \theta, \mathbf{r}_{h'}(s_{h'}, a_{h'}) \rangle \mid s_h, a_h = s, a \right] = \langle \theta, \mathbf{Q}_h^\pi(s, a) \rangle.$$

For a fixed $\theta \in \mathbb{R}^d$, there exists an optimal policy $\pi_\theta^\star$, maximizing value for all states (Puterman, 2014); i.e., $V_h^{\pi_\theta^\star}(s; \theta) = \sup_\pi V_h^\pi(s; \theta)$ for all $s \in \mathcal{S}$ and $h \in [H]$. We abbreviate $V^{\pi_\theta^\star}(\cdot; \theta)$ and $Q^{\pi_\theta^\star}(\cdot; \theta)$

as $V^\star(\cdot;\theta)$ and $Q^\star(\cdot;\theta)$ respectively.

4.2.1 Reward-free exploration (RFE) for VMDPs

The task of *reward-free exploration* (formalized by Jin et al. (2020b) for tabular MDPs) considers the scenario in which the agents interacts with the environment without guidance of reward information. Later, the reward information is revealed and the agents is required to compute the near-optimal policy. In this section, we describe its counterpart for VMDPs [1]. Formally, it consists of two phases:

Exploration phase. In the exploration phase, agent explores the unknown environment without observing any information regarding the return function. Namely, at each episode the agent executes policies to collect samples. The policies can depend on dynamic observations $\{s_h^k, a_h^k\}_{(k,h)\in[K]\times[H]}$ in the past episodes, but not the return vectors.

Planning phase. In the planning phase, the agent no longer interacts with the environment; however, stochastic samples of the d-dimensional return function for the collected episodes is revealed to the agent, i.e. $\{\mathbf{r}_h^k\}_{(k,h)\in[K]\times[H]}$. Based on the episodes collected during the exploration phase, the agent outputs the near-optimal policies of $\mathcal{M}_\theta$ given an arbitrary number of vectors $\theta \in \mathcal{B}(1)$.

Definition 4.2.1 (Reward-free algorithm for VMDPs)**.** For any $\epsilon, \delta > 0$, after collecting $m_{\mathrm{RFE}}(\epsilon, \delta)$ episodes during the exploration phase, with probability at least $1-\delta$, the algorithm satisfies

$$\forall \theta \in \mathcal{B}(1): \quad V_1^\star(s_1;\theta) - V_1^{\pi_\theta}(s_1;\theta) \le \epsilon, \tag{4.1}$$

where π_θ is the output of the planning phase for vector θ as input. The function m_{RFE} determines the *sample complexity* of the RFE algorithm.

Remark 4.2.2. Standard reward-free setup concerns MDPs with scalar reward, and requires the algorithm to find the near-optimal policies for N different prespecified reward functions in the planning phase, where the sample complexity typically scales with $\log N$. This type of results can

[1] RFE for VMDPs is also called preference-free exploration problem in Wu et al. (2020)

be adapted into a guarantee in the form of Eq. (4.1) for VMDP by ϵ-covering of θ over $\mathcal{B}(1)$ and a modified concentration arguments (see the proofs of Theorem 4.4.1 and Theorem 4.6.4 for more details).

4.2.2 Approachability for VMDPs

In this section we provide the description for the *approachability* task for VMPDs introduced by Miryoosefi et al. (2019). Given a vector-valued Markov decision process and a convex target set $\mathcal{C}$, the goal is to learn a policy whose expected cumulative return vector lies in the target set (akin to Blackwell approachability in single-turn games, Blackwell 1956). We consider the agnostic version of this task which is more general since it doesn't need to assume that such policy exists; instead, the agent learns to minimize the Euclidean distance between expected return of the learned policy and the target set.

Definition 4.2.3 (Approachability algorithm for VMDPs)**.** For any $\epsilon, \delta > 0$, after collecting $m_{\text{APP}}(\epsilon, \delta)$ episodes, with probability at least $1 - \delta$, the algorithm satisfies

$$\text{dist}(\mathbf{V}_1^{\pi^{\text{out}}}(s_1), \mathcal{C}) \leq \min_{\pi} \text{dist}(\mathbf{V}_1^{\pi}(s_1), \mathcal{C}) + \epsilon, \tag{4.2}$$

where π^{out} is the output of the algorithm and $\text{dist}(\mathbf{x}, \mathcal{C})$ is the Euclidean distance between point $\mathbf{x}$ and set $\mathcal{C}$. The function m_{APP} determines the *sample complexity* of the algorithm.

4.2.3 Constrained MDP (CMDP) with general convex constraints

In this section we describe *constrained Markov decision processes* (CMDPs) introduced by Altman (1999). The goal of this setting is to minimize cost while satisfying some linear constraints over consumption of d resources (resources are akin to $\mathbf{r}$ in our case). Although, the original definition only allows for linear constraints, we consider the more general case of arbitrary convex constraints. More formally, consider a VMDP $\mathcal{M}$, a cost function $c = \{c_h : \mathcal{S} \times \mathcal{A} \to [-1, 1]\}_{h=1}^{H}$, and a convex

constraint set $\mathcal{C}$. The agent goal is to compete against the following benchmark:

$$\min_{\pi} C_1^{\pi}(s_1) \quad \text{s.t.} \quad \mathbf{V}_1^{\pi}(s_1) \in \mathcal{C},$$

where $C_h^{\pi} = \mathbb{E}_{\pi}\left[\sum_{h'=h}^{H} c_{h'}(s_{h'}, a_{h'}) \mid s_h = s\right]$.

Definition 4.2.4 (Algorithm for CMDP)**.** For any $\epsilon, \delta > 0$, after collecting $m_{\mathrm{CMDP}}(\epsilon, \delta)$ episodes, with probability at least $1 - \delta$, the algorithm satisfies

$$\begin{cases} C_1^{\pi^{\mathrm{out}}}(s_1) - \min_{\pi: \mathbf{V}_1^{\pi}(s_1) \in \mathcal{C}} C_1^{\pi}(s_1) \leq \epsilon \\ \mathrm{dist}\big(\mathbf{V}_1^{\pi^{\mathrm{out}}}(s_1), \mathcal{C}\big) \leq \epsilon, \end{cases} \tag{4.3}$$

where π^{out} is the output of the algorithm. The function m_{CMDP} determines the *sample complexity* of the algorithm.

As also mentioned in the prior works (Miryoosefi et al., 2019; Yu et al., 2021), we formally show in the following theorem that approachability task (Definition 4.2.3) can be considered more general compared to CMDP (Definition 4.2.4); Namely, given any algorithm for the former we can obtain an algorithm for the latter by incurring only extra logarithmic factor and a negligible overhead. The idea is to incorporate cost into the constraint set $\mathcal{C}$ and perform an (approximate) binary search over the minimum attainable cost. The reduction and the proof can be found in Appendix C.1.

Theorem 4.2.5. *Given any approachability algorithm (Definition 4.2.3) with sample complexity m_{APP}, we can design an algorithm for CMDP (Definition 4.2.4) with sample complexity m_{CMDP}, satisfying*

$$m_{\mathrm{CMDP}}(\epsilon, \delta) \leq \tilde{\mathcal{O}}\Big(m_{\mathrm{APP}}\left(\tfrac{\epsilon}{6}, \tfrac{\epsilon\delta}{12H}\right) + \tfrac{H^2 \log[dH/\epsilon\delta]}{\epsilon^2}\Big).$$

4.3 Meta-algorithm for VMDPs

In this section, equipped with preliminaries discussed in Section 4.2, we are ready to introduce our main algorithmic framework for VMDPs bridging reward-free RL and approachability.

Before introducing the algorithm, we explain the intuition behind it. By Fenchel's duality (similar to Yu et al. 2021), one can show that

$$\begin{aligned}&\min_{\pi} \operatorname{dist}(\mathbf{V}_1^{\pi}(s_1, \mathcal{C}))\\ &= \min_{\pi} \max_{\theta \in \mathcal{B}(1)} \Big[\langle \theta, \mathbf{V}_1^{\pi}(s_1, \mathcal{C})\rangle - \max_{\mathbf{x}' \in \mathcal{C}} \langle \theta, \mathbf{x}'\rangle\Big].\end{aligned}$$

It satisfies the minimax conditions since it's concave in θ and convex in π (by allowing mixture policies); therefore, minimax theorem Neumann (1928) implies that we can equivalently solve

$$\max_{\theta \in \mathcal{B}(1)} \min_{\pi} \Big[\langle \theta, \mathbf{V}_1^{\pi}(s_1, \mathcal{C})\rangle - \max_{\mathbf{x}' \in \mathcal{C}} \langle \theta, \mathbf{x}'\rangle\Big].$$

This max-min form allows us to use general technique of Freund and Schapire (1999) for solving a max-min by repeatedly playing a no-regret online learning algorithm as the max-player against best-response for the min-player. In particular, for a fixed θ, minimizing over π is equivalent to finding optimal policy for scalarized MDP $\mathcal{M}_{-\theta}$. To achieve this, we can utilize a reward-free oracle as in Definition 4.2.1. On the other hand for θ-player we are able to use online gradient descent (Zinkevich, 2003). By combining ideas above, we obtain Algorithm 3.

Theorem 4.3.1. *There exists an absolute constant c, such that for any choice of RFE algorithm (Definition 4.2.1) and for any $\epsilon \in (0, H]$ and $\delta \in (0, 1]$, if we choose*

$$\begin{aligned}T &\geq c\big(H^2\iota/\epsilon^2\big),\\ K &\geq m_{\mathrm{RFE}}(\epsilon/2, \delta/2),\\ \eta^t &= \sqrt{1/(H^2 t)},\end{aligned}$$

where $\iota = \log(d/\delta)$; then, with probability at least $1-\delta$, Algorithm 3 outputs an ϵ-optimal policy for the approachability (Equation 4.2). Therefore, we have $m_{\mathrm{APP}}(\epsilon, \delta) \leq \mathcal{O}(m_{\mathrm{RFE}}(\epsilon/2, \delta/2) + H^2\iota/\epsilon^2)$.

Theorem 4.3.1 shows that given any reward-free algorithm, Algorithm 3 can solve the approachability task with negligible overhead. The proof for Theorem 4.3.1 is provided in Appendix C.2. Equipped

Algorithm 3 Meta-algorithm for VMDPs

Input: Reward-Free Algorithm RFE for VMDPs (as in Definiton 4.2.1), Target Set $\mathcal{C}$
Hyperparameters: learning rate η^t
Initialize: run exploration phase of RFE for K episodes
Set: $\theta^1 \in \mathcal{B}(1)$
for $t = 1, 2, \ldots, T$ **do**
  Obtain near optimal policy for $\mathcal{M}_{-\theta^t}$:

$$\pi^t \leftarrow \text{output of planning phase of RFE for preference vector } -\theta^t$$

  Estimate $\mathbf{V}_1^{\pi^t}(s_1)$ using one episode:

$$\text{Run } \pi^t \text{ for one episode and let } \widehat{\mathbf{v}}^t \text{ be the sum of vectorial returns}$$

  Apply online gradient ascent update for utility function $u^t(\theta) = \langle \theta, \widehat{\mathbf{v}}^t \rangle - \max_{\mathbf{x} \in \mathcal{C}} \langle \theta, \mathbf{x} \rangle$:

$$\theta^{t+1} \leftarrow \Gamma_{\mathcal{B}(1)}[\theta^t + \eta^t(\widehat{\mathbf{v}}^t - \text{argmax}_{\mathbf{x} \in \mathcal{C}} \langle \theta^t, \mathbf{x} \rangle)]$$

  where $\Gamma_{\mathcal{B}(1)}$ is the projection into Euclidean unit ball
Let π^{out} be uniform mixture of $\{\pi^1, \ldots, \pi^T\}$
Return π^{out}

with this theorem, since we have already shown the connection between approachability and constrained RL in Theorem 4.2.5, any results for RFE can be directly translated to results for constrained RL.

4.4 Tabular VMDPs

In this section, we consider tabular VMDPs; namely, we assume that $|\mathcal{S}| \leq S$ and $|\mathcal{A}| \leq A$. Utilizing prior work on tabular setting, we describe our choice of reward-free algorithm.

In the exploration phase, we use VI-Zero proposed by Liu et al. (2020). It can be seen as UCB-VI (Azar et al., 2017) with zero reward. Intuitively, the value function computed in the algorithm measures the level of uncertainty and incentivizes the greedy policy to visit underexplored states. The output of VI-Zero is $\widehat{\mathbb{P}}^{\text{out}}$, which is an estimation of the transition dynamics.

In the planning phase, given $\theta \in \mathcal{B}(1)$ we can use any planning algorithm (e.g., value iteration) for $\widehat{\mathcal{M}}_\theta = (\mathcal{S}, \mathcal{A}, H, \widehat{\mathbb{P}}^{\text{out}}, \langle \theta, \widehat{\mathbf{r}} \rangle)$ where $\widehat{\mathbf{r}}$ is empirical estimate of $\mathbf{r}$ using collected samples $\{\mathbf{r}_h^k\}$.

The following theorem state theoretical guarantees for tabular VMDPs. Proof of Theorem 4.4.1 and

more details can be found in Appendix C.3.

Theorem 4.4.1. *For tabular VMDP, we have a reward-free algorithm (Definiton 4.2.1) with* $m_{\mathrm{RFE}}(\epsilon, \delta) \leq \mathcal{O}(\min\{d, S\}H^4SA\iota/\epsilon^2 + H^3S^2A\iota^2/\epsilon)$*, an algorithm for approachability (Definition 4.2.3) with* $m_{\mathrm{APP}}(\epsilon, \delta) \leq \mathcal{O}(\min\{d, S\}H^4SA\iota/\epsilon^2 + H^3S^2A\iota^2/\epsilon)$ *, and an algorithm for CMDP (Definition 4.2.4) with* $m_{\mathrm{CMDP}}(\epsilon, \delta) \leq \mathcal{O}(\min\{d, S\}H^4SA\iota^2/\epsilon^2 + H^3S^2A\iota^3/\epsilon)$.

The reward-free algorithm with stated sample complexity in Theorem 4.4.1 is the VI-Zero algorithm (Algorithm 14 in Appendix C.3). Its sample complexity result is obtained by adapting the results in Liu et al. (2020) for scalar-valued MDPs to the settings of VMDPs. The algorithms for approachability and CMDP is based on pluging in VI-Zero into our meta algorithms, and the corresponding sample complexity results are obtained by applying Theorem 4.2.5 and our main result—Theorem 4.3.1.

Theorem 4.4.1 shows that the sample complexity of all three tasks are connected—the leading terms are all $\tilde{\mathcal{O}}(\min\{d, S\}H^4SA/\epsilon^2)$ which differ by only logarithmic factors. In particular, our sample complexity for the reward-free exploration (Definition 4.2.1) in the tabular setting matches the best result in Wu et al. (2020). It further shows that we can easily design an sample-efficient for approachability (Definition 4.2.3) and CMDP with general convex constraints (Definition 4.2.4) in the tabular setting, with sample complexity matching the best result in Yu et al. (2021) up to a single factor of H. [2] Therefore, our framework while being modular and enabling direct translation of reward-free RL to constrained RL, achieves sharp sample complexity guarantees. We comment that due to reward-free nature of our approach unlike Yu et al. (2021), we can no longer provide regret guarantees.

4.5 Linear function approximation: Linear VMDPs

In this section we consider the setting of linear function approximation and allow $\mathcal{S}$ and $\mathcal{A}$ to be infinitely large. We assume that agent has access to a feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_{\mathrm{lin}}}$ and the return function and transitions are linear functions of the feature map. We formally define the

[2] This H factor difference is due the Bernstein-type bonus used in Yu et al. (2021), which can not be adapted to the reward-free setting.

linear VMDPs in Assumption 4.5.1 which adapts the definition of linear MDPs (Jin et al., 2020c) for VMDPS; namely, they coincide for the case of $d = 1$.

Assumption 4.5.1 (Linear VMDP)**.** A VMDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, \mathbf{r})$ is said to be a linear with a feature map $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_{\text{lin}}}$, if for any $h \in [H]$:

1. There exists d_{lin} unknown (signed) measures $\boldsymbol{\mu}_h = \{\mu_h^{(1)}, \ldots, \mu_h^{(d_{\text{lin}})}\}$ over $\mathcal{S}$ such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have $\mathbb{P}_h(\cdot \mid s, a) = \langle \boldsymbol{\mu}(\cdot), \boldsymbol{\phi}(s, a) \rangle$.
2. There exists an unknown matrix $W_h \in \mathbb{R}^{d \times d_{\text{lin}}}$ such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have $\mathbf{r}_h(s, a) = W_h \boldsymbol{\phi}(s, a)$.

Similar to Jin et al. (2020c), we assume that $\|\boldsymbol{\phi}(s, a)\| \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\boldsymbol{\mu}_h(\mathcal{S})\| \leq \sqrt{d_{\text{lin}}}$ for all $h \in [H]$, and $\|W_h\| \leq \sqrt{d_{\text{lin}}}$ for all $h \in [H]$.

Wang et al. (2020a) has recently proposed a sample-efficient algorithm for reward-free exploration in linear MDPs. Utilizing that algorithm and tailoring it for our setting, we can obtain the following theoretical guarantee. The algorithm and the proof can be found in Appendix C.4.

Theorem 4.5.2. *For linear VMDPs (Assumption 4.5.1), we have a reward-free algorithm (Definiton 4.2.1) with* $m_{\text{RFE}}(\epsilon, \delta) \leq \mathcal{O}(d_{\text{lin}}^3 H^6 \iota^2 / \epsilon^2)$*, an approachability algorithm (Definition 4.2.3) with* $m_{\text{APP}}(\epsilon, \delta) \leq \mathcal{O}(d_{\text{lin}}^3 H^6 \iota^2 / \epsilon^2)$ *and an algorithm for CMDP (Definition 4.2.4) with* $m_{\text{CMDP}}(\epsilon, \delta) \leq \mathcal{O}(d_{\text{lin}}^3 H^6 \iota^3 / \epsilon^2)$.

The reward-free algorithm with stated sample complexity in Theorem 4.5.2 is the Algorithm 15 in Appendix C.4. it is a modified version of the reward-free algorithm introduced by Wang et al. (2020a). Its sample complexity result is again obtained by adapting the results in Wang et al. (2020a) for scalar-valued MDPs to the settings of VMDPs. The algorithms for approachability and CMDP is based on plugging in this reward-free algorithm into our meta algorithms, and the corresponding sample complexity results are obtained by applying Theorem 4.2.5 and our main result—Theorem 4.3.1.

Theorem 4.5.2 provides a new sample complexity result of $\tilde{\mathcal{O}}(d_{\text{lin}}^3 H^6 / \epsilon^2)$ for the reward-free ex-

ploration (Definition 4.2.1) in the linear setting (Assumption 4.5.1). It further provides a new sample complexity result of $\tilde{\mathcal{O}}(d_{\text{lin}}^3 H^6/\epsilon^2)$ for both approachability (Definition 4.2.3) and CMDP (Definition 4.2.4) in the linear setting (Assumption 4.5.1). To best our knowledge, this is the first sample-efficient result for constrained RL problems with linear function approximation and general convex constraints.

4.6 Vector-valued Markov games

4.6.1 Model and preliminaries

Similar to Section 4.2, we consider an episodic *vector-valued Markov game* (VMG) specified by a tuple $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathcal{B}, H, \mathbb{P}, \mathbf{r})$, where $\mathcal{A}$ and $\mathcal{B}$ are the action spaces for the min-player and max-player, respectively. The d-dimensional return function $\mathbf{r}$ and the transition probabilities $\mathbb{P}$, now depend on the current state and the action of both players.

Interaction protocol. In each episode, we start at a *fixed* initial state s_1. Then, at each step $h \in [H]$, both players observe the current state s_h, take their own actions $a_h \in \mathcal{A}$ and $b_h \in \mathcal{B}$ simultaneously, observe stochastic sample of the return vector $\mathbf{r}_h(s_h, a_h, b_h)$ along with their opponent's action, and it causes the environment to transit to $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, a_h, b_h)$. We assume that stochastic samples of the return function are also in $\mathcal{B}(1)$, almost surely.

Policy and value function. A policy μ of the min-player is a collection of H functions $\{\mu_h : \mathcal{S} \to \Delta(\mathcal{A})\}_{h=1}^H$. Similarly, a policy ν of the max-player is a collection of H functions $\{\nu_h : \mathcal{S} \to \Delta(\mathcal{B})\}_{h=1}^H$. If the players are following μ and ν, we have $a_h \sim \mu(\cdot|s)$ and $b_h \sim \nu(\cdot|s)$ at the h^{th} step. We use $\mathbf{V}_h^{\mu,\nu} : \mathcal{S} \to \mathcal{B}(H)$ and $\mathbf{Q}_h^{\mu,\nu} : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to \mathcal{B}(H)$ to denote the value function and Q-value function at step h under policies μ and ν.

Scalarized markov game and Nash equilibrium. For a VMG $\mathcal{G}$ and $\theta \in \mathcal{B}(1)$, we define scalar-valued Markov game $\mathcal{G}_\theta = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r_\theta)$, where $r_\theta = \{\langle \theta, \mathbf{r}_h \rangle : \mathcal{S} \times \mathcal{A} \times \mathcal{B} \to [-1, 1]\}_{h=1}^H$.

We use $V_h^{\mu,\nu}(\cdot;\theta)$ and $Q_h^{\mu,\nu}(\cdot,\cdot,\cdot;\theta)$ to denote value function and Q-value function of $\mathcal{G}_\theta$, respectively. Note that we have $V_h^{\mu,\nu}(s;\theta) = \langle \theta, \mathbf{V}_h^{\mu,\nu}(s) \rangle$ and $Q_h^{\mu,\nu}(s,a,b;\theta) = \langle \theta, \mathbf{Q}_h^{\mu,\nu}(s,a,b) \rangle$.

For any policy of the min-player μ, there exists a *best-response* policy $\nu_\dagger(\mu)$ of the max-player; i.e. $V_h^{\mu,\nu_\dagger(\mu)}(s;\theta) = \max_\nu V_h^{\mu,\nu}(s;\theta)$ for all $(s,h) \in \mathcal{S} \times [H]$. We use $V^{\mu,\dagger}$ to denote $V^{\mu,\nu_\dagger(\mu)}$. Similarly, we can define $\mu_\dagger(\nu)$ and $V^{\dagger,\nu}$. We further know (Filar and Vrieze, 2012) that there exist policies $(\mu^\star, \nu^\star)$, known as *Nash equilibrium*, satisfying the following equation for all $(s,h) \in \mathcal{S} \times [H]$:

$$\begin{aligned}
&\min_\mu \max_\nu V_h^{\mu,\nu}(s;\theta) \\
&= V_h^{\mu^\star,\dagger}(s;\theta) = V_h^{\mu^\star,\nu^\star}(s;\theta) = V_h^{\dagger,\nu^\star}(s;\theta) \\
&= \max_\nu \min_\mu V^{\mu,\nu}(s;\theta)
\end{aligned}$$

In words, it means that no player can gain anything by changing her own policy. We abbreviate $V_h^{\mu^\star,\nu^\star}$ and $Q_h^{\mu^\star,\nu^\star}$ as $V_h^\star$ and $Q_h^\star$.

Reward-free exploration (RFE) for VMGs

Similar to Section 4.2.1, we can define RFE algorithm for VMGs. Similarly, it consists of two phases. In the exploration phase, it explores the environment without guidance of return function. Later, in the planning phase, given any $\theta \in \mathcal{B}(1)$, it requires to output near optimal Nash equilibrium for $\mathcal{G}_\theta$.

Definition 4.6.1 (RFE algorithm for VMGs)**.** For any $\epsilon, \delta > 0$, after collecting $m_{\mathrm{RFE}}(\epsilon,\delta)$ episodes during the exploration phase, with probability at least $1-\delta$, the algorithm for all $\theta \in \mathcal{B}(1)$, satisfies

$$V_1^{\mu_\theta,\dagger}(s_1;\theta) - V_1^{\dagger,\nu_\theta}(s_1;\theta) \le \epsilon$$

where (μ_θ, ν_θ) is the output of the planning phase for vector θ as input. The function m_{RFE} determines the *sample complexity* of the RFE algorithm.

Blackwell approachability for VMGs

We assume we are given a VMG $\mathcal{G}$ and a target set $\mathcal{C}$. The goal of the min-player is for the return vector to lie in the set $\mathcal{C}$ while max-player wants the opposite. For the two-player vector-valued games it can be easily shown that the minimax theorem does no longer hold (see Section 2.1 of Abernethy et al. 2011). Namely, if for every policy of the max-player we have a response such that the return is in the set, we cannot hope to find a single policy for the min-player so that for every policy of the max-player the return vector lie in the set. However, approaching the set on average is possible.

Definition 4.6.2 (Blackwell approachability)**.** We say the min-player is approaching the target $\mathcal{C}$ with rate $\alpha(T)$, if for arbitrary sequence of max-player polices $\nu^1, \dots, \nu^T$, we have

$$\operatorname{dist}(\tfrac{1}{T}\textstyle\sum_{t=1}^{T} \mathbf{V}_1^{\mu^t,\nu^t}(s_1), \mathcal{C}) \leq \beta + \alpha(T),$$

where $\beta = \max_\nu \min_\mu \operatorname{dist}(\mathbf{V}_1^{\mu,\nu}(s_1), \mathcal{C})$.

4.6.2 Meta-algorithm for VMGs

Similar to Section 4.3, we introduce our main algorithmic framework for VMGs bridging reward-free algorithm and Blackwell approachability in VMGs. The pseudo-code is displayed in Algorithm 4 and the theoretical guarantees are provided in Theorem 4.6.3. The proof can be found in Appendix C.5.

Theorem 4.6.3. *For any choice of RFE algorithm (Definition 4.6.1) and for any $\epsilon \in (0, H]$ and $\delta \in (0, 1]$, if we choose $K = m_{\mathrm{RFE}}(\epsilon/2, \delta/2)$ and $\eta^t = \sqrt{1/H^2 t}$; then, with probability at least $1 - \delta$, the min-player in Algorithm 4, satisfies Definition 4.6.2 with rate $\alpha(T) = \mathcal{O}(\epsilon/2 + \sqrt{H^2\iota/T})$ where $\iota = \log(d/\delta)$. Therefore to obtain ϵ-optimality, the total sample complexity scales with $\mathcal{O}(m_{\mathrm{RFE}}(\epsilon/2, \delta/2) + H^2\iota/\epsilon^2)$.*

Algorithm 4 Meta-algorithm for VMGs

Input: Reward-Free Algorithm RFE for VMG (as in Definition 4.6.1), Target Set $\mathcal{C}$
Hyperparameters: learning rate η^t
Initialize: run exploration phase of RFE for K episodes
Set: $\theta^1 \in \mathcal{B}(1)$
for $t = 1, 2, \ldots, T$ **do**
  Obtain near optimal Nash equilibrium for $\mathcal{G}_{\theta^t}$:

$$(\mu^t, \omega^t) \leftarrow \text{output of planning phase of RFE for the vector } \theta^t \text{ as input}$$

  Play μ^t for one episode:

$$\text{Play } \mu^t \text{ against max-player playing arbitrary policy } \nu^t \text{ for one episode and let } \widehat{\mathbf{v}}^t \text{ be the sum of vectorial returns}$$

  Apply online gradient ascent update for utility function $u^t(\theta) = \langle \theta, \widehat{\mathbf{v}}^t \rangle - \max_{\mathbf{x} \in \mathcal{C}} \langle \theta, \mathbf{x} \rangle$:

$$\theta^{t+1} \leftarrow \Gamma_{\mathcal{B}(1)}[\theta^t + \eta^t(\widehat{\mathbf{v}}^t - \operatorname{argmax}_{\mathbf{x} \in \mathcal{C}} \langle \theta^t, \mathbf{x} \rangle)]$$

  where $\Gamma_{\mathcal{B}(1)}$ is the projection into Euclidean unit ball

4.6.3 Tabular VMGs

In this section, we consider tabular VMDPs; namely, we assume that $|\mathcal{S}| \leq S$, $|\mathcal{A}| \leq A$, and $|\mathcal{B}| \leq B$. Similar to Section 4.4, by utilizing VI-Zero (Liu et al., 2020) we can have the following theoretical guarantees. The algorithm and the proof can be found in Appendix C.5.

Theorem 4.6.4. *There exists a reward-free algorithm for tabular VMGs and a right choice of hyper-parameters that satisfies Definition 4.6.1 with sample complexity* $m_{\mathrm{RFE}}(\epsilon, \delta) \leq \mathcal{O}(\min\{d, S\}H^4SAB\iota/\epsilon^2 + H^3S^2AB\iota^2/\epsilon)$, *where* $\iota = \log[dSABH/(\epsilon\delta)]$.

The theorem provides a new sample complexity result of $\tilde{\mathcal{O}}(\min\{d, S\}H^4SAB\iota/\epsilon^2)$ for reward-free exploration in VMGs (Definition 4.6.1). It immediately follows from Theorem 4.6.4 and Theorem 4.6.3 that we can achieve total sample complexity of $\tilde{\mathcal{O}}(\min\{d, S\}H^4SAB\iota/\epsilon^2)$ for Blackwell approachability in VMGs (Definition 4.6.2). Our rate for $\alpha(T)$ scales with $\tilde{\mathcal{O}}(\sqrt{\mathrm{poly}(H)/T})$ while the results in Yu et al. (2021) has the rate of $\alpha(T)$ scaling with $\tilde{\mathcal{O}}(\sqrt{\mathrm{poly}(H)\min\{d, S\}SA/T})$. However, we require initial phase of self-play for $K = \mathcal{O}(m_{\mathrm{RFE}})$ episodes which is not needed by Yu et al. (2021).

4.7 Conclusion

This chapter provides a meta algorithm that takes a reward-free RL solver, and convert it to an algorithm for solving constrained RL problems. Our framework enables the direct translation of any progress in reward-free RL to constrained RL setting. Utilizing existing reward-free solvers, our framework provides sharp sample complexity results for constrained RL in tabular setting (matching best existing results up to factor of horizon dependence), new results for the linear function approximation setting. Our framework further extends to tabular two-player vector-valued Markov games for solving Blackwell approachability problem.

Part II

Reinforcement Learning with Function Approximation

Chapter 5

Bellman Eluder Dimension: New Rich Classes of RL Problems, and Sample-Efficient Algorithms

5.1 Introduction

Modern Reinforcement Learning (RL) commonly engages practical problems with an enormous number of states, where *function approximation* must be deployed to approximate the true value function using functions from a prespecified function class. Function approximation, especially based on deep neural networks, lies at the heart of the recent practical successes of RL in domains such as Atari (Mnih et al., 2015), Go (Silver et al., 2016), robotics (Kober et al., 2013), and dialogue systems (Li et al., 2016).

Despite its empirical success, RL with function approximation raises a new series of theoretical challenges when comparing to the classic tabular RL: (1) *generalization*, to generalize knowledge from the visited states to the unvisited states due to the enormous state space. (2) *limited expressiveness*,

to handle the complicated issues where true value functions or intermediate steps computed in the algorithm can be functions outside the prespecified function class. (3) *exploration*, to address the tradeoff between exploration and exploitation when above challenges are present.

Consequently, most existing theoretical results on efficient RL with function approximation rely on relatively strong structural assumptions. For instance, many require that the MDP admits a linear approximation (Wang et al., 2019; Jin et al., 2020c; Zanette et al., 2020a), or that the model is precisely Linear Quadratic Regulator (LQR) (Anderson and Moore, 2007; Fazel et al., 2018; Dean et al., 2019). Most of these structural assumptions rarely hold in practical applications. This naturally leads to one of the most fundamental questions in RL.

What are the minimal structural assumptions that empower sample-efficient RL?

We advance our understanding of this grand question via the following two steps: (1) identify a rich class of RL problems (with weak structural assumptions) that cover many practical applications of interests; (2) design sample-efficient algorithms that provably learn any RL problem in this class.

The attempts to find weak or minimal structural assumptions that allow statistical learning can be traced in supervised learning where VC dimension (Vapnik, 2013) or Rademacher complexity (Bartlett and Mendelson, 2002) is proposed, or in online learning where Littlestone dimension (Littlestone, 1988) or sequential Rademacher complexity (Rakhlin et al., 2010) is developed.

In the area of reinforcement learning, there are two intriguing lines of recent works that have made significant progress in this direction. To begin with, Jiang et al. (2017) introduces a generic complexity notion—Bellman rank, which can be proved small for many RL problems including linear MDPs (Jin et al., 2020c), reactive POMDPs (Krishnamurthy et al., 2016), etc. Jiang et al. (2017) further propose an hypothesis elimination-based algorithm—OLIVE for sample-efficient learning of problems with low Bellman rank. On the other hand, recent work by Wang et al. (2020b) considers general function approximation with low Eluder dimension (Russo and Van Roy, 2013), and designs a UCB-style algorithm with regret guarantee. Noticeably, generalized linear MDPs (Wang et al., 2019) and kernel MDPs (see Appendix D.3) are subclasses of low Eluder dimension problems, but

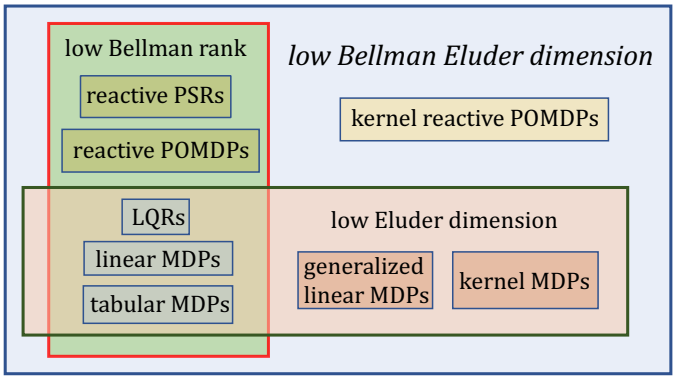


Figure 5.1: A schematic summarizing relations among families of RL problems[1]

not low Bellman rank.

In this paper, we make the following three contributions.

- We introduce a new complexity measure for RL—Bellman Eluder (BE) dimension. We prove that the family of RL problems of low BE dimension is remarkably rich, which subsumes both low Bellman rank problems and low Eluder dimension problems—two arguably most generic tractable function classes so far in the literature (see Figure 5.1). The family of low BE dimension further includes new problems such as kernel reactive POMDPs (see Appendix D.3) which were not known to be sample-efficiently learnable.
- We design a new optimization-based algorithm—GOLF, which provably learns near-optimal policies of low BE dimension problems in a number of samples that is polynomial in all relevant parameters, but independent of the size of state-action space. Our regret or sample complexity guarantees match Zanette et al. (2020a) which is minimax optimal when specified to the linear setting. Our rates further improve upon Jiang et al. (2017); Wang et al. (2020b) in low Bellman rank and low Eluder dimension settings, respectively.
- We reanalyze the hypothesis elimination based algorithm—OLIVE proposed in Jiang et al. (2017). We show it can also learn RL problems with low BE dimension sample-efficiently,

[1]The family of low Bellman rank problems and low Bellman Eluder dimension problems include both Q-type and V-type variants. Please refer to Section 5.3.1 and Appendix D.2 for more details.

under slightly weaker assumptions but with worse sample complexity comparing to GOLF.

5.1.1 Related works

This section reviews prior theoretical works on RL, under Markov Decision Process (MDP) models.

We remark that there has been a long line of research on function approximation in the *batch RL* setting (see, e.g., Szepesvári and Munos, 2005; Munos and Szepesvári, 2008; Chen and Jiang, 2019; Xie and Jiang, 2020). In this setting, agents are provided with exploratory data or simulator, so that they do not need to explicitly address the challenge of exploration. In this paper, we do not make such assumption, and attack the exploration problem directly. In the following we focus exclusively on the RL results in the general setting where exploration is required.

Tabular RL. Tabular RL concerns MDPs with a small number of states and actions, which has been thoroughly studied in recent years (see, e.g., Brafman and Tennenholtz, 2002; Jaksch et al., 2010; Dann and Brunskill, 2015; Agrawal and Jia, 2017; Azar et al., 2017; Zanette and Brunskill, 2019; Jin et al., 2018; Zhang et al., 2020b). In the episodic setting with non-stationary dynamics, the best regret bound $\tilde{\mathcal{O}}(\sqrt{H^2|\mathcal{S}||\mathcal{A}|T})$ is achieved by both model-based (Azar et al., 2017) and model-free (Zhang et al., 2020b) algorithms. Moreover, the bound is proved to be minimax-optimal (Jin et al., 2018; Domingues et al., 2021). This minimax bound suggests that when the state-action space is enormous, RL is information-theoretically hard without further structural assumptions.

RL with linear function approximation. A recent line of work studies RL with linear function approximation (see, e.g., Jin et al., 2020c; Wang et al., 2019; Cai et al., 2019; Zanette et al., 2020a,b; Agarwal et al., 2020; Neu and Pike-Burke, 2020; Sun et al., 2019a) These papers assume certain completeness conditions, as well as the optimal value function can be well approximated by linear functions. Under one formulation of linear approximation, the minimax regret bound $\tilde{\mathcal{O}}(d\sqrt{T})$ is achieved by algorithm ELEANOR (Zanette et al., 2020a), where d is the ambient dimension of the feature space.

RL with general function approximation. Beyond the linear setting, there is a flurry line of research studying RL with general function approximation (see, e.g., Osband and Van Roy, 2014; Jiang et al., 2017; Sun et al., 2019a; Dong et al., 2020; Wang et al., 2020b; Yang et al., 2020; Foster et al., 2020). Among them, Jiang et al. (2017) and Wang et al. (2020b) are the closest to our work.

Jiang et al. (2017) propose a complexity measure named Bellman rank and design an algorithm OLIVE with PAC guarantees for problems with low Bellman rank. We note that low Bellman rank is a special case of low BE dimension. When specialized to the low Bellman rank setting, our result for OLIVE exactly matches the guarantee in Jiang et al. (2017). Our result for GOLF requires an additional completeness assumption, but provides sharper sample complexity guarantee.

Wang et al. (2020b) propose a UCB-type algorithm with a regret guarantee under the assumption that the function class has a low eluder dimension. Again, we will show that low Eluder dimension is a special case of low BE dimension. Comparing to Wang et al. (2020b), our algorithm GOLF works under a weaker completeness assumption, with a better regret guarantee.

Finally, we remark that the algorithms proposed in (Jiang et al., 2017; Zanette et al., 2020b; Du et al., 2021) and this paper are all computationally inefficient in general. We notice several existing works (e.g., Jin et al., 2020c; Wang et al., 2020b) can be computationally efficient given suitable regression oracles but they require stronger representation conditions and also achieve worse regret guarantees.

Relation to bilinear classes Concurrent to this work, Du et al. (2021) propose a new general tractable class of RL problems—bilinear class with low effective dimension (also known as low critical information gain in Du et al. (2021)). We comment on the similarities and differences between two works as follows.

In terms of algorithms, both Algorithm 18 in this paper and the algorithm proposed in Du et al. (2021) are based on OLIVE originally proposed in Jiang et al. (2017). The two algorithms share similar guarantees in terms of assumptions and complexity results. More importantly, our work further develops a new type of algorithm for general function approximation—GOLF, a natural

and clean algorithm which can be viewed as an optimistic version of classical algorithm—Fitted Q-Iteration (Szepesvári, 2010). GOLF gives much sharper sample complexity guarantees compared to Du et al. (2021) for various settings, and is minimax-optimal when applied to the linear setting (Zanette et al., 2020a).

In terms of richness of new classes identified, it depends on (a) what structure of MDP the complexity measures are applied to, and (b) what complexity measures are used. For (a), BE dimension applies to the Bellman error, while the bilinear class allows general surrogate losses of the Bellman error. For (b), this paper uses Eluder dimension while Du et al. (2021) uses effective dimension. It can be shown that low effective dimension always implies low Eluder dimension (see Appendix D.3.2). In short, Du et al. (2021) is more general in (a), while our work is more general in (b). As a result, neither work fully captures the other.

In particular, our BE framework covers a majority of the examples identified in Du et al. (2021) including low occupancy complexity, linear $Q^\star/V^\star$, $Q^\star$ state aggregation, feature selection/FLAMBE. Nevertheless, our work can not address examples with model-based function approximation (e.g., low witness rank Sun et al. (2019a)) while Du et al. (2021) can. On the other hand, Du et al. (2021) can not address the class of RL problems with low Eluder dimension (Wang et al., 2020b) while our work can. Moreover, for several classes of RL problems that both works cover, our complexity measure is sharper. For example, in the setting of function approximation with generalized linear functions, the BE dimension is $\tilde{O}(d)$ where d is the ambient dimension of the feature vectors, while the effective dimension under the generalized bilinear framework of Du et al. (2021) is at least $\tilde{\Omega}(d^2)$.

5.1.2 Chapter organization

We present preliminaries in Section 5.2, the definition of Bellman-Eluder dimension as well as its relations to existing complexity notions in Section 5.3. We present the results for algorithm GOLF in Section 5.4, and conclude in Section 5.5. Due to space limit, we postpone the results for algorithm OLIVE to Appendix D.1. Further discussions on Q-type versus V-type variants of BE dimension, as well as the practical examples will be provided in Appendix D.2 and D.3. All the proofs are

postponed to the appendix.

5.2 Preliminaries

We consider episodic Markov Decision Process (MDP), denoted by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, H is the number of steps in each episode, $\mathbb{P} = \{\mathbb{P}_h\}_{h\in[H]}$ is the collection of transition measures with $\mathbb{P}_h(s' \mid s, a)$ equal to the probability of transiting to s' after taking action a at state s at the h^{th} step, and $r = \{r_h\}_{h\in[H]}$ is the collection of reward functions with $r_h(s, a)$ equal to the deterministic reward received after taking action a at state s at the h^{th} step. [2] Throughout this chapter, we assume reward is non-negative, and $\sum_{h=1}^{H} r_h(s_h, a_h) \le 1$ for all possible sequence $(s_1, a_1, \ldots, s_H, a_H)$.

In each episode, the agent starts at a *fixed* initial state s_1. Then, at each step $h \in [H]$, the agent observes its current state s_h, takes action a_h, receives reward $r_h(s_h, a_h)$, and causes the environment to transit to $s_{h+1} \sim \mathbb{P}_h(\cdot \mid s_h, a_h)$. Without loss of generality, we assume there is a terminating state s_{end} which the environment will *always* transit to at step $H + 1$, and the episode terminates when s_{end} is reached.

Policy and value functions A (deterministic) policy π is a collection of H functions $\{\pi_h : \mathcal{S} \to \mathcal{A}\}_{h=1}^{H}$. We denote $V_h^\pi : \mathcal{S} \to \mathbb{R}$ as the value function at step h for policy π, so that $V_h^\pi(s)$ gives the expected sum of the remaining rewards received under policy π, starting from $s_h = s$, till the end of the episode. In symbol,

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

Similarly, we denote $Q_h^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as the Q-value function at step h for policy π, where

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

There exists an optimal policy $\pi^\star$, which gives the optimal value function for all states (Puterman,

[2] We study deterministic reward for notational simplicity. Our results readily generalize to random rewards.

2014), in the sense, $V_h^{\pi^\star}(s) = \sup_\pi V_h^\pi(s)$ for all $h \in [H]$ and $s \in \mathcal{S}$. For notational simplicity, we abbreviate $V^{\pi^\star}$ as $V^\star$. We similarly define the optimal Q-value function as $Q^\star$. Recall that $Q^\star$ satisfies the Bellman optimality equation:

$$Q_h^\star(s,a) = (\mathcal{T}_h Q_{h+1}^\star)(s,a) := r_h(s,a) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} \max_{a' \in \mathcal{A}} Q_{h+1}^\star(s', a'). \tag{5.1}$$

for all $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$. We also call $\mathcal{T}_h$ the *Bellman operator* at step h.

ϵ-optimality and regret We say a policy π is ϵ-optimal if $V_1^\pi(s_1) \geq V_1^\star(s_1) - \epsilon$. Suppose an agent interacts with the environment for K episodes. Denote by π^k the policy the agent follows in episode $k \in [K]$. The (accumulative) regret is defined as

$$\mathrm{Reg}(K) := \sum_{k=1}^{K} [V_1^\star(s_1) - V_1^{\pi^k}(s_1)].$$

The objective of reinforcement learning is to find an ϵ-optimal policy within a small number of interactions or to achieve sublinear regret.

5.2.1 Function approximation

In this work, we consider reinforcement learning with value function approximation. Formally, the learner is given a function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$, where $\mathcal{F}_h \subseteq (\mathcal{S} \times \mathcal{A} \to [0,1])$ offers a set of candidate functions to approximate $Q_h^\star$—the optimal Q-value function at step h. Since no reward is collected in the $(H+1)^{\text{th}}$ steps, we always set $f_{H+1} = 0$.

Reinforcement learning with function approximation in general is extremely challenging without further assumptions (see, e.g., hardness results in Krishnamurthy et al. (2016); Weisz et al. (2021)). Below, we present two assumptions about function approximation that are commonly adopted in the literature.

Assumption 5.2.1 (Realizability)**.** $Q_h^\star \in \mathcal{F}_h$ for all $h \in [H]$.

Realizability requires the function class is well-specified, i.e., function class $\mathcal{F}$ in fact contains the

optimal Q-value function $Q^\star$ with no approximation error.

Assumption 5.2.2 (Completeness)**.** $\mathcal{T}_h\mathcal{F}_{h+1} \subseteq \mathcal{F}_h$ for all $h \in [H]$.

Note $\mathcal{T}_h\mathcal{F}_{h+1}$ is defined as $\{\mathcal{T}_h f_{h+1} : f_{h+1} \in \mathcal{F}_{h+1}\}$. Completeness requires the function class $\mathcal{F}$ to be closed under the Bellman operator.

When function class $\mathcal{F}$ has finite elements, we can use its cardinality $|\mathcal{F}|$ to measure the "size" of function class $\mathcal{F}$. When addressing function classes with infinite elements, we need a notion similar to cardinality. We use the standard ϵ-covering number.

Definition 5.2.3 (ϵ-covering number)**.** The ϵ-covering number of a set $\mathcal{V}$ under metric ρ, denoted as $\mathcal{N}(\mathcal{V}, \epsilon, \rho)$, is the minimum integer n such that there exists a subset $\mathcal{V}_o \subset \mathcal{V}$ with $|\mathcal{V}_o| = n$, and for any $x \in \mathcal{V}$, there exists $y \in \mathcal{V}_o$ such that $\rho(x, y) \leq \epsilon$.

We refer readers to standard textbooks (see, e.g., Wainwright, 2019) for further properties of covering number. In this chapter, we will always apply the covering number on function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$, and use metric $\rho(f, g) = \max_h \|f_h - g_h\|_\infty$. For notational simplicity, we omit the metric dependence and denote the covering number as $\mathcal{N}_\mathcal{F}(\epsilon)$.

5.2.2 Eluder dimension

One class of functions highly related to this work is the function class of low Eluder dimension (Russo and Van Roy, 2013).

Definition 5.2.4 (ϵ-independence between points)**.** Let $\mathcal{G}$ be a function class defined on $\mathcal{X}$, and $z, x_1, x_2, \ldots, x_n \in \mathcal{X}$. We say z is ϵ-independent of $\{x_1, x_2, \ldots, x_n\}$ with respect to $\mathcal{G}$ if there exist $g_1, g_2 \in \mathcal{G}$ such that $\sqrt{\sum_{i=1}^n (g_1(x_i) - g_2(x_i))^2} \leq \epsilon$, but $g_1(z) - g_2(z) > \epsilon$.

Intuitively, z is independent of $\{x_1, x_2, \ldots, x_n\}$ means if that there exist two "certifying" functions g_1 and g_2, so that their function values are similar at all points $\{x_i\}_{i=1}^n$, but the values are rather different at z. This independence relation naturally induces the following complexity measure.

Definition 5.2.5 (Eluder dimension)**.** Let $\mathcal{G}$ be a function class defined on $\mathcal{X}$. The Eluder dimension

$\dim_{\mathrm{E}}(\mathcal{G}, \epsilon)$ is the length of the longest sequence $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ such that there exists $\epsilon' \geq \epsilon$ where x_i is ϵ'-independent of $\{x_1, \ldots, x_{i-1}\}$ for all $i \in [n]$.

Recall that a vector space has dimension d if and only if d is the length of the longest sequence of elements $\{x_1, \ldots, x_d\}$ such that x_i is linearly independent of $\{x_1, \ldots, x_{i-1}\}$ for all $i \in [n]$. Eluder dimension generalizes the linear independence relation in standard vector space to capture both nonlinear independence and approximate independence, and thus is more general.

5.3 Bellman Eluder Dimension

In this section, we introduce our new complexity measure—Bellman Eluder (BE) dimension. As one of its most important properties, we will show that the family of problems with low BE dimension contains the two existing most general tractable problem classes in RL—problems with low Bellman rank, and problems with low Eluder dimension (see Figure 5.1).

We start by developing a new distributional version of the original Eluder dimension proposed by Russo and Van Roy (2013) (see Section 5.2.2 for more details).

Definition 5.3.1 (ϵ-independence between distributions)**.** Let $\mathcal{G}$ be a function class defined on $\mathcal{X}$, and $\nu, \mu_1, \ldots, \mu_n$ be probability measures over $\mathcal{X}$. We say ν is ϵ-independent of $\{\mu_1, \mu_2, \ldots, \mu_n\}$ with respect to $\mathcal{G}$ if there exists $g \in \mathcal{G}$ such that $\sqrt{\sum_{i=1}^n (\mathbb{E}_{\mu_i}[g])^2} \leq \epsilon$, but $|\mathbb{E}_{\nu}[g]| > \epsilon$.

Definition 5.3.2 (Distributional Eluder (DE) dimension)**.** Let $\mathcal{G}$ be a function class defined on $\mathcal{X}$, and Π be a family of probability measures over $\mathcal{X}$. The distributional Eluder dimension $\dim_{\mathrm{DE}}(\mathcal{G}, \Pi, \epsilon)$ is the length of the longest sequence $\{\rho_1, \ldots, \rho_n\} \subset \Pi$ such that there exists $\epsilon' \geq \epsilon$ where ρ_i is ϵ'-independent of $\{\rho_1, \ldots, \rho_{i-1}\}$ for all $i \in [n]$.

Definition 5.3.1 and Definition 5.3.2 generalize Definition 5.2.4 and Definition 5.2.5 to their distributional versions, by inspecting the expected values of functions instead of the function values at points, and by restricting the candidate distributions to a certain family Π. The main advantage of this generalization is exactly in the statistical setting, where estimating the expected values of functions with respect to a certain distribution family can be easier than estimating function values

at each point (which is the case for RL in large state spaces).

It is clear that the standard Eluder dimension is a special case of the distributional Eluder dimension, because if we choose $\Pi = \{\delta_x(\cdot) \mid x \in \mathcal{X}\}$ where $\delta_x(\cdot)$ is the dirac measure centered at x, then $\dim_{\mathrm{E}}(\mathcal{G}, \epsilon) = \dim_{\mathrm{DE}}(\mathcal{G} - \mathcal{G}, \Pi, \epsilon)$ where $\mathcal{G} - \mathcal{G} = \{g_1 - g_2 : \ g_1, g_2 \in \mathcal{G}\}$.

Now we are ready to introduce the key notion in this chapter—Bellman Eluder dimension.

Definition 5.3.3 (Bellman Eluder (BE) dimension)**.** Let $(I - \mathcal{T}_h)\mathcal{F} := \{f_h - \mathcal{T}_h f_{h+1} : \ f \in \mathcal{F}\}$ be the set of Bellman residuals induced by $\mathcal{F}$ at step h, and $\Pi = \{\Pi_h\}_{h=1}^H$ be a collection of H probability measure families over $\mathcal{S} \times \mathcal{A}$. The ϵ-Bellman Eluder of $\mathcal{F}$ with respect to Π is defined as

$$\dim_{\mathrm{BE}}(\mathcal{F}, \Pi, \epsilon) := \max_{h \in [H]} \dim_{\mathrm{DE}}\big((I - \mathcal{T}_h)\mathcal{F}, \Pi_h, \epsilon\big).$$

Remark 5.3.4 (Q-type v.s. V-type). Definition 5.3.3 is based on the Bellman residuals functions that take a state-action pair as input, thus referred to as Q-type BE dimension. Alternatively, one can define V-type BE dimension using a different set of Bellman residual functions that depend on states only (see Appendix D.2). We focus on Q-type in the main here, and present the results for V-type in Appendix D.2. Both variants are important, and they include different sets of examples (see Appendix D.2, D.3).

In short, Bellman Eluder dimension is simply the distributional Eluder dimension on the function class of Bellman residuals, maximizing over all steps. In addition to function class $\mathcal{F}$ and error ϵ, Bellman Eluder dimension also depends on the choice of distribution family Π. For the purpose of this chapter, we focus on the following two specific choices.

1. $\mathcal{D}_{\mathcal{F}} := \{\mathcal{D}_{\mathcal{F},h}\}_{h \in [H]}$, where $\mathcal{D}_{\mathcal{F},h}$ denotes the collection of all probability measures over $\mathcal{S} \times \mathcal{A}$ at the h^{th} step, which can be generated by executing the greedy policy π_f induced by any $f \in \mathcal{F}$, i.e., $\pi_{f,h}(\cdot) = \operatorname{argmax}_{a \in \mathcal{A}} f_h(\cdot, a)$ for all $h \in [H]$.

2. $\mathcal{D}_{\Delta} := \{\mathcal{D}_{\Delta,h}\}_{h \in [H]}$, where $\mathcal{D}_{\Delta,h} = \{\delta_{(s,a)}(\cdot) | s \in \mathcal{S}, a \in \mathcal{A}\}$, i.e., the collections of probability measures that put measure 1 on a single state-action pair.

We say a RL problem has low BE dimension if $\min_{\Pi \in \{\mathcal{D}_{\mathcal{F}}, \mathcal{D}_{\Delta}\}} \dim_{\mathrm{BE}}(\mathcal{F}, \Pi, \epsilon)$ is small.

5.3.1 Relations with known tractable classes of RL problems

Known tractable problem classes in RL include but not limited to tabular MDPs, linear MDPs (Jin et al., 2020c), linear quadratic regulators (Anderson and Moore, 2007), generalized linear MDPs (Wang et al., 2019), kernel MDPs (Appendix D.3), reactive POMDPs (Krishnamurthy et al., 2016), reactive PSRs (Singh et al., 2012; Jiang et al., 2017). There are two existing generic tractable problem classes that jointly contain all the examples mentioned above: the set of RL problems with low Bellman rank, and the set of RL problems with low Eluder dimension. However, for these two generic sets, one does not contain the other.

In this section, we will show that our new class of RL problems with low BE dimension in fact contains both low Bellman rank problems and low Eluder dimension problems (see Figure 5.1). That is, our new problem class covers almost all existing tractable RL problems, and to our best knowledge, is the most generic tractable function class so far.

Relation with low Bellman rank The seminal paper by Jiang et al. (2017) proposes the complexity measure—Bellman rank, and shows that a majority of RL examples mentioned above have low Bellman rank. They also propose a hypothesis elimination based algorithm—OLIVE, that learns any low Bellman rank problem within polynomial samples. Formally,

Definition 5.3.5 (Bellman rank)**.** The Bellman rank is the minimum integer d so that there exists $\phi_h : \mathcal{F} \to \mathbb{R}^d$ and $\psi_h : \mathcal{F} \to \mathbb{R}^d$ for each $h \in [H]$, such that for any $f, f' \in \mathcal{F}$, the average Bellman error.

$$\mathcal{E}(f, \pi_{f'}, h) := \mathbb{E}_{\pi_{f'}}[(f_h - \mathcal{T}_h f_{h+1})(s_h, a_h)] = \langle \phi_h(f), \psi_h(f') \rangle,$$

where $\|\phi_h(f)\|_2 \cdot \|\psi_h(f')\|_2 \leq \zeta$, and ζ is the normalization parameter.

We remark that similar to Bellman Eluder dimension, Bellman rank also has two variants—Q-type (Definition 5.3.5) and V-type (see Appendix D.2). Recall that we use π_f to denote the greedy policy induced by value function f. Intuitively, a problem with Bellman rank says its average Bellman error

can be decomposed as the inner product of two d-dimensional vectors, where one vector depends on the roll-in policy $\pi_{f'}$, while the other vector depends on the value function f. At a high level, it claims that the average Bellman error has a linear inner product structure.

Proposition 5.3.6 (low Bellman rank $\subset$ low BE dimension)**.** *If an MDP with function class $\mathcal{F}$ has Bellman rank d with normalization parameter ζ, then*

$$\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \leq \mathcal{O}(1 + d \log(1 + \zeta/\epsilon)).$$

Proposition 5.3.6 claims that problems with low Bellman rank also have low BE dimension, with a small multiplicative factor that is only logarithmic in ζ and ϵ^{-1}.

Relation with low Eluder dimension Wang et al. (2020b) study the setting where the function class $\mathcal{F}$ has low Eluder dimension, which includes generalized linear functions. They prove that, when the completeness assumption is satisfied,[3] low Eluder dimension problems can be efficiently learned in polynomial samples.

Proposition 5.3.7 (low Eluder dimension $\subset$ low BE dimension)**.** *Assume $\mathcal{F}$ satisfies completeness (Assumption 5.2.2). Then for all $\epsilon > 0$,*

$$\dim_{\mathrm{BE}}\left(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon\right) \leq \max_{h \in [H]} \dim_{\mathrm{E}}(\mathcal{F}_h, \epsilon).$$

Proposition 5.3.7 asserts that problems with low Eluder dimension also have low BE dimension, which is a natural consequence of completeness and the fact that Eluder dimension is a special case of distributional Eluder dimension.

Finally, we show that the set of low BE dimension problems is strictly larger than the union of low Eluder dimension problems and low Bellman rank problems.

Proposition 5.3.8 (low BE dimension $\not\subset$ low Eluder dimension $\cup$ low Bellman rank)**.** *For any $m \in$*

[3] Wang et al. (2020b) assume for any function g (not necessarily in $\mathcal{F}$), $\mathcal{T}g \in \mathcal{F}$, which is stronger than the completeness assumption presented in this work (Assumption 5.2.2).

Algorithm 5 GOLF$(\mathcal{F}, \mathcal{G}, K, \beta)$ — **G**lobal **O**ptimism based on **L**ocal **F**itting

Initialize: $\mathcal{D}_1, \ldots, \mathcal{D}_H \leftarrow \emptyset$, $\mathcal{B}^0 \leftarrow \mathcal{F}$.
for episode k from 1 to K **do**
  Choose policy $\pi^k = \pi_{f^k}$, where $f^k = \operatorname{argmax}_{f \in \mathcal{B}^{k-1}} f(s_1, \pi_f(s_1))$.
  Collect a trajectory $(s_1, a_1, r_1, \ldots, s_H, a_H, r_H, s_{H+1})$ by following π^k.
  Augment $\mathcal{D}_h = \mathcal{D}_h \cup \{(s_h, a_h, r_h, s_{h+1})\}$ for all $h \in [H]$.
  Update

$$\mathcal{B}^k = \left\{ f \in \mathcal{F} : \ \mathcal{L}_{\mathcal{D}_h}(f_h, f_{h+1}) \le \inf_{g \in \mathcal{G}_h} \mathcal{L}_{\mathcal{D}_h}(g, f_{h+1}) + \beta \text{ for all } h \in [H] \right\},$$

$$\text{where } \mathcal{L}_{\mathcal{D}_h}(\xi_h, \zeta_{h+1}) = \sum_{(s,a,r,s') \in \mathcal{D}_h} [\xi_h(s,a) - r - \max_{a' \in \mathcal{A}} \zeta_{h+1}(s', a')]^2. \tag{5.2}$$

Output π^{out} sampled uniformly at random from $\{\pi^k\}_{k=1}^K$.

$\mathbb{N}^+$, *there exists an MDP and a function class* $\mathcal{F}$ *so that for all* $\epsilon \in (0, 1]$, *we have* $\dim_{\text{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) = \dim_{\text{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon) \le 5$, *but* $\min\{\min_{h \in [H]} \dim_{\text{E}}(\mathcal{F}_h, \epsilon), \text{Bellman rank}\} \ge m$.

In particular, the family of low BE dimension includes new examples such as kernel reactive POMDPs (Appendix D.3), which can not be addressed by the framework of either Bellman rank or Eluder dimension.

5.4 Algorithm Golf

Section 5.3 defines a new class of RL problems with low BE dimension, and shows that the new class is rich, containing almost all the existing known tractable RL problems so far. In this section, we propose a new simple optimization-based algorithm—**G**lobal **O**ptimism based on **L**ocal **F**itting (GOLF). We prove that, low BE dimension problems are indeed tractable, i.e., GOLF can find near-optimal policies for these problems within a polynomial number of samples.

At a high level, GOLF can be viewed as an optimistic version of the classic algorithm—Fitted Q-Iteration (FQI) (Szepesvári, 2010). GOLF generalizes the ELEANOR algorithm (Zanette et al., 2020a) from the special linear setting to the general setting with arbitrary function classes.

The pseudocode of GOLF is given in Algorithm 5. GOLF initializes datasets $\{\mathcal{D}_h\}_{h=1}^H$ to be empty sets, and confidence set $\mathcal{B}^0$ to be $\mathcal{F}$. Then, in each episode, GOLF performs two main steps:

- Line 3 (Optimistic planning): compute the most optimistic value function f^k from the confidence set $\mathcal{B}^{k-1}$ constructed in the last episode , and choose π^k to be its greedy policy.
- Line 4-6 (Execute the policy and update the confidence set): execute policy π^k for one episode, collect data, and update the confidence set using the new data.

At the heart of GOLF is the way we construct the confidence set $\mathcal{B}^k$. For each $h \in [H]$, GOLF maintains a *local* regression constraint using the collected transition data $\mathcal{D}_h$ at this step

$$\mathcal{L}_{\mathcal{D}_h}(f_h, f_{h+1}) \leq \inf_{g \in \mathcal{G}_h} \mathcal{L}_{\mathcal{D}_h}(g, f_{h+1}) + \beta, \tag{5.3}$$

where β is a confidence parameter, and $\mathcal{L}_{\mathcal{D}_h}$ is the squared loss defined in Eq. (5.2), which can be viewed as a proxy to the squared Bellman error at step h. We remark that FQI algorithm (Szepesvári, 2010) simply updates $f_h \leftarrow \operatorname{argmin}_{\phi \in \mathcal{F}_h} \mathcal{L}_{\mathcal{D}_h}(\phi, f_{h+1})$. Our constraint Eq. (5.3) can be viewed as a relaxed version of this update, which allows f_h to be not only the minimizer of the loss $\mathcal{L}_{\mathcal{D}_h}(\cdot, f_{h+1})$, but also any function whose loss is only slightly larger than the optimal loss over the auxiliary function class $\mathcal{G}_h$.

We remark that in general, the optimization problem in Line 3 of GOLF can not be solved computationally efficiently.

5.4.1 Theoretical guarantees

In this subsection, we present the theoretical guarantees for GOLF, which hold under Assumption 5.2.1 (realizability) and the following generalized completeness assumption introduced in Antos et al. (2008); Chen and Jiang (2019). Let $\mathcal{G} = \mathcal{G}_1 \times \cdots \times \mathcal{G}_H$ be an auxiliary function class provided to the learner where each $\mathcal{G}_h \subseteq (\mathcal{S} \times \mathcal{A} \to [0, 1])$. Generalized completeness requires the auxiliary function class $\mathcal{G}$ to be rich enough so that applying Bellman operator to any function in the primary function class $\mathcal{F}$ will end up in $\mathcal{G}$.

Assumption 5.4.1 (Generalized completeness)**.** $\mathcal{T}_h \mathcal{F}_{h+1} \subseteq \mathcal{G}_h$ for all $h \in [H]$.

If we choose $\mathcal{G} = \mathcal{F}$, then Assumption 5.4.1 is equivalent to the standard completeness assumption

(Assumption 5.2.2). Now, we are ready to present the main theorem for GOLF.

Theorem 5.4.2 (Regret of GOLF)**.** *Under Assumption 5.2.1, 5.4.1, there exists an absolute constant c such that for any $\delta \in (0,1]$, $K \in \mathbb{N}$, if we choose parameter $\beta = c\log[\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(1/K)\cdot KH/\delta]$ in* GOLF*, then with probability at least $1-\delta$, for all $k \in [K]$, we have*

$$\mathrm{Reg}(k) = \sum_{t=1}^{k}\left[V_1^\star(s_1) - V_1^{\pi^t}(s_1)\right] \le \mathcal{O}(H\sqrt{dk\beta}),$$

where $d = \min_{\Pi\in\{\mathcal{P}_\Delta,\mathcal{P}_\mathcal{F}\}} \dim_{\mathrm{BE}}\left(\mathcal{F},\Pi,1/\sqrt{K}\right)$ is the BE dimension.

Theorem 5.4.2 asserts that, under the realizability and completeness assumptions, the general class of RL problems with low BE dimension is indeed tractable: there exists an algorithm (GOLF) that can achieve $\sqrt{K}$ regret, whose multiplicative factor depends only polynomially on the horizon of MDP H, the BE dimension d, and the log covering number of the two function classes. Most importantly, the regret is independent of the number of the states, which is crucial for dealing with practical RL problems with function approximation, where the state spaces are typically exponentially large.

We remark that when function class $\mathcal{F}\cup\mathcal{G}$ has finite number of elements, its covering number is upper bounded by its cardinality $|\mathcal{F}\cup\mathcal{G}|$. For a wide range of function classes in practice, the log ϵ'-covering number has only logarithmic dependence on ϵ'. Informally, we denote the log covering number as $\log\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}$ and omit its ϵ' dependency for clean presentation. Theorem 5.4.2 claims that the regret scales as $\tilde{\mathcal{O}}(H\sqrt{dK\log\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}})$, where $\tilde{\mathcal{O}}(\cdot)$ omits absolute constants and logarithmic terms.[4]

By the standard online-to-batch argument, we also derive the sample complexity of GOLF.

Corollary 5.4.3 (Sample Complexity of GOLF)**.** *Under Assumption 5.2.1, 5.2.2, there exists an absolute constant c such that for any $\epsilon \in (0,1]$, if we choose $\beta = c\log[\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\epsilon^2/(dH^2))\cdot HK]$ in*

[4] We will not omit $\log\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}$ in $\tilde{\mathcal{O}}(\cdot)$ notation since for many function classes, $\log\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}$ is not small. For instance, for a $\tilde{d}$-dimensional linear function class, $\log\mathcal{N}_{\mathcal{F}\cup\mathcal{G}} = \tilde{\mathcal{O}}(\tilde{d})$.

GOLF, *then the output policy* π^{out} *is* $\mathcal{O}(\epsilon)$*-optimal with probability at least* $1/2$*, if*

$$K \geq \Omega\left(\frac{H^2 d}{\epsilon^2} \cdot \log\left[\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}\left(\frac{\epsilon^2}{H^2 d}\right) \cdot \frac{Hd}{\epsilon}\right]\right),$$

where $d = \min_{\Pi \in \{\mathcal{D}_\Delta, \mathcal{D}_\mathcal{F}\}} \dim_{\mathrm{BE}}\left(\mathcal{F}, \Pi, \epsilon/H\right)$ *is the BE dimension.*

Corollary 5.4.3 claims that $\tilde{\mathcal{O}}(H^2 d \log(\mathcal{N}_{\mathcal{F}\cup\mathcal{G}})/\epsilon^2)$ samples are enough for GOLF to learn a near-optimal policy of any low BE dimension problem. Our sample complexity scales linear in both the BE dimension d, and the log covering number $\log(\mathcal{N}_{\mathcal{F}\cup\mathcal{G}})$.

To showcase the sharpness of our results, we compare them to the previous results when restricted to the corresponding settings. (1) For linear function class with ambient dimension d_{lin}, we have BE dimension $d = \tilde{\mathcal{O}}(d_{\mathrm{lin}})$ and $\log(\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}) = \tilde{\mathcal{O}}(d_{\mathrm{lin}})$. Our regret bound becomes $\tilde{\mathcal{O}}(H d_{\mathrm{lin}}\sqrt{K})$ which matches the best known result (Zanette et al., 2020a) up to logarithmic factors; (2) For function class with low Eluder dimension (Wang et al., 2020b), our results hold under weaker completeness assumptions. Our regret scales with $\sqrt{d_{\mathrm{E}}}$ in terms of dependency on Eluder dimension d_{E}, which improves the linear d_{E} scaling in the regret of Wang et al. (2020b); (3) Finally, for low Bellman rank problems, our sample complexity scales linearly with Bellman rank, which improves upon the quadratic dependence in Jiang et al. (2017). We remark that all results mentioned above assume (approximate) realizability. All except Jiang et al. (2017) assume (approximate) completeness.

5.4.2 Key ideas in proving Theorem 5.4.2

In this subsection, we present a brief proof sketch for the regret bound of GOLF. We defer all the details to Appendix D.5. For simplicity, we only discuss the case of choosing $\mathcal{D}_\mathcal{F}$ as the distribution family Π in the definition of Bellman Eluder dimension (Definition 5.3.3). The proof for using $\mathcal{D}_\Delta$ as the distribution family follows from similar arguments.

Our proof strategy consists of three main steps.

Step 1: Prove optimism. We firstly show that, with high probability, the optimal value function $Q^\star$ indeed lies in the confidence set $\mathcal{B}^k$ for all $k \in [K]$ (Lemma D.5.2 in Appendix D.5.1), which is a natural consequence of martingale concentration and the properties of the confidence set we designed. Because of $Q^\star \in \mathcal{B}^k$, the optimistic planning step (Line 3) in GOLF guarantees that $V_1^\star(s_1) \le \max_a f_1^k(s_1, a)$ for every episode k. This optimism allows the following upper bound on regret

$$\mathrm{Reg}(K) \le \sum_{k=1}^{K} \left(\max_a f_1^k(s_1, a) - V_1^{\pi^k}(s_1) \right) = \sum_{h=1}^{H} \sum_{k=1}^{K} \mathbb{E}_{\pi^k} \left[(f_h^k - \mathcal{T} f_{h+1}^k)(s_h, a_h) \right], \tag{5.4}$$

where the right equality follows from the standard policy loss decomposition (see, e.g., Lemma 1 in Jiang et al. (2017)), and $\mathbb{E}_\pi$ denotes the expectation taken over sequence $(s_1, a_1, \ldots, s_H, a_H)$ when executing policy π.

Step 2: Utilize the sharpness of our confidence set. Recall that our construction of the confidence set in Line 6 of GOLF forces f^k computed in episode k to have a small loss $\mathcal{L}_{\mathcal{D}_h}$, which is a proxy for empirical squared Bellman error under data $\mathcal{D}_h$. Since data $\mathcal{D}_h$ in episode k are collected by executing each π^i for one episode for all $i < k$, by standard martingale concentration arguments and the completeness assumption, we can show that with high probability (Lemma D.5.1 in Appendix D.5.1)

$$\sum_{i=1}^{k-1} \mathbb{E}_{\pi^i} \left[(f_h^k - \mathcal{T} f_{h+1}^k)(s_h, a_h) \right]^2 \le \mathcal{O}(\beta), \text{ for all } (k, h) \in [K] \times [H]. \tag{5.5}$$

Step 3: Establish relations between Eq. (5.4) and Eq. (5.5). So far, we want to upper-bound Eq. (5.4), while we know Eq. (5.5). We note that the RHS of Eq. (5.4) is very similar to the LHS of Eq. (5.5), except that the latter is the squared Bellman error, and the expectation is taken under previous policy π^i for $i < k$. To establish the connection between these two, it turns out that we need the Bellman Eluder dimension to be small. Concretely, we have the following lemma.

Lemma 5.4.4. *Given a function class Φ defined on $\mathcal{X}$ with $|\phi(x)| \le 1$ for all $(\phi, x) \in \Phi \times \mathcal{X}$,*

and a family of probability measures Π *over* $\mathcal{X}$. *Suppose sequence* $\{\phi_k\}_{k=1}^K \subset \Phi$ *and* $\{\mu_k\}_{k=1}^K \subset \Pi$ *satisfy that for all* $k \in [K]$, $\sum_{i=1}^{k-1}(\mathbb{E}_{\mu_i}[\phi_k])^2 \le \beta$. *Then for all* $k \in [K]$, $\sum_{i=1}^{k} |\mathbb{E}_{\mu_i}[\phi_i]| \le \mathcal{O}(\sqrt{\dim_{\mathrm{DE}}(\Phi, \Pi, 1/k)\beta k})$.

Lemma 5.4.4 is a simplification of Lemma D.5.3 in Appendix D.5, which is a modification of Lemma 2 in Russo and Van Roy (2013). Intuitively, Lemma 5.4.4 can be viewed as an analogue of the pigeon-hole principle for DE dimension. Choose Φ to be the function class of Bellman residuals, and μ_k to be the distribution under policy π^k, we finish the proof.

5.5 Conclusion

In this chapter, we propose a new complexity measure—Bellman Eluder (BE) dimension for reinforcement learning with function approximation. Our new complexity measure identifies a new rich class of RL problems that subsumes a majority of existing tractable problem classes in RL. We design a new optimization-based algorithm—GOLF, and provide a new analysis for algorithm OLIVE. Both algorithms show that the new rich class of RL problems we identified in fact can be learned within a polynomial number of samples. We hope our results shed light on the future research in finding the minimal structural assumptions that allow sample-efficient reinforcement learning.

Chapter 6

Provable Reinforcement Learning with a Short-Term Memory

6.1 Introduction

Reinforcement learning is a well-studied paradigm for sequential decision making, in which an agent learns to make decisions in a stateful environment to accumulate reward. The most common framework for reinforcement learning—particularly for theoretical analysis—is the Markov Decision Process (MDP), in which the environment is summarized by a state that is observable to the agent. One notable feature of the MDP is that the agent can be *memoryless*, meaning that it need not remember past states to make decisions in the present. However, many real world problems exhibit partial observability and require the agent to maintain a memory of the past to infer the latent states, plan, and make good decisions. These problems are best modeled via the framework of Partially Observable MDPs (POMDPs).

As a motivating example, consider a control task of navigating a robot that perceives the environment through a visual system like a first-person camera. Here, a single image may identify the agent's

location, but it would not identify the agent's velocity, which is necessary for deciding how much force should be applied in order to accelerate or brake. For optimal control, the agent would have to maintain a memory of past images and infer its velocity from this historical information. This problem can be modeled as a POMDP where the system state is the position and velocity of the agent. However, the state cannot be inferred using a single image, hence it is *partially observable*.

Maintaining a memory and reasoning over histories in POMDPs is notoriously challenging, as evidenced by a number of complexity-theoretic barriers: computing the optimal policy (or planning) is computationally intractable (Papadimitriou and Tsitsiklis, 1987) and learning an unknown POMDP incurs a sample complexity that scales exponentially with the horizon (Mossel and Roch, 2005; Jin et al., 2020a). These lower bounds often involve constructions that require the agent to reason over very long histories. However, they are worst-case in nature, so they leave open the possibility of obtaining positive results for subclasses of POMDPs with special structure of practical interest.

One such structure concerns applications of POMDPs where the agent only needs a *short-term memory*. This structure holds in our motivating example, since the velocity can be recovered from just the most recent images. Short-term memory is also frequently used in the design of practical algorithms, which concatenate observations from the most recent time steps and use them to make decisions—a technique called "frame-stacking" (Mnih et al., 2013, 2015; Hessel et al., 2018). This gives rise to a natural question: *Can we develop a theoretical framework and design provably efficient algorithms for reinforcement learning with a short-term memory?*

Our contributions. In this chapter, we address the question above by proposing a new class of models—m-step decodable POMDPs. This class is a subclass of general POMDPs where the latent state can be determined by the observations and actions of the m most recent time steps via an *unknown* decoding function $\phi^\star$ (see Assumption 6.2.2).

As a warm-up example, we first consider the tabular setting, where the number of states, observations, and actions, are all relatively small. Here a simple technique which stacks the observations and actions in the m most recent steps into a new "mega"-states yields an algorithm with sample

complexity $\mathcal{O}(H(OA)^m)$ where O, A are the number of observations and actions respectively and H is the episode length. We also show an $\Omega(A^m)$ lower bound, establishing that an exponential dependence on m is indeed necessary.

Our main result concerns the rich-observation setting where the observation space can be arbitrarily complex (O is arbitrarily large) and one must use function approximation for generalization. We present a clean solution to this problem with a simple variant of the GOLF algorithm (Jin et al., 2021), which was originally proposed for RL with general function approximation in the observable/Markovian setting. We show that our algorithm finds a near-optimal policy within $\mathcal{O}(\text{poly}(H)A^m S \cdot \log|\mathcal{F}|)$ samples, where S is the number of latent states and $|\mathcal{F}|$ is cardinality of the function class. Most importantly, our sample complexity does not depend on the number of observations O. We further extend our result to the setting where the latent dynamics correspond to a linear MDP, with S in the sample complexity replaced by latent dimension d.

Our results in the rich observation setting crucially rely on a novel concept that we call the "moment matching policy," which breaks historical dependencies while matching the joint distribution of states, observations, and actions for a short time interval (See Section 6.5.2). These policies enable a low-rank or bilinear decomposition of the Bellman error of any value function in the POMDP, which is essential for obtaining sample efficient results in the rich observation setting (Jiang et al., 2017; Jin et al., 2021; Du et al., 2021). As such, the moment matching policies might be of independent interest for future research in partial observability.

6.1.1 Related Work

Partial observability is a central challenge in practical reinforcement learning settings and, as such, it has been the focus of a large body of empirical work. The two most popular high-level approaches are to use recurrent or other "temporally extended" neural architectures (Hausknecht and Stone, 2015; Zhu et al., 2017; Igl et al., 2018; Hafner et al., 2019), or to employ feature engineering (McCallum, 1993), for example by providing the most recent observations as input to the agent (Mnih et al., 2013, 2015; Hessel et al., 2018). However, we are not aware of any theoretical treatment of these

methods in the RL context.

Turning to theoretical results, two lines of work are related to our own. The first addresses RL with partial observability. Kearns et al. (1999, 2002); Even-Dar et al. (2005) provide sparse sampling techniques that attain A^H-type sample complexity for various POMDP tasks, including without resets. These bounds have an undesirable exponential dependence on the horizon, which we show can be removed in some special cases. A more recent line of work (Azizzadenesheli et al., 2016; Guo et al., 2016; Jin et al., 2020a) use method of moment estimators (based on spectral methods for learning latent variable models (c.f., Anandkumar et al., 2014) to obtain guarantees in *undercomplete* tabular POMDPs. However, undercompleteness, which means that the emission matrix is robustly rank $|O|$, need not hold in our setting, so these results are orthogonal to ours.

The second line of work concerns rich observation RL, where the observation space can be infinite and arbitrarily complex, in (for the most part) *Markovian* environments. These works provide structural conditions that permit sample efficient RL with function approximation Jiang et al. (2017); Sun et al. (2019a); Jin et al. (2021); Du et al. (2021); Foster et al. (2021) as well as algorithms that are provably efficient in some special cases (Du et al., 2019; Misra et al., 2020; Agarwal et al., 2020; Uehara et al., 2021). However, as we will see, these structural conditions are not satisfied in our POMDP model so these results do not directly apply.

Outside of RL settings, the use of memory is prevalent in controls and time series prediction (Ljung, 1998; Box et al., 2015; Hamilton, 1994), dating back to the seminal work of Kalman (1960). Short-term memory is explicit in several autoregressive models, such as the AR and ARMA models. It is also classical to leverage memory in many control-theoretic settings. More recently, short-term memory has been employed in control settings, where one can use stability arguments to show that a short memory window suffices to approximate the optimal policy (Verhaegen, 1993; Arora et al., 2018; Agarwal et al., 2019; Oymak and Ozay, 2019; Simchowitz et al., 2019). These ideas provide further motivation for our study but the techniques developed in these continuous settings do not seem useful for discrete RL problems where exploration is challenging.

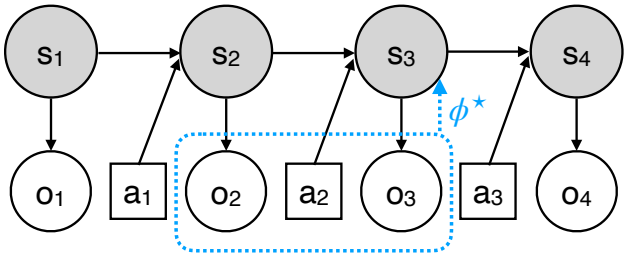


Figure 6.1: A schematic of a 2-step decodable POMDP. The latent state s_h can be recovered using only o_{h-1}, a_{h-1}, o_h, so a short-term memory suffices for decision making.

6.2 Preliminaries

Notation. We use $[H]$ to denote the set $\{1, \ldots, H\}$. For any indexed sequence $a_1, a_2, \ldots$, we use $a_{i:j}$ to denote the subsequence $(a_{\max\{1,i\}}, \ldots, a_{\max\{1,j\}})$ for any $i, j \in \mathbb{Z}$ with $i \le j$. We adopt the standard big-oh notation and write $f = \tilde{\mathcal{O}}(g)$ to denote that $f = \mathcal{O}(g \cdot \max\{1, \mathrm{polylog}(g)\})$.

POMDPs. We consider an episodic Partially Observable Markov Decision Process (POMDP), which can be specified by $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, H, \mathbb{P}, \mathbb{O}, r)$. Here $\mathcal{S}$ is the *unobservable* state space, $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, and H is the horizon. $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ is a collection of *unknown* transition probabilities with $\mathbb{P}_h(s' \mid s, a)$ equal to the probability of transitioning to s' after taking action a in state s at the h^{th} step. $\mathbb{O} = \{\mathbb{O}_h\}_{h=1}^H$ are the *unknown* emissions with $\mathbb{O}_h(o \mid s)$ equal to probability that the environment emits observation o when in state s at the h^{th} step. $r = \{r_h : \mathcal{O} \to [0,1]\}_{h=1}^H$ are the deterministic reward functions.[1] Throughout the chapter, we assume that $\sum_{h=1}^H r_h(o_h) \le 1$ almost surely. We assume our action space is finite, $|\mathcal{A}| \le A$, and in all sections except Section 6.4.1, we assume our state space is also finite, $|\mathcal{S}| \le S$.

Interaction protocol. In a POMDP, the states are hidden and unobservable; i.e., the agent is only able to see the observations and its own actions. Each episode starts with initial state s_1 which is sampled from some *unknown* initial distribution. Then, at each step $h \in [H]$, the environment emits observation $o_h \sim \mathbb{O}_h(\cdot \mid s_h)$, the agent observes $o_h \in \mathcal{O}$, receives reward $r_h(o_h)$, and takes action $a_h \in \mathcal{A}$ causing the environment to transition to $s_{h+1} \sim \mathbb{P}(\cdot \mid s_h, a_h)$.

[1] We study deterministic reward for simplicity. Our results readily generalize to random rewards.

Multi-step decodability. We first define the notion of reachable trajectories.

Definition 6.2.1 (Reachable trajectories)**.** We say a trajectory $\tau = (s_1, o_1, a_1, r_1, s_2, \ldots, s_H, o_H, a_H, r_H)$ is *reachable* if the probability $P((s,o)_{1:H}|a_{1:H}) = (\prod_{h=1}^{H} \mathbb{O}(o_h|s_h)) \cdot (\prod_{h=1}^{H-1} \mathbb{P}(s_{h+1}|s_h, a_h))$ is strictly positive.

Now we present the key structural assumption of this work, which assumes that a suffix of length m of the history suffices to decode the latent state. We use $\mathcal{Z}_h$ to denote the set of suffixes at step h, given by $\mathcal{Z}_h = (\mathcal{O} \times \mathcal{A})^{\min\{h-1, m-1\}} \times \mathcal{O}$.[2] Additionally, since it will appear frequently in subscripts in the sequel, let $m(h) = \min\{h - m + 1, 1\}$.

Assumption 6.2.2 (m-step decodability)**.** There exists an *unknown* decoder $\phi^\star = \{\phi_h^\star : \mathcal{Z}_h \to \mathcal{S}\}_{h=1}^{H}$ such that for every *reachable* trajectory $\tau = (s, o, a)_{1:H}$, we have $s_h = \phi_h^\star(z_h)$ for all $h \in [H]$, where $z_h = ((o,a)_{m(h):h-1}, o_h)$.

We call a POMDP satisfying Assumption 6.2.2 an **m-step decodable POMDP**. An example with $m = 2$ is illustrated in Fig. 6.1. Note that restricting decodability to only hold on reachable sequences results in a weaker assumption, which can include more practical settings.

Our model is a generalization of the block Markov decision process (BMDP) (Jiang et al., 2017; Du et al., 2019), which corresponds to the case where $m = 1$. However, we emphasize that when $m = 1$ there is no partial observability since the current observation suffices for decoding the hidden state. Thus the BMDP model does not require memory while, for $m > 1$, our model does.

Policies and value functions. For m-step decodable POMDPs, we consider the class of m-step policies. An m-step policy π is a collection $\pi = \{\pi_h : \mathcal{Z}_h \to \mathcal{A}\}$ that maps suffixes of length m of the history to actions. The agent follows policy π by choosing action $a_h = \pi_h(z_h)$ at the h^{th} step, where $z_h = ((o,a)_{m(h):h-1}, o_h) \in \mathcal{Z}_h$. We denote V^π as the value for policy π, defined as the expected total reward obtained when following policy π, that is $V^\pi = \mathbb{E}_\pi[\sum_{h=1}^{H} r_h(o_h)]$.

We can similarly define the value at step h to be the expected future reward when starting from step h. While this value may depend on the entire history in general, it is not hard to show that

[2] When $h \le m$, this suffix includes the entire history starting from time step 1.

in m-step decodable POMDPs with an m-step policy π, this value only depends on the suffix of length m. Mathematically, we can define $V_h^\pi : \mathcal{Z}_h \to [0,1]$ to be the value function at step h for (the m-step) policy π as

$$V_h^\pi(z) \coloneqq E_\pi\big[\sum_{h'=h+1}^{H} r_{h'}(o_{h'}) \mid z_h = z\big].$$

Similarly we define $Q_h^\pi : \mathcal{Z}_h \times \mathcal{A} \to [0,1]$ to be the Q-value function at step h for (the m-step) policy π as

$$Q_h^\pi(z,a) \coloneqq E_\pi\big[\sum_{h'=h+1}^{H} r_{h'}(o_{h'}) \mid z_h = z, a_h = a\big].$$

Furthermore, Assumption 6.2.2 guarantees that there exists an m-step policy $\pi^\star$ which is optimal in the sense $V^{\pi^\star} = \max_{\pi \in \Pi} V^\pi$ where Π is the class of all policies, which may depend on the entire history. We use $V^\star$, $V_h^\star$, and $Q_h^\star$ to denote $V^{\pi^\star}$, $V_h^{\pi^\star}$, and $Q_h^{\pi^\star}$ respectively.

We define the *Bellman operator* $\mathcal{T}_h$ at step h as

$$(\mathcal{T}_h g)(z,a) \coloneqq \mathbb{E}\big[r_{h+1}(o_{h+1}) + \max_{a_{h+1} \in \mathcal{A}} g(z_{h+1}, a_{h+1}) \mid z_h = z, a_h = a\big],$$

for any function $g : \mathcal{Z}_{h+1} \times \mathcal{A} \to [0,1]$ that depends on m-step suffix. It is not hard to check that $Q^\star$ satisfies the Bellman optimality equation $Q_h^\star(z,a) = (\mathcal{T}_h Q_{h+1}^\star)(z,a)$ for all $h \in [H]$ and $(z,a) \in \mathcal{Z}_h \times \mathcal{A}$.

Finally, for two non-stationary policies π_1, π_2 we use the notation $\pi_1 \circ_t \pi_2$ be a non-stationary policy that executes π_1 for $t-1$ time steps and then, starting from the t^{th} time step, executes π_2.

Learning objective. Our objective is to learn an ϵ-optimal policy $\widehat{\pi}$, which satisfies $V^{\widehat{\pi}} \geq V^\star - \epsilon$.

6.2.1 Function approximation

In the function approximation setting, the learner is given a function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$, where $\mathcal{F}_h \subseteq (\mathcal{Z}_h \times \mathcal{A} \to [0,1])$ consists of candidate functions to approximate $Q_h^\star$—the optimal

Q-value function at step h. Without loss of generality we assume that $f_{H+1} \equiv 0$. We present two assumptions that are commonly adopted in the literature to avoid challenges associated with reinforcement learning with function approximation (e.g., the hardness results in Krishnamurthy et al. 2016; Weisz et al. 2021).

Assumption 6.2.3 (Realizability)**.** $Q_h^\star \in \mathcal{F}_h$ for all $h \in [H]$.

This assumption requires that our function class $\mathcal{F}$ in fact contains the the optimal Q-value function, $Q^\star$.

Assumption 6.2.4 (Generalized Completeness)**.** $\mathcal{T}_h f_{h+1} \in \mathcal{G}_h$ for all $h \in [H]$ and $f_{h+1} \in \mathcal{F}_{h+1}$, where $\mathcal{G} = \mathcal{G}_1 \times \cdots \times \mathcal{G}_H$ is an auxiliary function class provided to the learner, with $\mathcal{G}_h \subseteq (\mathcal{Z}_h \times \mathcal{A} \to [0, 1])$.

The *generalized completeness* (Antos et al., 2008; Chen and Jiang, 2019) assumption requires the auxiliary function class $\mathcal{G}$ to be rich enough so that applying the Bellman operator on any function in the original class $\mathcal{F}$ results in a function in $\mathcal{G}$. If we choose $\mathcal{G} = \mathcal{F}$, Assumption 6.2.4 reduces to the standard completeness assumption, but separating the two classes provides more flexibility.

We use covering numbers to capture the statistical complexity, or effective size, of the classes $\mathcal{F}$ and $\mathcal{G}$.

Definition 6.2.5 (ϵ-cover)**.** The ϵ-covering of a set $\mathcal{X}$ under a metric ρ, denoted by $\mathcal{N}(\mathcal{X}, \epsilon, \rho)$ is the minimum integer n such that there exists a subset $\mathcal{X}_0 \subseteq \mathcal{X}$ with $|\mathcal{X}_0| = n$ and for any $x \in \mathcal{X}$ there exists $y \in \mathcal{X}_0$ such that $\rho(x, y) \leq \epsilon$.

In this work, for the function class $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$, we use the metric $\rho(f^{(1)} - f^{(2)}) = \max_{h \in H} \|f_h^{(1)} - f_h^{(2)}\|_\infty$ where $f^{(1)}, f^{(2)} \in \mathcal{F}$. Since this metric is fixed throughout the chapter, we use a simpler notation of $\mathcal{N}_{\mathcal{F}}(\epsilon)$ to denote the ϵ-covering number of $\mathcal{F}$.

Finally, let $\pi_f = \{z_h \mapsto \arg\max_{a \in \mathcal{A}} f_h(z_h, a)\}_{h=1}^H$ denote the greedy policy with respect to $f \in \mathcal{F}$, where ties are broken in a canonical fashion.

6.3 Warmup: Tabular Case

We start by considering a basic setting where the numbers of states, actions, and observations are all finite and small, so we additionally have $|\mathcal{O}| \leq O$. In this setting, we describe a simple reduction from an m-step decodable POMDP to a new MDP with augmented states. With this reduction at hand, we can to apply any RL algorithms designed for the fully observable setting to learn a near optimal policy.

In the reduction to an MDP, instead of using only the current observation o_h as the state at time h, we use the m-length suffix of observations and actions z_h. We refer to such a suffix as a *megastate*. Formally, the reduction uses a time-dependent extended state space $\mathcal{S}^{m,h} = \mathcal{Z}_h$, and the next result establishes that $\mathcal{S}^{m,h}$ induces Markovian dynamics. Additionally, an optimal policy of this MDP is also an optimal policy of the original m-step decodable POMDP.[3]

Proposition 6.3.1 (Megastate MDP)**.** *The state space $\mathcal{S}^{m,h}$ induces Markovian dynamics $\mathbb{P}^m$ and reward r^m. Let this MDP be $\mathcal{M}^m = \left(\mathcal{S}^{m,h}, \mathcal{A}, H, \mathbb{P}, r\right)$. An optimal policy of $\mathcal{M}^m$ is an optimal policy of the m-step decodable POMDP.*

We refer to $\mathcal{M}^m$ as the megastate MDP. With this proposition, we can apply any RL algorithm (e.g., UCB-VI by Azar et al. 2017) to the megastate MDP to learn a near optimal policy for the original POMDP. Since the cardinality of the state space of $\mathcal{M}^m$ at each step is $\max_{h\in[H]} \left|\mathcal{S}^{m,h}\right| \leq O^m A^{m-1}$. We immediately obtain the following result.

Corollary 6.3.2 (Upper bound, tabular setting)**.** *For any $\epsilon, \delta \in (0,1)$, UCB-VI applied on to the megastate-MDP $\mathcal{M}^m$ learns an ϵ-optimal policy for the original m-step decodable POMDP with probability greater than $1-\delta$ given $O\left(O^m A^m poly(H) \log\left(1/\delta\right)/\epsilon^2\right)$ samples.*

We remark that the sample complexity scales exponentially with the decoding length m. The next lower-bound verifies the necessity of the $O(A^m)$ term in the upper bound, so some exponential dependence is required. It follows by a reduction to the lower bound of Krishnamurthy et al. (2016); we show that their construction is, in fact, an m-step decodable POMDP. This yields the following

[3] All proofs are deferred to the appendices.

result.

Proposition 6.3.3 (Lower bound, tabular setting)**.** *There exists an m-step decodable MDP that requires at least $\Omega(A^m/\epsilon^2)$ samples to find an ϵ-optimal policy.*

Thus the A^m dependence in the megastate reduction is optimal, although it is not clear whether the O^m dependence is necessary, which we discuss in more detail in Section 6.6. Regardless, the megastate reduction is a reasonable approach for m-step decodable POMDPs when the observation space is small, but, in many applications, the observations represent complex objects (like images or high-dimensional data) so that even linear in O dependence is unsatisfactory. Such problems lie outside the scope of tabular methods, and a fundamentally different approach is required.

6.4 Main results

In this section we present our main results which address the *rich observation* setting, where the number of observation O is extremely large or infinite. The standard approach to tackle such problems is via *value function approximation*: we assume access to a function class $\mathcal{F}$ of candidate Q-value functions. Given such a class, the goal is to learn a near-optimal policy with sample complexity scaling with the statistical complexity of $\mathcal{F}$—in our case the log covering number $\log \mathcal{N}_{\mathcal{F}}$—but independent of the size of the observation space. In this section, we develop an algorithm for rich observation m-step decodable POMDPs and analyze its sample complexity.

Our algorithm, which we call m-GOLF, is displayed in Algorithm 6. It is an adaptation of the GOLF algorithm, developed by Jin et al. (2021), for the rich observation MDP setting. m-GOLF itself differs from GOLF only in one seemingly minor way, although this is quite critical for our analysis. Before turning to this difference, let us review the high-level algorithmic approach.

GOLF, and m-GOLF, are optimistic algorithms that maintain a confidence-set of plausible Q-value functions, and act optimistically with respect to this set. Given a function class $\mathcal{F}$, we first collect a few observations o_1 and estimate the predicted initial value, i.e., $\mathbb{E}\left[f(o_1, \pi_f(o_1))\right]$, for each $f \in \mathcal{F}$. Then, we initialize the confidence set $\mathcal{B}^0 \leftarrow \mathcal{F}$ and empty datasets $\{\mathcal{D}_h\}_{h=1}^H$, one for each time step.

Algorithm 6 m-GOLF: GOLF for m-step decodable POMDP

Initialize: $\mathcal{D}_1, \dots, \mathcal{D}_H \leftarrow \emptyset$, $\mathcal{B}^0 \leftarrow \mathcal{F}$.
Estimate value of initial state by collecting K_{est} episodes and only keeping their first observations, denoted by $\hat{o}_1^1, \dots, \hat{o}_1^{K_{\mathrm{est}}}$. For $f \in \mathcal{F}$, define

$$\hat{f}_1 = (1/K_{\mathrm{est}}) \sum_{i=1}^{K_{\mathrm{est}}} f(\hat{o}_1^i, \pi_f(\hat{o}_1^i))$$

for epoch k from 1 to K **do**
  Choose policy $\pi^k = \pi_{f^k}$, where $f^k = \mathrm{argmax}_{f \in \mathcal{B}^{k-1}} \hat{f}_1$.
  for step h from 1 to H **do**
    Collect $z_h = (o_{h-m+1}, a_{h-m+1}, \dots, o_h)$, a_h, r_h, and o_{h+1} by executing π^k at step $1, \dots, h-m$ and taking action uniformly at random at step $h-m+1, \dots, h$.
    Augment $\mathcal{D}_h = \mathcal{D}_h \cup (z_h, a_h, r_h, o_{h+1})$ for all $h \in [H]$.
  Update

$$\mathcal{B}^k = \left\{ f \in \mathcal{F} : \ \mathcal{L}_{\mathcal{D}_h}(f_h, f_{h+1}) \le \inf_{g \in \mathcal{G}_h} \mathcal{L}_{\mathcal{D}_h}(g, f_{h+1}) + \beta \text{ for all } h \in [H] \right\},$$

$$\text{where } \mathcal{L}_{\mathcal{D}_h}(\xi_h, \zeta_{h+1}) = \sum_{(z_h, a_h, r_h, o_{h+1}) \in \mathcal{D}_h} [\xi_h(z_h, a_h) - r_h - \max_{a' \in \mathcal{A}} \zeta_{h+1}(z_{h+1}, a')]^2.$$

Output π^{out} uniform mixture policy over $\{\pi^k\}_{k=1}^K$.

Then for each epoch $k \in [K]$ we follow three steps:

1. *Optimistic planning.* Compute the function $f \in \mathcal{B}^{k-1}$ with largest predicted initial value.
2. *Data collection.* Collect one trajectory by following $\pi_{f_k} \circ_{m(h)} \mathrm{Uniform}(\mathcal{A})$ for each $h \in [H]$. That is we collect h trajectories total, rolling in with the greedy policy π_{f_k} until time $h-m$ and rolling out randomly.
3. *Refine the confidence set.* Update the confidence set to $\mathcal{B}^k$ using the newly collected trajectories. The confidence set is designed so that $Q^\star \in \mathcal{B}^k$ for all $k \in [K]$ and that all functions in $\mathcal{B}^k$ have low squared Bellman error on the data collected in the previous episodes.

After iterating through these steps for several epochs, m-GOLF outputs uniform mixture over all previous policies $\{\pi^k\}_{k=1}^K$.

The main difference between GOLF and m-GOLF is in the data collection procedure. Instead of

collecting H trajectories per epoch, GOLF collets a single trajectory where all actions are taken by the greedy policy π_{f_k}. On the other hand, in m-GOLF, we interrupt the greedy policy and execute random actions so that the tuple z_h that is added to $\mathcal{D}^h$ is collected from $\pi_{f_k} \circ_{m(h)}$ Uniform. At face value, this modification is relatively benign, but we will see how interrupting the greedy policy is critical to establishing sample complexity guarantees in the m-step decodable POMDP.

We analyze m-GOLF in two settings. The first is where the underlying/latent MDP is tabular, meaning that S and A are small. The second setting is where the latent MDP has a linear or low rank structure. Our first theorem provides a sample complexity guarantee for m-GOLF when the latent dynamics are tabular.

Theorem 6.4.1. *Under Assumptions 6.2.2, 6.2.3, and 6.2.4, there exists an absolute constant c such that for any $\delta \in (0,1]$ and $\epsilon > 0$, if we choose*

$$
\begin{aligned}
K_{\mathrm{est}} &= c \cdot \Big(\log[\mathcal{N}_{\mathcal{F}}(\epsilon)/\delta]/\epsilon^2 \Big) \\
\beta &= c \cdot \Big(\log\left[\mathcal{N}_{\mathcal{G}}(\rho) K H/\delta\right] + K\rho \Big) \\
\rho &= \epsilon^2 \cdot \left[H^2 A^m S \log[S/\epsilon]\right]^{-1}
\end{aligned}
$$

in m-GOLF (Algorithm 6), then the output policy π^{out} is $\mathcal{O}(\epsilon)$-optimal with probability at least $1-\delta$ if

$$
K \geq \tilde{\Omega}\left(\frac{H^2 A^m S}{\epsilon^2} \cdot \log\left[\frac{\mathcal{N}_{\mathcal{G}}(\rho)}{\delta} \right] \right).
$$

Theorem 6.4.1 establishes a sample complexity bound for m-GOLF scaling as $\mathrm{poly}(S, A^m, H, \mathrm{comp}(\mathcal{F}, \mathcal{G}), 1/\epsilon)$ where $\mathrm{comp}(\cdot)$ is our measure of statistical complexity. Unlike the megastate reduction, there is no explicit dependence on the size of the observation space O; instead the bound scales with the complexity of the function class, which allows us to exploit domain knowledge and inductive biases when deploying the algorithm. We emphasize that these previous results (Jiang et al., 2017; Jin et al., 2021; Du et al., 2021) do not yield guarantees when $m \geq 2$, as we will see in Section 6.5.

6.4.1 Linear m-step Decodable POMDP

In this subsection, we show that m-GOLF extends to the setting where the number of state S is also large. Specifically, we consider the case where the latent MDP is a linear MDP Jin et al. (2020c)—there exists known feature map $\psi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_{lin}}$ such that the transition dynamics are linear in ψ. Interestingly, we show that m-GOLF is still applicable without change. It retains a similar sample complexity guarantee where we replace the dependence on S with a dependence on the latent dimensionality d_{lin}.

Formally, a linear MDP is defined as follows:

Definition 6.4.2 (Linear MDP)**.** An MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ is said to be a linear with a feature map $\psi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_{\text{lin}}}$, if for any $h \in [H]$: There exists d_{lin} unknown (signed) measures $\boldsymbol{\mu}_h = \{\mu_h^{(1)}, \ldots, \mu_h^{(d_{\text{lin}})}\}$ over $\mathcal{S}$ such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have

$$\mathbb{P}_h(\cdot \mid s, a) = \langle \boldsymbol{\mu}_h(\cdot), \psi(s, a) \rangle$$

We assume the standard normalization: $\|\psi(s, a)\| \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\int v(s)\boldsymbol{\mu}_h(s)\|_2 \leq \sqrt{d_{\text{lin}}}$ for all $h \in [H]$ and v with $\|v\|_\infty \leq 1$.

The following result gives a sample complexity guarantee for m-GOLF in the more general linear m-step decodable POMDP model.

Theorem 6.4.3. *Under Assumptions 6.2.2, 6.2.3, and 6.2.4 and assuming linear latent MDP; there exists an absolute constant c such that for any $\delta \in (0, 1]$ and $\epsilon > 0$, if we choose*

$$K_{\text{est}} = c \cdot \left(\log[\mathcal{N}_{\mathcal{F}}(\epsilon)/\delta]/\epsilon^2 \right)$$
$$\beta = c \cdot \left(\log \left[\mathcal{N}_{\mathcal{G}}(\rho) K H/\delta\right] + K\rho \right)$$
$$\rho = \epsilon^2 \cdot \left[H^2 A^m d_{\text{lin}} \log[d_{\text{lin}}/\epsilon]\right]^{-1}$$

in m-GOLF (Algorithm 6), then the output policy π^{out} is $\mathcal{O}(\epsilon)$-optimal with probability at least $1 - \delta$

if

$$K \geq \tilde{\Omega}\left(\frac{H^2 A^m d_{\text{lin}}}{\epsilon^2} \cdot \log\left[\frac{\mathcal{N}_{\mathcal{G}}(\rho)}{\delta}\right]\right).$$

Theorem 6.4.3 is almost the same as Theorem 6.4.1 with the dependency on the number of latent state S replaced by the ambient dimensionality d_{lin}. As a result, Theorem 6.4.3 can apply to the case where the number of state S is extremely large or even infinite, as long as the underlying MDP has a linear structure.

6.5 Challenges and Proof Overview

In this section we elaborate on the main challenges in analysis, explain our main technique and provide a proof overview for Theorem 6.4.1. For clarity, we will focus on the special case of 2-step decodable POMDP in this setting. We refer reader to Appendix E.2 for cases where $m > 2$.

6.5.1 Challenges: Bellman Rank is Prohibitively Large

We first note that existing postive results for RL algorithms with general function approximation such as OLIVE Jiang et al. (2017), GOLF Jin et al. (2021) all rely on the structural properties that certain complexity measure on the Bellman error is small. One such complexity is the Bellman rank Jiang et al. (2017), which explains the tractability of block MDP (the special case of m-step decodable POMDP with $m = 1$).

Consider the Bellman error at the h^{th} time step of a function $f \in \mathcal{F}$ when executing roll-in policy π, given by

$$\mathcal{E}_h(\pi, f) = \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h)) \mid a_{1:h-1} \sim \pi].$$

Bellman rank is defined as the smallest integer M such that the Bellman error can be factorized as inner product in M dimensional linear space. That is, there exists $\zeta, \xi \in \mathbb{R}^M$ such that $\mathcal{E}_h(\pi, f) = \langle \zeta(\pi), \xi(f) \rangle$.

Intuitively, Bellman rank describes how much information is shared among past (roll-in policy π)

and future (value function f) at step h. In the special case of 1-step decodable POMDP, it suffices to consider 1-step policy where the choice of action a_h only depends on the current observation o_h. In this case, given the state s_h at the current step h, the past—$(s, o, a)_{1:h-1}$ (which only depends on roll-in policy π) is completely independent of the future—$(o_h, a_h, (s, o, a)_{h+1:H})$ (which only depends on function f). Therefore, it can be shown the Bellman rank of 1-step decodable POMDP (i.e. block MDP) is upper bounded by the number of states S Jiang et al. (2017).

However, such independent structure completely collapses in 2-step decodable POMDP, where we must consider 2-step policy. Due to the nature of such policies, the choice of action a_h not only depends on the current observation o_h, but also the observation and action in the previous step o_{h-1}, a_{h-1} (as shown in Figure 6.2 blue box). Therefore, conditioning s_h, the past is no longer independent of the future. This can potentially lead to very large Bellman rank.

Formally, our next result shows that the Bellman rank in 2-step decodable POMDP can be prohibitively large—there exists examples where the Bellman rank can be lower bounded by the cardinality of the observation space $\Omega(O)$. This is highly undesirable in the rich observation setting where O can be even infinite. Furthermore, we also show that OLIVE algorithm—which was proposed in Jiang et al. (2017) to solve all RL problems with small Bellman rank—needs at least $\Omega(O)$ samples to find an $O(1)$ optimal policy.

Proposition 6.5.1 (Bellman rank of m-step decodable POMDP is large)**.** *There exists a 2-step decodable POMDP $\mathcal{M}$ and a function class $\mathcal{F}$ such that the Bellman rank of $(\mathcal{M}, \mathcal{F})$ is $\Omega(O)$. Additionally,* OLIVE *instantiated with $\mathcal{F}$ requires $\Omega(O)$ samples to find an $o(1)$ optimal policy.*

This highlights the challenge on directly applying existing results or techniques to solve m-step decodable POMDPs. Although OLIVE solves a 1-step decodable POMDP—namely, a block MDP—it fails in solving an m-step decodable POMDP for $m \geq 2$.

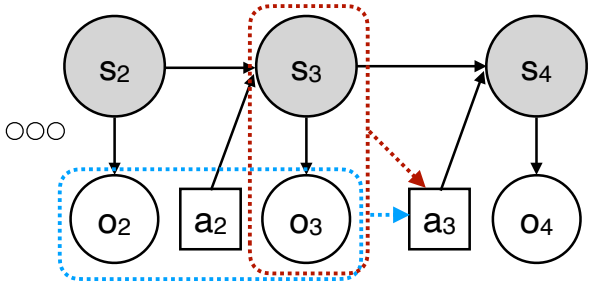


Figure 6.2: An illustration of the dependency structure of a moment matching policy, depicted in red, and a regular policy, depicted in blue, in a 2-step decodable POMDP. The moment matching policy $\mu^{\pi,h+1}$ selects action a_h based on the state s_h and observation o_h to match the distribution $\mathbb{P}^{\pi}[a_h \mid s_h, o_h]$. It breaks the dependence on the history by marginalizing out (o_{h-1}, a_{h-1}), but correctly matches the distribution $\mathbb{P}^{\pi}[o_{h+1}, a_h, o_h]$.

6.5.2 Proof Overview & Moment Matching Policy

Our main proof idea revolves around breaking the complicated dependencies introduced by multiple-step policies, which requires a number of crucial observations.

Our first key observation is that, in order to establish the sample complexity for GOLF algorithm, we don't necessarily need to prove the low rank structure of the Bellman error. We only need to alternatively identify an auxiliary function $\mathcal{E}_h^\star(\pi, f)$ which satisfies the following two properties (see formal statement in Lemma E.2.9):

1. *Matches with standard bellman error when $\pi = \pi_f$*:

$$\mathcal{E}_h^\star(\pi_f, f) = \mathcal{E}_h(\pi_f, f).$$

2. *Has a low-rank decomposition*:

$$\mathcal{E}_h^\star(\pi, f) = \langle \zeta(\pi), \xi(f) \rangle .$$

for some $\zeta(\cdot), \xi(\cdot) \in \mathbb{R}^M$ with small M,

This discovery gives us a lot extra freedom in designing the functional form of the $\mathcal{E}_h^\star$. In particular,

for 2-step decodable POMDP, we define $\mathcal{E}_h^\star$ to be the normal Bellman error but with the policy at step $h-1$ changed from roll-in policy π to a new policy μ_f which depends only on f instead of π.

$$\mathcal{E}_h^\star(\pi, f) \equiv \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h)) \mid a_{1:h-1} \sim \pi \circ_{h-1} \mu_f].$$

The second key observation is that we can choose μ_f in a form which breaks the dependency and allows low-rank dependency. Concretely, instead of choosing μ_f to be standard 2-step policy where a_{h-1} will then depend on $(o_{h-2}, a_{h-2}, o_{h-1})$, we choose μ_f to be the policy that only depends on (s_{h-1}, o_{h-1}) (See Figure 6.2 red box). The benefit of considering such policy is that now conditioned on s_{h-1} at step $h-1$, the past—$(s,o,a)_{1:h-2}$ (which only depends on roll-in policy π) is now independent of the future—$(o_{h-1}, a_{h-1}, (s,o,a)_{h:H})$ (which only depends on function f). This immediately leads to a low-rank decomposition of $\mathcal{E}_h^\star(\pi, f)$ with rank $M = S$.

Our third key observation is that we can carefully choose the value of μ_f within the form specified above, so that $\mathcal{E}_h^\star(\pi, f)$ matches the Bellman error $\mathcal{E}_h(\pi, f)$ when roll-in policy is the greedy policy of f, i.e. $\pi = \pi_f$. This is done by the idea of "*moment-matching*", which is the reason we call policy μ_f the "moment matching policy". Specifically, we choose policy μ_f such that

$$\mu_f(a_{h-1}|(o,s)_{h-1}) = \mathbb{E}_{\pi_f}[\pi_f(a_{h-1}|z_{h-1})|(o,s)_{h-1}]$$

which is policy of π_f averaging over all trajectories with $(o,s)_{h-1}$ fixed. The most important property of this policy is that the joint distributions over z_h for policy π_f and policy $\pi_f \circ_{h-1} \mu_f$ (which switches at time step $h-1$) are the same. In symbol:

$$P_{\pi_f}(z_h) = P_{\pi_f \circ_{h-1} \mu_f}(z_h)$$

This directly leads to the matching in the Bellman error. This finishes our construction of $\mathcal{E}_h^\star(\pi, f)$ satisfying the two properties mentioned earlier and the main part of proof overview.

Finally, we comment that our construction of μ_f depends on the latent state s which can not be

observed in POMDP. Nevertheless, m-GOLF bypasses this problem by executing a uniform action for m time steps, instead of executing μ_f; taking the uniform action for the last m time steps allows us to upper bound $\mathcal{E}_h^\star(\pi, f)$ using the importance sampling trick, while only suffering an A^m degradation in the sample complexity. Such factor is necessary according to Proposition 6.3.3.

6.6 Conclusion

In this chapter, we initiate the study of m-step decodable POMDPs as a model for understanding the role of short-term memory in sequential decision making. We consider both the tabular and function approximation setting and obtain results that scale exponential with the memory window rather than the horizon, which could be much larger. In the function approximation case, our techniques rely crucially on the moment matching policy to break dependency on the history, and we hope this concept may be useful in other settings with partial observability.

We believe our progress on understanding short-term memory is just scratching the surface and there are many questions that remain open even in the m-step decodable POMDP model. The most basic question pertains to the tabular setting, where the upper bound in Corollary 6.3.2 and the lower bound in Proposition 6.3.3 differ by an O^m factor. Instantiating m-GOLF in the tabular setting also incurs an O^m factor. On the other hand, the next result shows that by using a carefully constructed policy class in an importance sampling approach, we can avoid the O^m factor in exchange for an A^H factor, which could be more favorable in some settings. See Appendix E.3 for details and the proof.

Proposition 6.6.1. *There exists an algorithm such that for any $m \leq H$ and any m-step decodable POMDP, the algorithm returns an ϵ-optimal policy with probability greater than $1-\delta$ given $poly(A^H, O, S, H, \log(1/\delta))/\epsilon^2$ samples.*

Based on this result, we conjecture that the O^m factor can be avoided and that $A^m \mathrm{poly}(H, S, O, A)$ is the optimal sample complexity for m-step decodable POMDPs. However, this question remains open.

The second question concerns whether we can avoid completeness, as defined in Assumption 6.2.4,

in the rich observation setting. Intuition from prior works suggests that if we could replace the squared bellman error constraint with one on the average Bellman errors, then an algorithm and analysis similar to OLIVE would successfully do this. However, when working with average Bellman errors, introducing the moment matching policy requires explicitly importance weighting with them, meaning that we must use these policies in the algorithm and not just the analysis. Unfortunately since we do not know the moment matching policies (or a small class containing them), this approach seems to fail.

We believe that characterizing the optimal sample complexity (in the tabular setting) or removing the completeness assumption (in the rich observation setting) will require new techniques and be a mark of significant progress toward expanding our understanding of decision making with short-term memory. We look forward to studying these questions in future work.

Bibliography

Abernethy, J., Bartlett, P. L., and Hazan, E. (2011). Blackwell approachability and no-regret learning are equivalent. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 27–46.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. (2017). Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org.

Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. E. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. *arXiv preprint arXiv:1402.0555*.

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. (2020). Flambe: Structural complexity and representation learning of low rank mdps. *Advances in Neural Information Processing Systems*, 33.

Agarwal, N., Bullins, B., Hazan, E., Kakade, S., and Singh, K. (2019). Online control with adversarial disturbances. In *International Conference on Machine Learning*, pages 111–119. PMLR.

Agrawal, S. and Devanur, N. R. (2014). Bandits with concave rewards and convex knapsacks. In *Proceedings of the 15th ACM Conference on Economics and Computatxion (EC)*.

Agrawal, S. and Jia, R. (2017). Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pages 1184–1194.

Altman, E. (1999). *Constrained Markov decision processes*, volume 7. CRC Press.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832.

Anderson, B. D. and Moore, J. B. (2007). *Optimal control: linear quadratic methods*. Courier Corporation.

Antos, A., Szepesvári, C., and Munos, R. (2008). Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71(1):89–129.

Arora, S., Hazan, E., Lee, H., Singh, K., Zhang, C., and Zhang, Y. (2018). Towards provable control for unknown linear dynamical systems. In *International Conference on Learning Representations, Workshop Track*.

Azar, M. G., Osband, I., and Munos, R. (2017). Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR.

Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. (2016). Reinforcement learning of pomdps using spectral methods. In *Conference on Learning Theory*, pages 193–256. PMLR.

Babaioff, M., Dughmi, S., Kleinberg, R. D., and Slivkins, A. (2015). Dynamic pricing with limited supply. *TEAC*, 3(1):4. Special issue for *13th ACM EC*, 2012.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2018). Bandits with knapsacks. *Journal of the ACM*, 65(3):13:1–13:55. Preliminary version in *FOCS 2013*.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

Bellman, R. (1957). A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684.

Besbes, O. and Zeevi, A. (2009). Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms. *Operations Research*, 57(6):1407–1420.

Besbes, O. and Zeevi, A. (2011). On the minimax complexity of pricing in a changing environment. *Operations Reseach*, 59(1):66–79.

Blackwell, D. (1956). An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Boyle, J. P. and Dykstra, R. L. (1986). A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in order restricted statistical inference*, pages 28–47. Springer.

Brafman, R. I. and Tennenholtz, M. (2002). R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231.

Brantley, K., Dudik, M., Lykouris, T., Miryoosefi, S., Simchowitz, M., Slivkins, A., and Sun, W. (2020). Constrained episodic reinforcement learning in concave-convex and knapsack settings. In *Advances in Neural Information Processing Systems*, volume 33, pages 16315–16326. Curran Associates, Inc.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. (2019). Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830*.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Chen, J. and Jiang, N. (2019). Information-theoretic considerations in batch reinforcement learning. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1042–1051. PMLR.

Cheung, W. C. (2019). Regret minimization for reinforcement learning with vectorial feedback and complex objectives. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dann, C. and Brunskill, E. (2015). Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2818–2826.

Dann, C., Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. (2018). On oracle-efficient pac rl with rich observations. In *Advances in neural information processing systems*, pages 1422–1432.

Dann, C., Lattimore, T., and Brunskill, E. (2017). Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5713–5723.

Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S. (2019). On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, pages 1–47.

Ding, D., Wei, X., Yang, Z., Wang, Z., and Jovanovic, M. (2021). Provably efficient safe exploration via primal-dual policy optimization. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3304–3312. PMLR.

Domingues, O. D., Ménard, P., Kaufmann, E., and Valko, M. (2021). Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pages 578–598. PMLR.

Dong, K., Peng, J., Wang, Y., and Zhou, Y. (2020). Root-n-regret for learning in markov decision processes with function approximation and low bellman rank. In *Conference on Learning Theory*, pages 1554–1557. PMLR.

Du, S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudik, M., and Langford, J. (2019). Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR.

Du, S. S., Kakade, S. M., Lee, J. D., Lovett, S., Mahajan, G., Sun, W., and Wang, R. (2021). Bilinear classes: A structural framework for provable generalization in rl. *arXiv preprint arXiv:2103.10897*.

Efroni, Y., Jin, C., Krishnamurthy, A., and Miryoosefi, S. (2022). Provable reinforcement learning with a short-term memory. *arXiv preprint arXiv:2202.03983*.

Efroni, Y., Mannor, S., and Pirotta, M. (2020). Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189*.

Even-Dar, E., Kakade, S. M., and Mansour, Y. (2005). Reinforcement learning in pomdps without resets. In *International Joint Conference on Artificial Intelligence*.

Fazel, M., Ge, R., Kakade, S., and Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476. PMLR.

Filar, J. and Vrieze, K. (2012). *Competitive Markov decision processes*. Springer Science & Business Media.

Foster, D. J., Kakade, S. M., Qian, J., and Rakhlin, A. (2021). The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*.

Foster, D. J., Rakhlin, A., Simchi-Levi, D., and Xu, Y. (2020). Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *arXiv preprint arXiv:2010.03104*.

Freund, Y. and Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103.

Guo, Z. D., Doroudi, S., and Brunskill, E. (2016). A pac rl algorithm for episodic pomdps. In *Artificial Intelligence and Statistics*, pages 510–518. PMLR.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. (2019). Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.

Hamilton, J. D. (1994). *Time series analysis*. Princeton university press.

Hausknecht, M. and Stone, P. (2015). Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*.

Hazan, E., Kakade, S. M., Singh, K., and Van Soest, A. (2018). Provably efficient maximum entropy exploration. *arXiv preprint arXiv:1812.02690*.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. (2018). Deep variational reinforcement learning for pomdps. In *International Conference on Machine Learning*, pages 2117–2126. PMLR.

Ingram, J. M. and Marsh, M. (1991). Projections onto convex cones in hilbert space. *Journal of approximation theory*, 64(3):343–350.

Jabbari, S., Joseph, M., Kearns, M., Morgenstern, J., and Roth, A. (2017). Fairness in reinforcement learning. In *International Conference on Machine Learning*, pages 1617–1626. PMLR.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. (2017). 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.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. (2018). Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pages 4863–4873.

Jin, C., Kakade, S. M., Krishnamurthy, A., and Liu, Q. (2020a). Sample-efficient reinforcement learning of undercomplete pomdps. *arXiv:2006.12484*.

Jin, C., Krishnamurthy, A., Simchowitz, M., and Yu, T. (2020b). Reward-free exploration for reinforcement learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR.

Jin, C., Liu, Q., and Miryoosefi, S. (2021). Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*.

Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2019). A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020c). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*.

Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2):193–208.

Kearns, M. and Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2):209–232.

Kearns, M. J., Mansour, Y., and Ng, A. Y. (1999). Approximate planning in large pomdps via reusable trajectories. In *NIPS*, pages 1001–1007. Citeseer.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Krishnamurthy, A., Agarwal, A., and Langford, J. (2016). Pac reinforcement learning with rich observations. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Le, H., Voloshin, C., and Yue, Y. (2019). Batch policy learning under constraints. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*.

Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., and Legg, S. (2017). Ai safety gridworlds. *arXiv preprint arXiv:1711.09883*.

Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., and Jurafsky, D. (2016). Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318.

Liu, Q., Yu, T., Bai, Y., and Jin, C. (2020). A sharp analysis of model-based reinforcement learning with self-play. *arXiv preprint arXiv:2010.01604*.

Ljung, L. (1998). *System Identification: Theory for the User*. Pearson Education.

Mao, H., Alizadeh, M., Menache, I., and Kandula, S. (2016). Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, page 50–56, New York, NY, USA. Association for Computing Machinery.

McCallum, R. A. (1993). Overcoming incomplete perception with utile distinction memory. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 190–196.

Miryoosefi, S., Brantley, K., Daume III, H., Dudik, M., and Schapire, R. E. (2019). Reinforcement learning with convex constraints. In *Advances in Neural Information Processing Systems*, volume 32, pages 14093–14102. Curran Associates, Inc.

Miryoosefi, S. and Jin, C. (2021). A simple reward-free approach to constrained reinforcement learning. *arXiv preprint arXiv:2107.05216*.

Misra, D., Henaff, M., Krishnamurthy, A., and Langford, J. (2020). Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *International conference on machine learning*, pages 6961–6971. PMLR.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529.

Mossel, E. and Roch, S. (2005). Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 366–375.

Munos, R. and Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857.

Neu, G. and Pike-Burke, C. (2020). A unifying view of optimism in episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 33.

Neumann, J. v. (1928). Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320.

Osband, I. and Van Roy, B. (2014). Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pages 1466–1474.

Oymak, S. and Ozay, N. (2019). Non-asymptotic identification of lti systems from a single trajectory. In *2019 American control conference (ACC)*, pages 5655–5661. IEEE.

Papadimitriou, C. H. and Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qiu, S., Wei, X., Yang, Z., Ye, J., and Wang, Z. (2020). Upper confidence primal-dual optimization: Stochastically constrained markov decision processes with adversarial losses and unknown transitions. *arXiv preprint arXiv:2003.00660*.

Rakhlin, A., Sridharan, K., and Tewari, A. (2010). Online learning: Random averages, combinatorial parameters, and learnability.

Ray, A., Achiam, J., and Amodei, D. (2020). Benchmarking safe exploration in deep reinforcement learning. https://cdn.openai.com/safexp-short.pdf. Accessed March 11, 2020.

Rockafellar, R. T. (2015). *Convex analysis*. Princeton university press.

Rosenberg, A. and Mansour, Y. (2019). Online convex optimization in adversarial markov decision processes. In *International Conference on Machine Learning*, pages 5478–5486.

Russo, D. and Van Roy, B. (2013). Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pages 2256–2264.

Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. (2015). Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Simchowitz, M., Boczar, R., and Recht, B. (2019). Learning linear dynamical systems with semi-parametric least squares. In *Conference on Learning Theory*, pages 2714–2802. PMLR.

Singh, R., Gupta, A., and Shroff, N. B. (2020). Learning in markov decision processes under constraints. *arXiv preprint arXiv:2002.12435*.

Singh, S., James, M., and Rudary, M. (2012). Predictive state representations: A new theory for modeling dynamical systems. *arXiv preprint arXiv:1207.4167*.

Sion, M. (1958). On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286. Also available at `https://arxiv.org/abs/1904.07272`.

Song, Z. and Sun, W. (2019). Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A., and Langford, J. (2019a). Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pages 2898–2933.

Sun, W., Vemula, A., Boots, B., and Bagnell, J. A. (2019b). Provably efficient imitation learning from observation alone. *arXiv preprint arXiv:1905.10948*.

Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4):160–163.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, first edition.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, second edition.

Syed, U. and Schapire, R. E. (2007). A game-theoretic approach to apprenticeship learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1449–1456, Red Hook, NY, USA. Curran Associates Inc.

Szepesvári, C. (2010). Algorithms for reinforcement learning. *Synthesis lectures on artificial intelligence and machine learning*, 4(1):1–103.

Szepesvári, C. and Munos, R. (2005). Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887.

Tarbouriech, J. and Lazaric, A. (2019). Active exploration in markov decision processes. *arXiv preprint arXiv:1902.11199*.

Tessler, C., Mankowitz, D. J., and Mannor, S. (2019). Reward constrained policy optimization. In *International Conference on Learning Representations*.

Uehara, M., Zhang, X., and Sun, W. (2021). Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Verhaegen, M. (1993). Subspace model identification part 3. analysis of the ordinary output-error state-space model identification algorithm. *International Journal of control*, 58(3):555–586.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

von Neumann, J. (1928). Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100:295–320.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Wang, R., Du, S. S., Yang, L. F., and Salakhutdinov, R. (2020a). On reward-free reinforcement learning with linear function approximation. *arXiv preprint arXiv:2006.11274*.

Wang, R., Salakhutdinov, R., and Yang, L. F. (2020b). Provably efficient reinforcement learning with general value function approximation. *arXiv preprint arXiv:2005.10804*.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. (2019). Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*.

Wang, Z., Deng, S., and Ye, Y. (2014). Close the gaps: A learning-while-doing algorithm for single-product revenue management problems. *Operations Research*, 62(2):318–331.

Weisz, G., Amortila, P., and Szepesvári, C. (2021). Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In *Algorithmic Learning Theory*, pages 1237–1264. PMLR.

Wu, J., Braverman, V., and Yang, L. F. (2020). Accommodating picky customers: Regret bound and exploration complexity for multi-objective reinforcement learning. *arXiv preprint arXiv:2011.13034*.

Xie, T. and Jiang, N. (2020). Batch value-function approximation with only realizability. *arXiv preprint arXiv:2008.04990*.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. I. (2020). Bridging exploration and general function approximation in reinforcement learning: Provably efficient kernel and neural value iterations. *arXiv preprint arXiv:2011.04622*.

Yu, T., Tian, Y., Zhang, J., and Sra, S. (2021). Provably efficient algorithms for multi-objective competitive rl. *arXiv preprint arXiv:2102.03192*.

Zanette, A. and Brunskill, E. (2019). Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. *arXiv preprint arXiv:1901.00210*.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. (2020a). Learning near optimal policies with low inherent bellman error. *arXiv preprint arXiv:2003.00153*.

Zanette, A., Lazaric, A., Kochenderfer, M. J., and Brunskill, E. (2020b). Provably efficient reward-agnostic navigation with linear value iteration. *arXiv preprint arXiv:2008.07737*.

Zhang, X., Singla, A., et al. (2020a). Task-agnostic exploration in reinforcement learning. *arXiv preprint arXiv:2006.09497*.

Zhang, Z., Zhou, Y., and Ji, X. (2020b). Almost optimal model-free reinforcement learning via reference-advantage decomposition. *arXiv preprint arXiv:2004.10019*.

Zheng, L. and Ratliff, L. J. (2020). Constrained upper confidence reinforcement learning. *arXiv preprint arXiv:2001.09377*.

Zhu, P., Li, X., Poupart, P., and Miao, G. (2017). On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1704.07978*.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Part III

Appendix

Appendix A

Remaining Proofs of Chapter 2

A.1 Online gradient descent (OGD)

Algorithm 7 Online gradient descent (OGD)

input: projection oracle Γ_Λ $\{\Gamma_\Lambda(\boldsymbol{\lambda}) = \operatorname{argmin}_{\boldsymbol{\lambda}' \in \Lambda} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|\}$
init: $\boldsymbol{\lambda}_1$ arbitrarily
parameters: step size η_t
for $t = 1$ **to** T **do**
    observe convex loss function $\ell_t : \Lambda \to \mathbb{R}$
    $\boldsymbol{\lambda}'_{t+1} = \boldsymbol{\lambda}_t - \eta_t \nabla \ell_t(\boldsymbol{\lambda}_t)$
    $\boldsymbol{\lambda}_{t+1} = \Gamma_\Lambda(\boldsymbol{\lambda}'_{\mathbf{t+1}})$

Theorem A.1.1. *(Zinkevich, 2003) Assume that for any $\boldsymbol{\lambda}, \boldsymbol{\lambda}' \in \Lambda$ we have $\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\| \leq D$ and also $\|\nabla \ell_t(\boldsymbol{\lambda})\| \leq G$. Let $\eta_t = \eta = \frac{D}{G\sqrt{T}}$. Then the regret of OGD is*

$$\text{Regret}_T(OGD) = \sum_{t=1}^{T} \ell_t(\boldsymbol{\lambda}_t) - \min_{\boldsymbol{\lambda}} \sum_{t=1}^{T} \ell_t(\boldsymbol{\lambda}) \leq DG\sqrt{T}.$$

A.2 Proof of Theorem 2.3.1

We have that

$$\begin{aligned}
\frac{1}{T}\sum_{t=1}^{T} g(\boldsymbol{\lambda}_t, \mathbf{u}_t) &= \frac{1}{T}\sum_{t=1}^{T} \min_{\mathbf{u}\in\mathcal{U}} g(\boldsymbol{\lambda}_t, \mathbf{u}) && \text{(A.1)}\\
&\leq \frac{1}{T}\min_{\mathbf{u}\in\mathcal{U}} \sum_{t=1}^{T} g(\boldsymbol{\lambda}_t, \mathbf{u}) && \text{(A.2)}\\
&\leq \min_{\mathbf{u}\in\mathcal{U}} g\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\lambda}_t, \mathbf{u}\right) && \text{(A.3)}\\
&\leq \max_{\boldsymbol{\lambda}\in\Lambda}\min_{\mathbf{u}\in\mathcal{U}} g(\boldsymbol{\lambda}, \mathbf{u}). && \text{(A.4)}
\end{aligned}$$

Eq. (A.1) is because the $\mathbf{u}$-player is playing best response so that $\mathbf{u}_t = \text{argmin}_{\mathbf{u}\in\mathcal{U}} g(\boldsymbol{\lambda}_t, \mathbf{u})$. Eq. (A.2) is because taking the minimum of each term of a sum cannot exceed the minimum of the sum as a whole. Eqs. (A.3) and (A.4) use the concavity of g with respect to $\boldsymbol{\lambda}$, and the definition of max, respectively. By letting $\delta = \frac{1}{T}\text{Regret}_T$, writing the definition of regret for the $\boldsymbol{\lambda}$-player, and using $\ell_t(\boldsymbol{\lambda}) = -g(\boldsymbol{\lambda}, \mathbf{u}_t)$, we have

$$\frac{1}{T}\sum_{t=1}^{T} g(\boldsymbol{\lambda}_t, \mathbf{u}_t) + \delta = \frac{1}{T}\max_{\boldsymbol{\lambda}\in\Lambda}\sum_{t=1}^{T} g(\boldsymbol{\lambda}, \mathbf{u}_t) \geq \max_{\boldsymbol{\lambda}\in\Lambda} g\left(\boldsymbol{\lambda}, \frac{1}{T}\sum_{t=1}^{T}\mathbf{u}_t\right) \geq \min_{\mathbf{u}\in\mathcal{U}}\max_{\boldsymbol{\lambda}\in\Lambda} g(\boldsymbol{\lambda}, \mathbf{u}),$$

where the second and third inequalities use convexity of g with respect to $\mathbf{u}$ and definition of min, respectively. Combining yields

$$\min_{\mathbf{u}\in\mathcal{U}} g\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\lambda}_t, \mathbf{u}\right) \geq \min_{\mathbf{u}\in\mathcal{U}}\max_{\boldsymbol{\lambda}\in\Lambda} g(\boldsymbol{\lambda}, \mathbf{u}) - \delta,$$

and also

$$\max_{\boldsymbol{\lambda}\in\Lambda} g\left(\boldsymbol{\lambda}, \frac{1}{T}\sum_{t=1}^{T}\mathbf{u}_t\right) \leq \max_{\boldsymbol{\lambda}\in\Lambda}\min_{\mathbf{u}\in\mathcal{U}} g(\boldsymbol{\lambda}, \mathbf{u}) + \delta,$$

completing the proof.

A.3 Proof of Theorem 2.3.3

Let v be the value of the game in Eq. (2.7):

$$v = \min_{\mu \in \Delta(\Pi)} \operatorname{dist}(\overline{\mathbf{z}}(\mu), \mathcal{C}), \tag{A.5}$$

and let $\ell_t(\boldsymbol{\lambda}) = -\boldsymbol{\lambda} \cdot \hat{\mathbf{z}}_t$ (i.e., the loss function that OGD observes).

Lemma A.3.1. *For $t = 1, 2, \ldots, T$ we have*

$$\ell_t(\boldsymbol{\lambda}_t) = -\boldsymbol{\lambda}_t \cdot \hat{\mathbf{z}}_t \geq -v - (\epsilon_0 + \epsilon_1).$$

Proof. By Eq. (2.5) (which must hold by Lemma 2.3.2), and by Eq. (A.5), there exists $\mu^* \in \Delta(\Pi)$ such that

$$v = \operatorname{dist}(\overline{\mathbf{z}}(\mu^*), \mathcal{C}) = \max_{\boldsymbol{\lambda} \in \Lambda} \boldsymbol{\lambda} \cdot \overline{\mathbf{z}}(\mu^*).$$

Thus, $\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\mu^*) \leq v$ since $\boldsymbol{\lambda}_t \in \Lambda$ for all t. By our assumed guarantee for the policy π_t returned by the planning oracle, we have

$$-\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\pi_t) \geq -\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\mu^*) - \epsilon_0 \geq -v - \epsilon_0.$$

Now using the error bound of the estimation oracle,

$$\|\overline{\mathbf{z}}(\pi_t) - \hat{\mathbf{z}}_t\| \leq \epsilon_1, \tag{A.6}$$

and the fact that $\|\boldsymbol{\lambda}_t\| \leq 1$, we have

$$(-\boldsymbol{\lambda}_t \cdot \hat{\mathbf{z}}_t) + \epsilon_1 \geq -\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\pi_t).$$

Combining completes the proof. □

Now we are ready to prove Theorem 2.3.3. Using the definition of mixed policy $\bar{\mu}$ returned by the

algorithm we have

$$\begin{aligned}
\operatorname{dist}(\overline{\mathbf{z}}(\bar{\mu}), \mathcal{C}) &= \operatorname{dist}\left(\frac{1}{T}\sum_{t=1}^{T}\overline{\mathbf{z}}(\pi_t), \mathcal{C}\right) \\
&= \max_{\boldsymbol{\lambda}\in\Lambda} \boldsymbol{\lambda}\cdot\left(\frac{1}{T}\sum_{t=1}^{T}\overline{\mathbf{z}}(\pi_t)\right) && \text{(A.7)} \\
&= \frac{1}{T}\max_{\boldsymbol{\lambda}\in\Lambda}\sum_{t=1}^{T}\boldsymbol{\lambda}\cdot\overline{\mathbf{z}}(\pi_t) \\
&\le \frac{1}{T}\max_{\boldsymbol{\lambda}\in\Lambda}\sum_{t=1}^{T}\boldsymbol{\lambda}\cdot\hat{\mathbf{z}}_t + \epsilon_1 && \text{(A.8)} \\
&= -\frac{1}{T}\min_{\boldsymbol{\lambda}\in\Lambda}\sum_{t=1}^{T}\ell_t(\boldsymbol{\lambda}) + \epsilon_1 && \text{(A.9)} \\
&\le -\frac{1}{T}\min_{\boldsymbol{\lambda}\in\Lambda}\sum_{t=1}^{T}\ell_t(\boldsymbol{\lambda}) + \epsilon_1 + \frac{1}{T}\sum_{t=1}^{T}(\ell_t(\boldsymbol{\lambda}_t) + \epsilon_1 + \epsilon_0 + v) && \text{(A.10)} \\
&= v + \left(-\frac{1}{T}\min_{\boldsymbol{\lambda}\in\Lambda}\sum_{t=1}^{T}\ell_t(\boldsymbol{\lambda}) + \frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{\lambda}_t)\right) + 2\epsilon_1 + \epsilon_0 \\
&= v + \frac{\operatorname{Regret}_T(\mathrm{OGD})}{T} + 2\epsilon_1 + \epsilon_0.
\end{aligned}$$

Here, Eq. (A.7) is by Eq. (2.5). Eq. (A.8) uses Eq. (A.6) and the fact that $\|\boldsymbol{\lambda}\| \le 1$. Eq. (A.11) uses Lemma A.3.1.

The diameter of decision set $\Lambda = \mathcal{C}^\circ \cap \mathcal{B}$ is at most 1. The gradient of the loss function $\nabla(\ell_t(\boldsymbol{\lambda})) = -\hat{\mathbf{z}}_t$ has norm at most $\|\overline{\mathbf{z}}(\pi_t)\| + \epsilon_1 \le \frac{B}{1-\gamma} + \epsilon_1$. Therefore, setting $\eta = \left((\frac{B}{1-\gamma} + \epsilon_1)\sqrt{T}\right)^{-1}$ based on Theorem A.1.1, we get

$$\frac{\operatorname{Regret}_T(\mathrm{OGD})}{T} \le \left(\frac{B}{1-\gamma} + \epsilon_1\right)T^{-1/2}$$

A.4 ApproPO for feasibility

Algorithm 8 APPROPO – Feasibility

input projection oracle $\Gamma_{\mathcal{C}}(\cdot)$ for target set $\mathcal{C}$ which is a convex cone,
positive response oracle PosPlan$(\cdot)$, estimation oracle Est$(\cdot)$,
step size η, number of iterations T
define $\Lambda \triangleq \mathcal{C}^\circ \cap \mathcal{B}$, and its projection operator $\Gamma_\Lambda(\mathbf{x}) \triangleq (\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x}))/\max\{1, \|\mathbf{x} - \Gamma_{\mathcal{C}}(\mathbf{x})\|\}$
initialize $\boldsymbol{\lambda}_1$ arbitrarily in Λ
for $t = 1$ **to** T **do**
  Call positive response oracle for the standard RL with scalar reward $r = -\boldsymbol{\lambda}_t \cdot \mathbf{z}$:
    $\pi_t \leftarrow \text{PosPlan}(\boldsymbol{\lambda}_t)$
  Call the estimation oracle to approximate long-term measurement for π_t:
    $\hat{\mathbf{z}}_t \leftarrow \text{Est}(\pi_t)$
  Update using online gradient descent with the loss function $\ell_t(\boldsymbol{\lambda}) = -\boldsymbol{\lambda} \cdot \hat{\mathbf{z}}_t$:
    $\boldsymbol{\lambda}_{t+1} \leftarrow \Gamma_\Lambda\left(\boldsymbol{\lambda}_t + \eta \hat{\mathbf{z}}_t\right)$
  if $\ell_t(\boldsymbol{\lambda}_t) < -(\epsilon_0 + \epsilon_1)$ **then**
    return *problem is infeasible*
return $\bar{\mu}$, a uniform mixture over $\pi_1, \ldots, \pi_T$

A.4.1 Proof of Theorem 2.3.4

Lemma A.4.1. *If the problem is feasible, then for $t = 1, 2, \ldots, T$ we have*

$$\ell_t(\boldsymbol{\lambda}_t) = -\boldsymbol{\lambda}_t \cdot \hat{\mathbf{z}}_t \geq -(\epsilon_0 + \epsilon_1).$$

Proof. If the problem is feasible, then there exists μ^* such that $\overline{\mathbf{z}}(\mu^*) \in \mathcal{C}$. Since all $\boldsymbol{\lambda}_t \in \mathcal{C}^\circ$, they all have non-positive inner product with every point in $\mathcal{C}$ including $\overline{\mathbf{z}}(\mu^*)$. Since $-\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\mu^*) \geq 0$, we can conclude that $\max_{\pi \in \Pi} R(\pi) = \max_{\pi \in \Pi} -\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\pi) \geq 0$. Therefore, by our guarantee for the positive response oracle,

$$R(\pi_t) = -\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\pi) \geq -\epsilon_0.$$

Now using Eq. (A.6) and the fact that $\|\boldsymbol{\lambda}_t\| \leq 1$, we have

$$(-\boldsymbol{\lambda}_t \cdot \hat{\mathbf{z}}_t) + \epsilon_1 \geq -\boldsymbol{\lambda}_t \cdot \overline{\mathbf{z}}(\pi_t).$$

Combining completes the proof. □The proof of Theorem 2.3.4 is similar to that of Theorem 2.3.3. If the algorithm reports infeasibility then the problem is infeasible as a result of

Lemma A.4.1. Otherwise, we have

$$\frac{1}{T}\sum_{t=1}^{T}(\ell_t(\boldsymbol{\lambda}_t)+\epsilon_1+\epsilon_0)\geq 0,$$

which can be combined with Eq. (A.9) as before. Continuing this argument as before yields

$$\operatorname{dist}(\overline{\mathbf{z}}(\mu),\mathcal{C})\leq\left(\frac{B}{1-\gamma}+\epsilon_1\right)T^{-1/2}+2\epsilon_1+\epsilon_0,$$

completing the proof.

A.5 Proof of Lemma 2.3.5

Let $\mathcal{C}'=\mathcal{C}\times\{\kappa\}$ and $\mathbf{q}$ be the projection of $\tilde{\mathbf{x}}=\mathbf{x}\oplus\langle\kappa\rangle$ onto $\tilde{\mathcal{C}}=\operatorname{cone}(\mathcal{C}')$, i.e.,

$$\mathbf{q}=\arg\min_{\mathbf{y}\in\tilde{\mathcal{C}}}\|\tilde{\mathbf{x}}-\mathbf{y}\|.$$

Let r be the last coordinate of $\mathbf{q}$. We prove the lemma in cases based on the value of r (which cannot be negative by construction).

Case 1 $(r>\kappa)$**:** Since $\mathbf{q}\in\operatorname{cone}(\mathcal{C}')$ with $r>0$, there exists $\alpha>0$ and $\mathbf{q}'\in\mathcal{C}'$ so that $\mathbf{q}=\alpha\mathbf{q}'$. See Figure A.1(a). Consider the plane defined by the three points $\tilde{\mathbf{x}},\mathbf{q},\mathbf{q}'$. Since the origin $\mathbf{0}$ is on the line passing through $\mathbf{q}$ and $\mathbf{q}'$, it must also be in this plane. Now consider the line that passes through $\tilde{\mathbf{x}}$ and $\mathbf{q}'$. Note that all points on this line have last coordinate equal to κ, and they are all also in the aforementioned plane. Let $\mathbf{v}\oplus\langle\kappa\rangle$ be the projection of $\mathbf{0}$ onto this line ($\mathbf{v}\in\mathbb{R}^d$).

Note that the two triangles $\Delta(\tilde{\mathbf{x}},\mathbf{q},\mathbf{q}')$ and $\Delta(\mathbf{0},\mathbf{v}\oplus\langle\kappa\rangle,\mathbf{q}')$ are similar since they are right triangles with opposite angles at $\mathbf{q}'$. Therefore, by triangle similarity,

$$\frac{\|\mathbf{q}'\|}{\|\mathbf{v}\oplus\langle\kappa\rangle\|}=\frac{\|\tilde{\mathbf{x}}-\mathbf{q}'\|}{\|\tilde{\mathbf{x}}-\mathbf{q}\|}\geq\frac{\operatorname{dist}(\tilde{\mathbf{x}},\mathcal{C}')}{\operatorname{dist}(\tilde{\mathbf{x}},\tilde{\mathcal{C}})}=\frac{\operatorname{dist}(\mathbf{x},\mathcal{C})}{\operatorname{dist}(\tilde{\mathbf{x}},\tilde{\mathcal{C}})}.$$

Since $\mathbf{q}' \in \mathcal{C}'$, we have $\|\mathbf{q}'\| \leq \sqrt{(\max_{\mathbf{x}\in\mathcal{C}} \|\mathbf{x}\|)^2 + \kappa^2}$, resulting in

$$\frac{\|\mathbf{q}'\|}{\|\mathbf{v} \oplus \langle\kappa\rangle\|} \leq \frac{\sqrt{(\max_{\mathbf{x}\in\mathcal{C}} \|\mathbf{x}\|)^2 + \kappa^2}}{\kappa} = \sqrt{1+2\delta} \leq 1 + \delta$$

by the choice of κ given in the lemma. Combining completes the proof for this case.

Case 2 $(r = \kappa)$**:** Since $\mathbf{q} \in \text{cone}(\mathcal{C}')$ with κ as last coordinate, we have $\mathbf{q} \in \mathcal{C}'$. Thus,

$$\text{dist}(\mathbf{x}, \mathcal{C}) = \text{dist}(\tilde{\mathbf{x}}, \mathcal{C}') \leq \|\tilde{\mathbf{x}} - \mathbf{q}\| = \text{dist}(\tilde{\mathbf{x}}, \tilde{\mathcal{C}})$$

which completes the proof for this case.

Case 3 $(0 < r < \kappa)$**:** The proof for this case is formally identical to that of Case 1, except that, in this case, the two triangles $\Delta(\tilde{\mathbf{x}}, \mathbf{q}, \mathbf{q}')$ and $\Delta(\mathbf{0}, \mathbf{v} \oplus \langle\kappa\rangle, \mathbf{q}')$ are now similar as a result of being right triangles with a shared angle at $\mathbf{q}'$. See Figure A.1(b).

Case 4 $(r = 0)$**:** Since $\mathbf{q} \in \text{cone}(\mathcal{C}')$, $\mathbf{q}$ must have been generated by multiplying some $\alpha \geq 0$ by some point in $\mathcal{C}'$. Since all points in $\mathcal{C}'$ have last coordinate equal to $\kappa > 0$, and since $r = 0$, it must be the case that $\alpha = 0$, and thus, $\mathbf{q} = \mathbf{0}$. Let $\mathbf{q}'$ be the projection of $\tilde{\mathbf{x}}$ onto $\mathcal{C}'$. See Figure A.1(c). Consider the plane defined by the three points $\tilde{\mathbf{x}}, \mathbf{q} = \mathbf{0}, \mathbf{q}'$. Let $\mathbf{q}''$ be the projection of $\tilde{\mathbf{x}}$ onto the line passing through $\mathbf{q}$ and $\mathbf{q}'$. Then

$$\|\tilde{\mathbf{x}} - \mathbf{q}''\| \leq \|\tilde{\mathbf{x}}\| = \text{dist}(\tilde{\mathbf{x}}, \tilde{\mathcal{C}}).$$

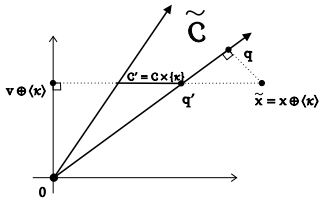
Now consider the line passing through $\tilde{\mathbf{x}}$ and $\mathbf{q}'$. Note that all points on this line have last coordinate equal to κ and are also in the aforementioned plane. Let $\mathbf{v} \oplus \langle\kappa\rangle$ be the projection of $\mathbf{0}$ onto this line ($\mathbf{v} \in \mathbb{R}^d$). Note that the two triangles $\Delta(\tilde{\mathbf{x}}, \mathbf{q}'', \mathbf{q}')$ and $\Delta(\mathbf{0}, \mathbf{v} \oplus \langle\kappa\rangle, \mathbf{q}')$ are similar since they are right triangles with a shared angle at $\mathbf{q}'$. Therefore, by triangle similarity,

$$\frac{\|\mathbf{q}'\|}{\|\mathbf{v} \oplus \langle\kappa\rangle\|} = \frac{\|\tilde{\mathbf{x}} - \mathbf{q}'\|}{\|\tilde{\mathbf{x}} - \mathbf{q}''\|} \geq \frac{\text{dist}(\tilde{\mathbf{x}}, \mathcal{C}')}{\text{dist}(\tilde{\mathbf{x}}, \tilde{\mathcal{C}})} = \frac{\text{dist}(\mathbf{x}, \mathcal{C})}{\text{dist}(\tilde{\mathbf{x}}, \tilde{\mathcal{C}})}.$$

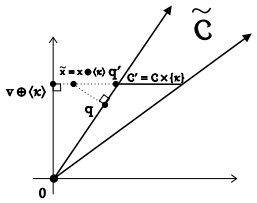
The rest of the proof for this case is exactly as in Case 1.

A.6 Additional experimental details

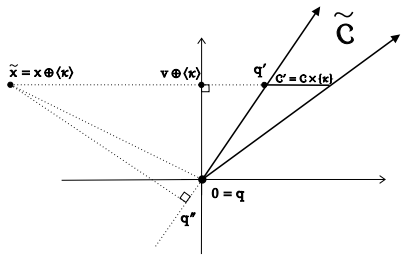
All the models were trained using the following hyperparameters: policy network consists of 2-layer fully-connected MLP with ReLU activation and 128 hidden units and a A2C learning rate of 10^{-2}. For APPROPO, the constant κ (subsection 2.3.3) is set to be 20. In the following figures, the performance of the algorithms has been depicted using different hyperparamters; showing average and standard deviation over 25 runs,.



(a) $r > \kappa$

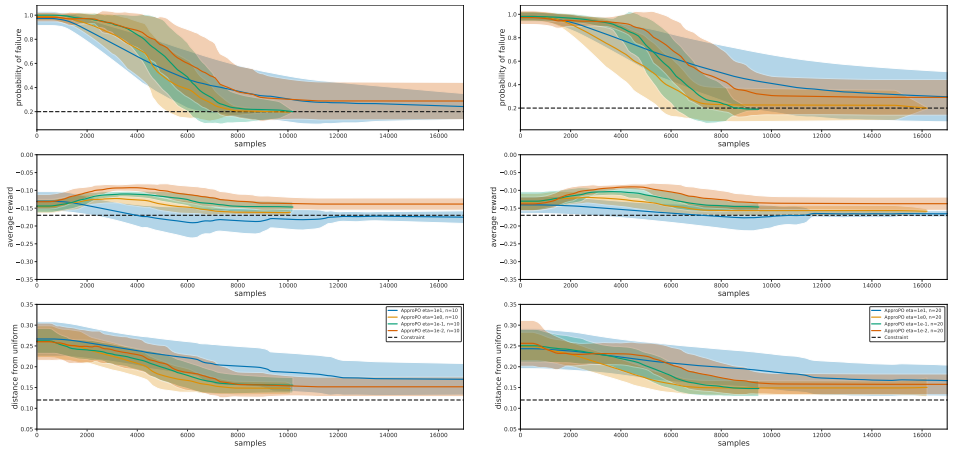


(b) $0 < r < \kappa$



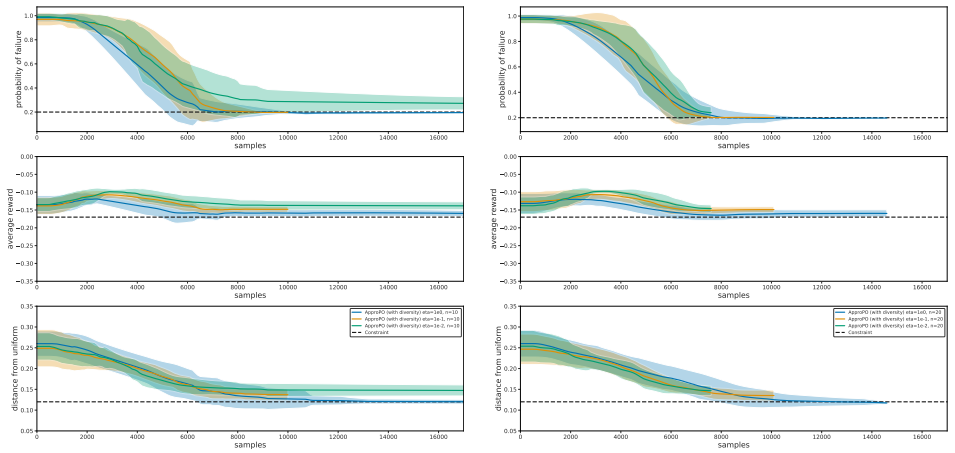
(c) $r = 0$

Figure A.1: Geometric Interpretation of the proof of Lemma 2.3.5



(a) $n = 10$ (b) $n = 20$

Figure A.2: Performance of APPROPO using different hyperparameters. The two numbers are learning rate for the online learning algorithm and n (subsection 2.3.4) respectively. In all figures, the x-axis is number samples. The vertical axes correspond to the three constraints, with thresholds shown as a dashed line; for reward (middle) this is a lower bound; for the others it is an upper bound.



(a) $n = 10$ (b) $n = 20$

Figure A.3: Performance of APPROPO with diversity constraints using different hyperparameters. The two numbers are learning rate for the online learning algorithm and n (subsection 2.3.4) respectively. In all figures, the x-axis is number samples. The vertical axes correspond to the three constraints, with thresholds shown as a dashed line; for reward (middle) this is a lower bound; for the others it is an upper bound.

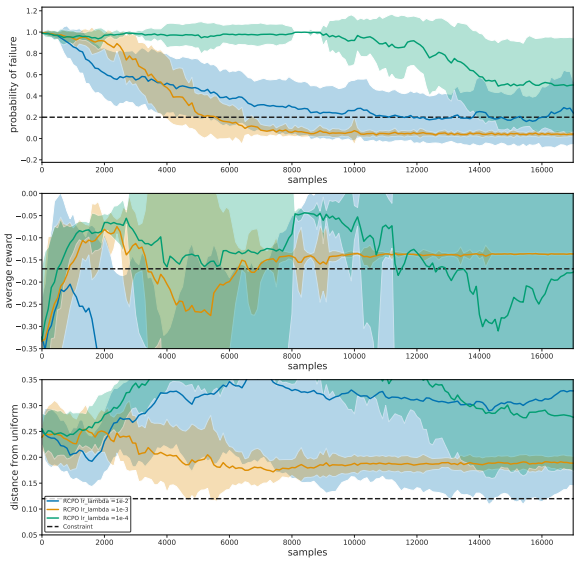


Figure A.4: Performance of RCPO using different learning rates for Lagrange multiplier. In all figures, the x-axis is number samples. The vertical axes correspond to the three constraints, with thresholds shown as a dashed line; for reward (middle) this is a lower bound; for the others it is an upper bound.

Appendix B

Remaining Proofs of Chapter 3

Structure of the supplementary material.

The supplementary material consists of six sections:

- Appendix B.1 provides the formal description of the algorithm and the instantiations of ConPlanner as well as how it can be expressed as a (linear/convex) mathematical program.
- Appendix B.2 provides the proofs for the results of the basic setting presented in Section 3.3.
- Appendix B.3 provides the proofs and additional discussion for the results of the concave-convex setting presented in Section 3.4.
- Appendix B.4 provides the proofs and additional discussion for the results of knapsack setting presented in Section 3.5.
- Appendix B.5 provides further details regarding the experiments presented in Section 3.6.
- Appendix B.6 provides auxiliary concentration lemmas useful for the derivation of our results.

.

B.1 Algorithm: Formal description and design choices

Our main algorithm, denoted by ConRL, is presented at Algorithm 9. We instantiate ConRL for our different settings (i.e. basic setting, concave-convex, and knapsack) by using the appropriate ConPlanner that we discuss in the remainder of this section.

Algorithm 9 ConRL

1: **for** Episode k from 1 to K **do**

2: **Compute empirical estimates:**

$$\text{Compute } N_k,\ \widehat{p}_k,\ \widehat{r}_k, \text{ and } \widehat{\mathbf{c}}_k \text{ based on equations (3.3)}$$

3: **Compute bonus:**

$$\text{Compute } \widehat{b}_k \text{ as equation (3.5)}$$

4: **Call constrained planner**:

$$\pi_k \leftarrow \textsc{ConPlanner}(\widehat{p}_k, \widehat{r}_k, \widehat{\mathbf{c}}_k, \widehat{b}_k)$$

5: **Execute policy**: initial state $s_{k,1} = s_0$

6: **for** Stage h from 1 to H **do**

7: Select $a_{k,h} \sim \pi_k\Big(s_{k,h}\Big)$

8: Observe reward $r_{k,h}$, consumptions $\forall i \in \mathcal{D} : c_{k,h,i}$, and new state $s_{k,h+1}$

B.1.1 Basic setting - BasicConPlanner

We define the bonus-enhanced cMDP, i.e. $\mathcal{M}^{(k)} = \big(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)}\big)$, as

$$p^{(k)}(s'|s,a) = \widehat{p}_k(s'|s,a) \quad \forall s, a, s'$$

$$r^{(k)}(s,a) = \widehat{r}_k(s,a) + \widehat{b}_k(s,a) \quad \forall s, a$$

$$c^{(k)}(s,a,i) = \widehat{c}_k(s,a,i) - \widehat{b}_k(s,a) \quad \forall s, a, i \in \mathcal{D}$$

then we solve the following optimization problem

$$\max_{\pi} \mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} r^{(k)}\big(s_h,a_h\big)\Big] \qquad \text{s.t.} \qquad \forall i\in\mathcal{D}: \mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} c^{(k)}\big(s_h,a_h,i\big)\Big] \le \xi(i).$$

This optimization problem can be solved exactly since it is equivalent to the following linear program on occupation measures (Rosenberg and Mansour, 2019; **?**). Decision variables are $\rho(s,a,h)$, i.e. probability of agent being at state action pair (s,a) at time step h.

$$\begin{aligned}
\max_{\rho} \sum_{s,a,h} \rho(s,a,h) r^{(k)}(s,a) \quad \text{s.t.} \quad & \sum_{s,a,h} \rho(s,a,h) c^{(k)}(s,a,i) \le \xi(i) \quad \forall i\in\mathcal{D} \\
& \forall s', h \quad \sum_{a} \rho(s',a,h+1) = \sum_{s,a} \rho(s,a,h) p^{(k)}(s'|s,a) \\
& \forall s,a,h \quad 0 \le \rho(s,a,h) \le 1 \qquad \sum_{s,a} \rho(s,a,h) = 1
\end{aligned} \tag{B.1}$$

B.1.2 Concave-convex setting - ConvexConPlanner

In this setting, unlike basic setting, objective and constraints are not linear. Therefore, due to lack of monotonicity, we cannot explicitly define the bonus-enhanced CMDP $\mathcal{M}^{(k)} = \big(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)}\big)$. The bonus-enhanced CMDP is implicit in the following program that we solve (see section 3.4)

$$\max_{\pi} \max_{r^{(k)}\in\left[\widehat{r}_k \pm \widehat{b}_k\right]} f\Big(\mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} r^{(k)}\big(s_h,a_h\big)\Big]\Big) \text{ s.t. } \min_{\mathbf{c}^{(k)}\in\left[\widehat{\mathbf{c}}_k \pm \widehat{b}_k\cdot\mathbf{1}\right]} g\Big(\mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} \mathbf{c}^{(k)}\big(s_h,a_h\big)\Big]\Big) \le 0.$$

Similar to before, expressing this program based on occupation measures provides a convex program.

$$\begin{aligned}
\max_{\rho} \max_{r\in\left[\widehat{r}_k \pm \widehat{b}_k\right]} f\Big(\sum_{s,a,h} \rho(s,a,h) r(s,a)\Big) \quad \text{s.t.} \quad & \min_{\mathbf{c}\in\left[\widehat{\mathbf{c}}_k \pm \widehat{b}_k\cdot\mathbf{1}\right]} g\Big(\sum_{s,a,h} \rho(s,a,h) \mathbf{c}(s,a)\Big) \le 0 \\
& \forall s', h: \quad \sum_{a} \rho(s',a,h+1) = \sum_{s,a} \rho(s,a,h) \widehat{p}_k(s'|s,a) \\
& \forall s,a,h: \quad 0 \le \rho(s,a,h) \le 1 \quad \text{and} \quad \sum_{s,a} \rho(s,a,h) = 1
\end{aligned} \tag{B.2}$$

The notations $r \in \left[\widehat{r}_k \pm \widehat{b}_k\right]$ and $\mathbf{c} \in \left[\widehat{\mathbf{c}}_k \pm \widehat{b}_k \cdot \mathbf{1}\right]$ are defined as

$$r \in \left[\widehat{r}_k \pm \widehat{b}_k\right] \iff \forall s, a: \quad r(s,a) \in [\widehat{r}_k(s,a) - \widehat{b}_k(s,a), \widehat{r}_k(s,a) + \widehat{b}_k(s,a)]$$

$$\mathbf{c} \in \left[\widehat{\mathbf{c}}_k \pm \widehat{b}_k \cdot \mathbf{1}\right] \iff \forall i \in \mathcal{D}, s, a: \quad c(s,a,i) \in [\widehat{c}_k(s,a,i) - \widehat{b}_k(s,a), \widehat{c}_k(s,a,i) + \widehat{b}_k(s,a)]$$

Note that if f and g are linear, we end up with a linear program similar to (B.1)

B.1.3 Knapsack setting - KnapsackConPlanner

We define the bonus-enhanced cMDP, i.e. $\mathcal{M}^{(k)} = \left(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)}\right)$ similar to basic setting (B.1.1). We also solve a similar optimization problem with tighter constraints:

$$\max_{\pi} \mathbb{E}^{\pi, p^{(k)}}\Big[\sum_{h=1}^{H} r^{(k)}\big(s_h, a_h\big)\Big] \quad \text{s.t.} \quad \forall i \in \mathcal{D}: \mathbb{E}^{\pi, p^{(k)}}\Big[\sum_{h=1}^{H} c^{(k)}\big(s_h, a_h, i\big)\Big] \leq \frac{(1-\epsilon)B_i}{K}.$$

This optimization problem can again be solved using the following linear program on occupation measures. Decision variables are $\rho(s,a,h)$, i.e. probability of agent being at state action pair (s,a) at step h.

$$\begin{aligned} \max_{\rho} \sum_{s,a,h} \rho(s,a,h) r^{(k)}(s,a) \quad \text{s.t.} \quad & \sum_{s,a,h} \rho(s,a,h) c^{(k)}(s,a,i) \leq \frac{(1-\epsilon)B_i}{K} \quad \forall i \in \mathcal{D} \\ & \forall s', h \quad \sum_{a} \rho(s', a, h+1) = \sum_{s,a} \rho(s,a,h) p^{(k)}(s'|s,a) \\ & \forall s, a, h \quad 0 \leq \rho(s,a,h) \leq 1 \qquad \sum_{s,a} \rho(s,a,h) = 1 \end{aligned} \tag{B.3}$$

B.2 Analysis: Basic setting (Section 3.3)

In this section, we prove the main guarantee for the basic setting.

B.2.1 Validity of bonus (Lemma 3.3.2)

We first prove that $\widehat{b}_k(s,a) = \min\left\{2H, H\sqrt{\frac{2\ln\left(8SAH(d+1)k^2/\delta\right)}{N_k(s,a)}}\right\}$ of Eq. Eq. (3.5) is valid as in the Definition 3.3.1.

Proof of Lemma 3.3.2. We focus on a single state-action pair s, a, stage h, and objective m. Since the support of m is in $[0,1]$ and the one of the value is in $[0, H-1]$, by Hoeffding inequality (see Lemma B.6.2), it holds that, for all k, since (s,a)-pair is visited $N_k(s,a)$ times prior to episode k, with probability at least $1-\delta'$:

$$\left|\left(\widehat{m}_k(s,a) - m^\star(s,a)\right) + \sum_{s'\in\mathcal{S}}\left(\widehat{p}_k(s'|s,a) - p^\star(s'|s,a)\right)V\right| \leq H\sqrt{\frac{2\ln(2/\delta')}{N_k(s,a)}}.$$

Also note that $\widehat{m}_k(s,a) \in [0,1], m^\star(s,a) \in [0,1]$, and $\|V\|_\infty \leq H$, the LHS of the above inequality must be less than $1 + H \leq 2H$.

As a result, the bonus $\widehat{b}_k(s,a,\delta)$ satisfies this inequality for a particular state-action-step-objective with failure probability at most $\delta' = \frac{\delta}{4SAH(d+1)k^2}$ and is therefore valid (satisfying it for all states-actions-steps-objectives) with failure probability $\frac{\delta}{4k^2}$. Union bounding across episodes, the probability of $\widehat{b}_k(s,a,\delta)$ not being valid for some k is at most $\sum_{k=1}^{K}\frac{\delta}{4k^2} \leq \delta$. □

B.2.2 Valid bonus implies optimism

The main reason to optimize a bonus-enhanced model with valid bonuses is because the latter render the model *optimistic*, i.e., its estimated reward is an overestimate of the true reward. Similarly, in constrained settings, its estimated resource consumptions are underestimates of the true resource consumptions. This is formalized in the following definition.

Definition B.2.1. A CMDP $\mathcal{M} = (p, r, \mathbf{c})$ is *optimistic* if its estimated reward (resp. consumption) value function for policy $\pi^\star$ upper (resp. lower) bounds its corresponding value function under the

ground truth:

$$\mathbb{E}\Big[V_r^{\pi^\star,p}(s_1,1)\Big] \geq \mathbb{E}\Big[V_{r^\star}^{\pi^\star,p^\star}(s_1,1)\Big] \quad \text{and} \quad \mathbb{E}\Big[V_{c_i}^{\pi^\star,p}(s_1,1)\Big] \leq \mathbb{E}\Big[V_{c_i^\star}^{\pi^\star,p^\star}(s_1,1)\Big] \forall i \in \mathcal{D}.$$

An important block of the analysis for the basic setting is to show that, when using a bonus-enhanced model with valid bonuses, the resulting CMDP is optimistic.

Lemma B.2.2. *If the bonus $\widehat{b}_k(s,a)$ of Eq. Eq. (3.5) in episode k is valid (Definition 3.3.1) for the corresponding* CMDP $\mathcal{M}^{(k)} = \big(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)}\big)$ *then $\mathcal{M}^{(k)}$ is optimistic.*

Proof. We first prove the optimism of the model for the reward objective. More concretely, we show by induction that for any state s, action a, and stage h, $Q_{r^{(k)}}^{\pi^\star,p^{(k)}}(s,a,h) \geq Q_{r^\star}^{\pi^\star,p^\star}(s,a,h)$; taking expectation on the state-action pair of the first state, the claim then follows.

Since the setting ends at episode H, $Q_{r^{(k)}}^{\pi^\star,p^{(k)}}(s,a,H+1) = Q_{r^\star}^{\pi^\star,p^\star}(s,a,H+1) = 0$.

We assume that the inductive hypothesis $Q_{r^{(k)}}^{\pi^\star,p^{(k)}}(s,a,h+1) \geq Q_{r^\star}^{\pi^\star,p^\star}(s,a,h+1)$ (and thus also $V_{r^{(k)}}^{\pi^\star,p^{(k)}}(s,h+1) \geq V_{r^\star}^{\pi^\star,p^\star}(s,h+1)$) holds, and proceed with the inductive step. The Q-functions in question are:

$$\begin{aligned}
Q_{r^{(k)}}^{\pi^\star,p^{(k)}}(s,a,h) &= r^{(k)}(s,a) + \sum_{s'\in\mathcal{S}} p^{(k)}(s'|s,a) V_{r^{(k)}}^{\pi^\star,p^{(k)}}(s',h+1) \\
&\geq r^{(k)}(s,a) + \sum_{s'\in\mathcal{S}} p^{(k)}(s'|s,a) V_{r^\star}^{\pi^\star,p^\star}(s',h+1) \\
Q_{r^\star}^{\pi^\star,p^\star}(s,a,h) &= r^\star(s,a) + \sum_{s'\in\mathcal{S}} p^\star(s'|s,a) V_{r^\star}^{\pi^\star,p^\star}(s',h+1)
\end{aligned}$$

Subtracting, we have:

$$\begin{aligned}
Q_{r^{(k)}}^{\pi^\star,p^{(k)}}(s,a,h) - Q_{r^\star}^{\pi^\star,p^\star}(s,a,h) \geq &\Big(\widehat{r}_k(s,a) + \widehat{b}_k(s,a) - r^\star(s,a)\Big) \\
&+ \sum_{s'\in\mathcal{S}} \Big(\widehat{p}_k(s'|s,a) - p^\star(s'|s,a)\Big) V_{r^\star}^{\pi^\star,p^\star}(s',h+1) \geq 0,
\end{aligned}$$

where the last inequality holds since the bonuses are valid.

The optimism of the model with respect to the consumption objectives follows the same steps altering the direction of the inequalities and setting the estimate as empirical mean minus the bonus. □

We emphasize that our bonus in Eq Eq. (3.5) does not scale polynomially with respect to $|\mathcal{S}|$; despite that, as indicated by the above lemma, it suffices to prove optimism.

B.2.3 Simulation lemma

To prove the Bellman-error regret decomposition, an essential piece is the so called *simulation lemma* (Kearns and Singh, 2002) which we adapt to constrained settings below:

Lemma B.2.3 (Simulation lemma)**.** *For any policy π, any* CMDP *$\mathcal{M} = (p, r, \mathbf{c})$, and any objective $m \in \{r\} \cup \{c_i\}_{i\in\mathcal{D}}$ with corresponding true objective $m^\star \in \{r^\star\} \cup \{c_i^\star\}_{i\in\mathcal{D}}$,, it holds that:*

$$\mathbb{E}^\pi\Big[V_m^{\pi,p}(s_1,1)\Big] - \mathbb{E}^\pi\Big[V_{m^\star}^{\pi,p^\star}(s_1,1)\Big] = \mathbb{E}^\pi\Big[\sum_{h=1}^{H} \textsc{Bell}_m^{\pi,p}(s_h,a_h,h)\Big]. \tag{B.4}$$

Proof. For all of $m \in \{r\} \cup \{c_i\}_{i\in\mathcal{D}}$, rearranging the definitions of Bellman errors, we obtain:

$$Q_m^{\pi,p}(s,a,h) = \Big(\textsc{Bell}_m^{\pi,p}(s,a,h) + m^\star(s,a)\Big) + \sum_{s'\in\mathcal{S}} p^\star(s'|s,a)V_m^{\pi,p}(s',h+1)$$
$$Q_{m^\star}^{\pi,p^\star}(s,a,h) = \Big(\textsc{Bell}_{m^\star}^{\pi,p^\star}(s,a,h) + m^\star(s,a)\Big) + \sum_{s'\in\mathcal{S}} p^\star(s'|s,a)V_{m^*}^{\pi,p^*}(s',h+1)$$

By definition of the Bellman error, the Bellman error with respect to the true model is equal to 0. As a result, subtracting the two above equations, we obtain:

$$Q_m^{\pi,p}(s,a,h) - Q_{m^\star}^{\pi,p^\star}(s,a,h) = \textsc{Bell}_m^{\pi,p}(s,a,h) + \sum_{s'\in\mathcal{S}} p^\star(s'|s,a)\Big(V_m^{\pi,p}(s',h+1) - V_{m^\star}^{\pi,p^\star}(s',h+1)\Big).$$

Taking expectation over policy π to select a, the initial state s_1, and setting $h = 1$, we obtain:

$$\mathbb{E}_{s_1}\Big[V_m^{\pi,p}\big(s(1),1\big) - V_{m^\star}^{\pi,p^\star}\big(s_1,1\big)\Big] = \mathbb{E}^{\pi}\Big[\textsc{Bell}_m^{\pi,p}\big(s_1,a_1,1\big)\Big] + \mathbb{E}^{\pi}\Big[V_m^{\pi,p}\big(s_2,2\big) - V_{m^\star}^{\pi,p^*}\big(s_2,2\big)\Big].$$

Recursively bounding the second term of the RHS as above concludes the lemma. □

B.2.4 Bellman-error regret decomposition (Proposition 3.3.3)

Proof of Proposition 3.3.3. The consumption requirement Eq. (3.6) for resource i follows by applying the simulation lemma (Lemma B.2.3) on cMDP $\mathcal{M}^{(k)}$ and objective $m = c_i^{(k)}$ (with corresponding true objective $m^\star = c_i^\star$) and using that π_k is feasible for $\textsc{ConPlanner}(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)})$:

$$\begin{aligned}\mathbb{E}^{\pi_k,p^\star}\Big[\sum_{h=1}^{H} c^\star(s_h,a_h,i)\Big] = \mathbb{E}^{\pi_k}\Big[V_{c_i^\star}^{\pi,p^\star}(s_1,1)\Big] &= \mathbb{E}\Big[V_{c_i}^{\pi_k,p}(s_1,1)\Big] - \mathbb{E}^{\pi_k}\Big[\sum_{h=1}^{H}\textsc{Bell}_{c_i^{(k)}}^{\pi_k,p^{(k)}}\big(s_h,a_h,h\big)\Big] \\ &\le \xi(i) + \mathbb{E}^{\pi_k}\Big[\sum_{h=1}^{H}\Big|\textsc{Bell}_{c_i^{(k)}}^{\pi_k,p^{(k)}}\big(s_h,a_h,h\big)\Big|\Big]\end{aligned}$$

Regarding the reward requirement Eq. (3.6), what we wish to bound is:

$$\mathbb{E}^{\pi^\star,p^\star}\Big[\sum_{h=1}^{H} r^\star(s_h,a_h)\Big] - \mathbb{E}^{\pi_k,p^\star}\Big[\sum_{h=1}^{H} r^\star(s_h,a_h)\Big] = \mathbb{E}\Big[V_{r^\star}^{\pi^\star,p^\star}(s_1,1)\Big] - \mathbb{E}\Big[V_{r^\star}^{\pi_k,p^\star}(s_1,1)\Big]$$

the validity of the bonus implies that the model $\mathcal{M}^{(k)}$ is optimistic (Lemma B.2.2), i.e., we have that $\mathbb{E}\Big[V_{r^\star}^{\pi^\star,p^\star}(s_1,1)\Big] \le \mathbb{E}\Big[V_{r^{(k)}}^{\pi^\star,p^{(k)}}(s_1,1)\Big]$. If $\pi^\star$ is feasible for $\textsc{ConPlanner}(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)})$ then, since π_k is the maximizer for this program:

$$\begin{aligned}\mathbb{E}\Big[V_{r^{(k)}}^{\pi^\star,p^{(k)}}(s_1,1)\Big] - \mathbb{E}\Big[V_{r^\star}^{\pi_k,p^\star}(s_1,1)\Big] &\le \mathbb{E}\Big[V_{r^{(k)}}^{\pi_k,p^{(k)}}(s_1,1)\Big] - \mathbb{E}\Big[V_{r^\star}^{\pi_k,p^\star}(s_1,1)\Big] \\ &= \mathbb{E}^{\pi_k}\Big[\sum_{h=1}^{H}\textsc{Bell}_{r^{(k)}}^{\pi_k,p^{(k)}}\big(s_h,a_h,h\big)\Big]\end{aligned}$$

where the last equality holds by applying the simulation lemma with $m = r$. Hence, this proves Eq. (3.6).

What is left to show is that $\pi^\star$ is indeed feasible for $\textsc{ConPlanner}(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)})$. Since $\mathcal{M}^{(k)}$ is optimistic and $\pi^\star$ is feasible for the ground truth $\mathcal{M}^\star$, for all resources $i \in \mathcal{D}$:

$$\mathbb{E}\Big[V_{c_i^{(k)}}^{\pi^\star, p^{(k)}}(s_1, 1)\Big] \leq \mathbb{E}\Big[V_{c_i^\star}^{\pi^\star, p^\star}(s_1, 1)\Big] \leq \xi(i).$$

This completes the proof of the proposition. □

B.2.5 Bounding the Bellman error

We now provide an upper bound on the Bellman error which arises in the RHS of the regret decomposition (Proposition 3.3.3).

Lemma B.2.4. *Let $\epsilon > 0$. If the bonus $\widehat{b}_k$ is valid for all episodes k simultaneously then, with probability at least $1-\delta$: for all objectives $m^{(k)} \in \{r^{(k)}\} \cup \{c_i^{(k)}\}_{i \in \mathcal{D}}$, transitions $p = p^{(k)}$, and stages h, the Bellman error at episode k is upper bounded by:*

$$\left|\textsc{Bell}_{m^{(k)}}^{\pi_k, p^{(k)}}(s, a, h)\right| \leq 4H^2 \sqrt{\frac{2S \ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}{N_k(s,a)}} + \epsilon S.$$

Proof of Lemma B.2.4. Let Ψ be an ϵ-net in $[-2H^2, 2H^2]^S$. For a fixed value $\bar{V} \in \Psi$, similar to Lemma 3.3.2, with probability $1-\delta'$, simultaneously for all states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, steps $h \in [H]$, episodes $k \in [K]$, and objectives $m^{(k)} \in \{r^{(k)}\} \cup \{c_i^{(k)}\}_{i \in \mathcal{D}}$, it holds that:

$$\begin{aligned}\Big|m^{(k)}(s,a) - m^\star(s,a) + \sum_{s' \in \mathcal{S}} \Big(p(s'|s,a) - p^\star(s'|s,a)\Big)\bar{V}(s')\Big| \\ \leq \widehat{b}_k(s,a) + 2H^2 \sqrt{\frac{2\ln\left(8SAH(d+1)k^2/\delta'\right)}{N_k(s,a)}}\end{aligned}$$

Since Ψ is an ϵ-net for $\bar{V}$, there are $(2H^2/\epsilon)^S$ potential values. In order to have the above hold simultaneously for all these values with probability $1-\delta$, we need to set $\delta' = \frac{\delta}{(2H^2/\epsilon)^S}$.

Since the value $\left(p^{(k)}(s'|s,a) - p^\star(s'|s,a)\right)V^{\pi_k,p}_{m^{(k)}}(s', h+1)$ is in $[-2H^2, 2H^2]$ for all s', it holds that there exists a value V in the ϵ-net with distance at most ϵS. As a result, since $\widehat{b}_k(s,a)$ is valid for k:

$$\begin{aligned}\left|\textsc{Bell}^{\pi_k,p^{(k)}}_{m^{(k)}}(s,a,h)\right| &\leq \left|m^{(k)}(s,a) - m^\star(s,a) + \sum_{s'\in\mathcal{S}}\left(p^{(k)}(s'|s,a) - p^\star(s'|s,a)\right)V(s')\right| \\ &\quad + \left|\sum_{s'\in\mathcal{S}}\left(p^{(k)}(s'|s,a) - p^\star(s'|s,a)\right)\left(V(s') - V^{\pi_k,p^{(k)}}_{m^{(k)}}(s', h+1)\right)\right| \\ &\leq \widehat{b}_k(s,a) + 2H^2\sqrt{\frac{2S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}{N_k(s,a)}} + \epsilon S.\end{aligned}$$

Upper bounding $\widehat{b}_k(s,a) \leq 2H^2\sqrt{\frac{2S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}{N_k(s,a)}}$ completes the lemma.

□

B.2.6 Final guaraantee for the basic setting (Theorem 3.3.4)

Proof. The failure probability of the algorithm is δ due to the validity of bonus $\widehat{b}_k(s,a)$ (Lemma 3.3.2) and another δ by the bound on Bellman error (Lemma B.2.4). When neither failure events occur (probability $1-2\delta$), Proposition 3.3.3 upper bounds either of reward or consumption regret by $\mathbb{E}^{\pi_k}\left[\left|\textsc{Bell}^{\pi_k,p^{(k)}}_{m^{(k)}}(s_h,a_h,h)\right|\right]$. By Lemma B.2.4, the Bellman error at episode t, for $\epsilon > 0$, is at most:

$$\left|\textsc{Bell}^{\pi_t,p^{(t)}}_{m^{(t)}}(s_{t,h},a_{t,h},h)\right| \leq 4H^2\sqrt{\frac{2S\ln\left(16SAH^2(d+1)t^2/(\epsilon\delta)\right)}{N_t(s,a)}} + \epsilon S$$

Summing across all $h = 1\ldots H$ and $t = 1,\ldots,k$, the sum of Bellman errors is at most:

$$\begin{aligned}&\sum_{t=1}^{k}\sum_{h=1}^{H}\left|\textsc{Bell}^{\pi_t,p^{(t)}}_{m^{(t)}}(s_{t,h},a_{t,h},h)\right| \\ &\quad\leq \sum_{t=1}^{k}\sum_{h=1}^{H}\left(4H^2\sqrt{\frac{2S\ln\left(16SAH^2(d+1)t^2/(\epsilon\delta)\right)}{N_t(s,a)}} + \epsilon S\right) \\ &\quad\leq \sum_{s,a}\left(\sum_{j=1}^{2H}4H^2\sqrt{2S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}\right.\end{aligned}$$

$$+\sum_{j=H+1}^{N_k(s,a)} 4H^2\sqrt{\frac{4S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}{j}}+\epsilon S\Big)$$

The second inequality follows since a particular state-action pair may have the same visitations for H times (as we only update this quantity at the end of the episode). To avoid incurring an additional dependence on H, we separate the first H visitations of each state-action pair and treat the bound as if $j=1$ for them. [1] For the remaining visitations, j and $N_k(s,a)$ are always within a factor of 2 and this factor therefore appears within the square root.

We now bound the second term:

$$\begin{aligned}
&\sum_{s,a}\Big(\sum_{j=H+1}^{N_k(s,a)} 4H^2\sqrt{\frac{4S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}{j}}+\epsilon S\Big)\\
&\le 4SAH^2\sqrt{N_k(s,a)\ln\left(N_k(s,a)\right)\cdot 4S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}+\epsilon kHS\\
&\le 4SAH^2\sqrt{\frac{kH\cdot 4S\cdot\ln(k)\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}{SA}}+\epsilon kHS\\
&\le 16S\sqrt{AH^5}\cdot\sqrt{k}\cdot\sqrt{\ln(k)\ln\left(2SAH(d+1)k/\delta\right)}+1.
\end{aligned}$$

The last inequality holds by setting $\epsilon=\frac{1}{kHS}$.

The first term can be bounded by additive terms that depend only logarithmically on k:

$$\sum_{s,a}\Big(\sum_{j=1}^{2H} 4H^2\sqrt{2S\ln\left(16SAH^2(d+1)k^2/(\epsilon\delta)\right)}\le 32S^{3/2}AH^3\sqrt{\ln(2SAH(d+1)k/\delta)}$$

As a result:

$$\begin{aligned}
\sum_{t=1}^{k}\sum_{h=1}^{H}\left|\textsc{Bell}_{m^{(t)}}^{\pi_t,p^{(t)}}(s_{t,h},a_{t,h},h)\right| &\le 16S\sqrt{AH^5}\sqrt{k}\cdot\sqrt{\ln(k)\ln\left(2SAH(d+1)k/\delta\right)}+1\\
&\quad+32S^{3/2}AH^3\sqrt{\ln\left(2SAH(d+1)k/\delta\right)}
\end{aligned}$$

[1] The reason why we sum until $2H$ in the first term is since we want to consider all such visitations that occur in an episode that started with $N_k(s,a)<H$; the additional factor of 2 in the second term comes since, $j/N_t(s,a)\le 2$ if $N_t(s,a)\ge H$ and the j-th visitation happens within the same episode.

Now we link the additive Bellman error to the expected sum of Bellman errors under the expectation of the policies $\{\pi_t\}$ (as needed by Proposition 3.3.3) via a simple martingale argument. From Lemma B.6.3, with probability at least $1-\delta$, we have:

$$\left|\sum_{t=1}^{k}\sum_{h=1}^{H}\left|\textsc{Bell}_{m^{(t)}}^{\pi_t,p^{(t)}}(s_{t,h},a_{t,h},h)\right| - \sum_{t=1}^{k}\sum_{h=1}^{H}\mathbb{E}^{\pi_t}\left[\sum_{h=1}^{H}\left|\textsc{Bell}_{m^{(t)}}^{\pi_t,p^{(t)}}(s_h,a_h,h)\right|\right]\right| \leq 5H^{2.5}\sqrt{2\ln(4k^2/\delta)k},$$

where we use the fact that $|\textsc{Bell}_m^{\pi,p}| \leq 5H^2$ due to of $Q_m^{\pi,p}(s,a) \in [0,2H^2]$, $m^\star(s,a) \in [0,1]$, and $V_m^{\pi,p}(s) \in [0,2H^2]$. Combining the above, we conclude the proof. □

B.3 Analysis: concave-convex setting (Section 3.4)

In this section, we prove the main guarantee for the convex-concave setting. Since the regret decomposition of the basic setting (Proposition 3.3.3) does not hold direclty as f and g are not linear, we need to create an analogous regret decomposition (Proposition B.3.2) for the convex-concave setting. This can be done by leveraging the Lipschitzness of the functions. Armed with this new regret decomposition, we can directly call the results we have for for the basic setting (e.g., upper bounds of Bellman errors) to conclude the regret analysis for the convex-concave setting. The first step leading to this regret decomposition is to show that $\pi^\star$ is a feasible solution of ConvexConPlanner.

B.3.1 Feasibility of optimal policy in concave-convex setting (Lemma B.3.1)

Lemma B.3.1. *If the bonus $\widehat{b}_k$ is valid (in the sense of Definition 3.3.1) then policy $\pi^\star$ that maximizes the objective of the convex-concave setting is feasible in* ConvexConPlanner.

Proof. Unlike the linear case, the feasibility of $\pi^\star$, requires more care. Applying the same dynamic

programming arguments as in Lemma B.2.2, it follows that:

$$\forall i \in \mathcal{D}: \qquad \mathbb{E}\Big[V^{\pi^\star,p^{(k)}}_{\widehat{c}_{i,k}-b_k}(s_1,1)\Big] \leq \mathbb{E}_s\Big[V^{\pi^\star,p^\star}_{c^\star_i}(s_1,1)\Big] \leq \mathbb{E}\Big[V^{\pi^\star,p^{(k)}}_{\widehat{c}_{i,k}+b_k}(s_1,1)\Big].$$

Letting $\widetilde{g}(\alpha) = \mathbb{E}\Big[V^{\pi^\star,p^{(k)}}_{\widehat{c}_{i,k}+\alpha b_k}(s(1),1)\Big]$, the above can be rewritten as:

$$\forall i \in \mathcal{D}: \qquad \widetilde{g}(-1) \leq \mathbb{E}\Big[V^{\pi^\star,p^\star}_{c^\star_i}(s_1,1)\Big] \leq \widetilde{g}(1).$$

Since $\widetilde{g}(\cdot)$ is the expected value over the same policy and under the same transitions, it is continuous with respect to its argument. As a result, applying mean-value theorem on each i separately, there exists some α_i such that $\widetilde{g}(\alpha_i) = \mathbb{E}_s\Big[V^{\pi^\star,p^\star}_{c^\star_i}(s_1,1)\Big]$. Due to the feasibility of $\pi^\star$ on the true transitions and consumptions, it holds that $g\Big(\widetilde{\mathbf{g}}(\alpha_i)\Big) \leq 0$. Hence, selecting estimates $\widehat{c}_{i,k} + \alpha_i \widehat{b}_k$ creates a feasible solution for $\pi^\star$ under the estimated transitions of the ConvexConPlanner program. The final value of $\pi^\star$ at this program maximizes the objective retaining feasibility; hence the existence of one feasible selection of consumption estimates concludes the proof of the lemma. □

We conclude by remarking that proving optimism feasibility for the concave-convex setting in multiple-step RL setting is more challenging than that in single-step multi-arm bandit setting Agrawal and Devanur (2014) since in bandits, there are no transitions. In the proof above, to show that $\pi^\star$ is feasible in ConvexConPlanner which is defined with respect to $p^{(k)}$, we leverage the fact that $\widetilde{g}(\alpha)$ is continuous and a novel application of mean-value theorem to link $\pi^\star$'s performance in the optimistic model $\mathbb{E}\Big[V^{\pi^\star,p^{(k)}}_{\widehat{c}_{i,k}+\alpha_i b_k}(s_1,1)\Big]$ and $\pi^\star$'s performance under the real model $\mathbb{E}_s\Big[V^{\pi^\star,p^\star}_{c^\star_i}(s_1,1)\Big]$.

B.3.2 Regret decomposition for concave-convex setting

Using the Lipschitz continuous assumption of f and g, we can decompose the regret into a sum of Bellman errors as before, but scaled by the Lipschitz constant this time.

Proposition B.3.2. *Let L be the Lipschitz constant for f and g. If $\widehat{b}_k(s,a,\delta)$ is valid for all episodes k simultaneously then the per-episode reward and consumption regrets can be upper bounded*

by:

$$f\left(\mathbb{E}^{\pi^\star,p^\star}\left[\sum_{h=1}^{H} r^\star(s_h,a_h)\right]\right) - f\left(\mathbb{E}^{\pi_k,p^\star}\left[\sum_{h=1}^{H} r^\star(s_h,a_h)\right]\right) \le L\cdot\mathbb{E}^{\pi_k}\left[\sum_{h=1}^{H}\textsc{Bell}_{r^{(k)}}^{\pi_k,p^{(k)}}(s_h,a_h,h)\right]\right)$$

$$g\left(\mathbb{E}^{\pi_k,p^\star}\left[\sum_{h=1}^{H}\mathbf{c}^\star(s_h,a_h,i)\right]\right) \le L\sum_{i\in\mathcal{D}}\cdot\mathbb{E}^{\pi_k}\left[\sum_{h=1}^{H}\left|\textsc{Bell}_{c_i^{(k)}}^{\pi_k,p^{(k)}}(s_h,a_h,h)\right|\right]$$

Proof. We first prove the reward requirement. Let $r(\pi)$ be the solution of the inner maximization program for policy π, and we define $r^{(k)} = r(\pi_k)$. For notational convenience, we denote $V_m^{\pi,p} = \mathbb{E}^{\pi,p}\left[V_m^{\pi,p}\right]$ Since $r^\star(s,a) \in [\widehat{r}(s,a) - \widehat{b}_k(s,a,\delta), \widehat{r}(s,a) + \widehat{b}_k(s,a,\delta)]$ and the bonus $\widehat{b}_k$ is valid, similar to Lemma B.2.2, it holds:

$$V_{r^\star}^{\pi^\star,p^\star} \in \left[V_{\widehat{r}-b}^{\pi^\star,p^{(k)}}, V_{\widehat{r}+b}^{\pi^\star,p^{(k)}}\right]. \tag{B.5}$$

As a result, by mean-value theorem, there exists $\alpha \in [-1,1]$ such that $V_{r^\star}^{\pi^\star,p^\star} = V_{\widehat{r}+\alpha b}^{\pi^\star,p^{(k)}}$. Since π_k is the maximizer of ConvexConPlanner and $\pi^\star$ is feasible for that program, it holds that:

$$f\left(V_{r(\pi_k)}^{\pi_k,p^{(k)}}\right) \ge f\left(V_{r(\pi^\star)}^{\pi^\star,p^{(k)}}\right) \ge f\left(V_{\widehat{r}+\alpha b}^{\pi^\star,p^{(k)}}\right) = f\left(V_{r^\star}^{\pi^\star,p^\star}\right), \tag{B.6}$$

where the second-to-last inequality holds since $r(\pi^\star)$ is the maximizer of the inner program for $\pi^\star$ and the equality holds by Eq. (B.5).

We are now ready to provide the equivalent of the regret decomposition:

$$\begin{aligned} f(V_{r^\star}^{\pi^\star,p^\star}) - f(V_{r^\star}^{\pi_k,p^\star}) &\le f(V_{r(\pi_k)}^{\pi_k,p^{(k)}}) - f(V_{r^\star}^{\pi_k,p^\star}) \le L\cdot\left|V_{r(\pi_k)}^{\pi_k,p^{(k)}} - V_{r^\star}^{\pi_k,p^\star}\right| \\ &\le L\cdot\mathbb{E}^{\pi_k}\left(\sum_{h=1}^{H}\textsc{Bell}_{r^{(k)}}^{\pi_k,p^{(k)}}(s_h,a_h,h)\right) \end{aligned}$$

where the first inequality holds by Eq. (B.6). the second inequality by Lipschitzness and the last inequality holds by simulation lemma (Lemma B.2.3).

For the consumption requirement, since π_k is feasible in ConvexConPlanner, denoting again by $\mathbf{c}(\pi)$ the consumption in the maximizer for policy π in the inner mathematical program. Same as

above we define $\mathbf{c}^{(k)} = \mathbf{c}(\pi_k)$. It holds that:

$$g\Big(\mathbb{E}^{\pi_k, p^{(k)}}\Big[\sum_{h=1}^{H} \mathbf{c}_h(\pi_k)\Big]\Big) \leq 0 \tag{B.7}$$

As a result,

$$\begin{aligned} g\Big(\mathbb{E}^{\pi_k, p^\star}\Big[\sum_{h=1}^{H} \mathbf{c}_h^\star\Big]\Big) - g\Big(\mathbb{E}^{\pi_k, p^{(k)}}\Big[\sum_{h=1}^{H} \mathbf{c}_h(\pi_k)\Big]\Big) &\leq L \left\| \mathbb{E}^{\pi_k, p^\star}\Big[\sum_{h=1}^{H} \mathbf{c}_h^\star\Big] - \mathbb{E}^{\pi_k, p^{(k)}}\Big[\sum_{h=1}^{H} \mathbf{c}_h(\pi_k)\Big] \right\|_1 \\ &= L \sum_{i \in \mathcal{D}} \left| \mathbb{E}^{\pi_k, p^\star}\Big[\sum_{h=1}^{H} c_h^\star(i)\Big] - \mathbb{E}^{\pi_k, p^{(k)}}\Big[\sum_{h=1}^{H} c_h(\pi_k, i)\Big] \right| \\ &\leq L \cdot \sum_{i \in \mathcal{D}} \mathbb{E}^{\pi} \left(\sum_{h=1}^{H} \left| \textsc{Bell}_{c_i^{(k)}}^{\pi_k, p^{(k)}}(s_h, a_h, h) \right| \right), \end{aligned}$$

where again we applied Lipschitness and simulation lemma. □

B.3.3 Concave-convex theorem (Theorem 3.4.1)

Proof of Theorem 3.4.1. The proof follows similarly to the proof of Theorem 3.3.4 by replacing Proposition 3.3.3 with Proposition B.3.2. The linear dependency on d in the consumption regret comes from the fact that the Lipschitzness of g is defined in L1 norm. □

B.4 Analysis: Knapsack setting (Section 3.5)

In this section, we prove the guarantee for the hard-constraint setting. The goal is to show that over K episodes, our algorithm has sublinear reward regret comparing to the best dynamic policy (formally defined in Appendix B.4.2), while satisfying hard budget constraints with high probability.

B.4.1 Theorem with hard constraints (Theorem 3.5.1)

Proof of Theorem 3.5.1. We denote by Opt the expected total reward of $\pi^\star$. Consider now the policy $\widetilde{\pi}^\star$ that selects the null policy with probability ϵ and follows $\pi^\star$ otherwise. This policy is feasible for Eq. (3.8); as a result the expected reward $\widetilde{\pi}^\star$ for Eq. (3.8) is at least $(1-\epsilon)$Opt. Since

the total reward is upper bounded by KH, it therefore holds that:

$$\sum_{k=1}^{K} \mathbb{E}^{\widetilde{\pi}^\star}\Big[\sum_{h=1}^{H} r^\star\big(s_h, a_h\big)\Big] \geq (1-\epsilon)\textsc{Opt} \geq \textsc{Opt} - \epsilon KH \tag{B.8}$$

In the high-probability event where the regret guarantee of $\textsc{AggReg}(\delta)$ does not fail, the reward of the algorithm is at least:

$$\sum_{k=1}^{K}\sum_{h=1}^{H} r_{k,h} \geq \sum_{k=1}^{K} \mathbb{E}^{\widetilde{\pi}^\star}\Big[\sum_{h=1}^{H} r^\star(s_h, a_h)\Big] - \textsc{AggReg}(\delta), \tag{B.9}$$

Combining Eq. (B.8) and Eq. (B.9), with probability $1-\delta$, the reward regret with respect to $\pi^\star$ is at most:

$$\textsc{RewReg}(\mathrm{K}) \leq \frac{1}{K}\textsc{AggReg}(\delta) + \epsilon H \tag{B.10}$$

We now focus on the consumption. Since we optimize Eq. (3.8), for any resource $i \in \mathcal{D}$, when the regret guarantee $\textsc{AggReg}(\delta)$ against $\widetilde{\pi}^\star$ does not fail and given that $\widetilde{\pi}^\star$ is feasible for Eq. (3.8), it holds that:

$$\sum_{k=1}^{K}\sum_{h=1}^{H} c_{k,h,i} \leq \sum_{k=1}^{K} \mathbb{E}^{\widetilde{\pi}^\star}\Big[\sum_{h=1}^{H} c\big(s_h, a_h, i\big)\Big] + \textsc{AggReg}(\delta) \leq (1-\epsilon)B_i + \textsc{AggReg}(\delta)$$

Hence, when the regret guarantee $\textsc{AggReg}(\delta)$ does not fail, the consumption is less than B_i for all i as long as $\epsilon \geq \frac{\textsc{AggReg}(\delta)}{\min_i B_i}$. Moreover ϵ is a probability as a result it should also be less than 1 which holds when $\min_i B_i \geq \textsc{AggReg}(\delta)$. Applying on Eq. (B.10) and assuming without loss of generality that $KH > \min_i B_i$ (otherwise the setting is essentially unconstrained), the reward regret is at most

$$\textsc{RewReg}(K) \leq \frac{2H\textsc{AggReg}(\delta)}{\min_i B_i}.$$

□

B.4.2 Dynamic policy benchmark

We call a policy *dynamic* if it maps the entire history to a distribution over the action space. Specifically we denote history $\mathcal{H}_{k,h}$ as the history that contains all the information from the beginning of the first episode to the end of the step $h-1$ at the k-th episode plus the state at step h in episode k. At any episode k and step h, a dynamic policy $\widetilde{\pi}(\cdot|\mathcal{H}_{k;h}) \in \Delta(\mathcal{A})$ maps history $\mathcal{H}_{k;h}$ to a distribution over action space. We denote Π_{dynamic} as the set of all dynamic policies that satisfies the budget constraints deterministically, i.e., for any $\widetilde{\pi} \in \Pi_{\text{dynamic}}$, when executed for K episodes in the MDP, we have $\sum_{k=1}^{K}\sum_{h=1}^{H} c_i(s_{k,h}, a_{k,h}) \leq B_i$ for all $i \in \mathcal{D}$, deterministically. Ideally we want to compare against the best dynamic policy that maximizes the expected total reward $\max_{\widetilde{\pi}\in\Pi_{\text{dynamic}}} \mathbb{E}^{\widetilde{\pi}}\left[\sum_{k=1}^{K}\sum_{h=1}^{K} r_{k,h}\right]$. We denote such an optimal dynamic policy as $\widetilde{\pi}^{\star}$ and its expected total reward across K episodes as

$$\textsc{Opt} := \max_{\widetilde{\pi}\in\Pi_{\text{dynamic}}} \mathbb{E}^{\widetilde{\pi}}\left[\sum_{k=1}^{K}\sum_{h=1}^{K} r_{k,h}\right].$$

The lemma below shows that indeed the stationary Markovian policy $\pi^{\star}$ actually achieves no smaller expected total reward across K episodes than that of the best dynamic policy.

Lemma B.4.1. *The reward of the policy $\pi^{\star}$ maximizing program Eq. (3.1) with $\xi(i) = \frac{B_i}{K}$ is at least as large as the per-episode reward of the optimal dynamic policy that is subject to hard constraints instead:*

$$\mathbb{E}^{\pi^{\star}}\Big[\sum_{h=1}^{H} r^{\star}\big(s_h, a_h\big)\Big] \geq \frac{1}{K}\max_{\widetilde{\pi}\in\Pi_{dynamic}} \mathbb{E}^{\widetilde{\pi}}\Big[\sum_{k=1}^{K}\sum_{h=1}^{H} r(s_{k,h}, a_{k,h})\Big] = \frac{\textsc{Opt}}{K}.$$

Proof. Denote $\widetilde{\pi}^{\star}$ as the optimal dynamic policy from Π_{dynamic}. Any policy induces a state-action distribution at episode k and stage h, denoted as $\rho_{\widetilde{\pi}}(s, a; h, k)$, which stands for the probability of $\widetilde{\pi}$ visits state-action pair (s, a) at stage h in episode k. Denote $\rho_{\widetilde{\pi}}(s, a; h) = \sum_{k=1}^{K} \rho_{\widetilde{\pi}}(s, a; h, k)/K$

which stands for the probability of $\widetilde{\pi}$ visiting (s, a) at stage h. We have:

$$\sum_a \rho_{\widetilde{\pi}}(s', a; h, k) = \sum_{s,a} \rho_{\widetilde{\pi}}(s, a; h-1, k) p^\star(s'|s, a), \forall s',$$

due to the Markovian transition $p^\star(s'|s, a)$, which implies that:

$$\sum_a \rho_{\widetilde{\pi}}(s', a; h) = \sum_{s,a} \rho_{\widetilde{\pi}}(s, a; h-1) p^\star(s'|s, a), \forall s'.$$

Hence, $\rho_{\widetilde{\pi}}(s, a; h)$ satisfies the flow constraints, and hence induces a stationary Markovian policy:

$$\pi_{\widetilde{\pi}}(a|s) \propto \rho_{\widetilde{\pi}}(s, a; h) / \sum_a \rho_{\widetilde{\pi}}(s, a; h),$$

and $\pi_{\widetilde{\pi}}$ induces state-action visitation distribution that are exactly equal to $\rho_{\widetilde{\pi}}(s, a; h)$.

Note that $\widetilde{\pi}^\star$ satisfies the budget constraints deterministically, which means in expectation, it will satisfies the constraints as well, i.e.,

$$\sum_{k=1}^{K} \sum_{h=1}^{H} \sum_{(s,a)} \rho_{\widetilde{\pi}^\star}(s, a; h) c_i(s, a) \leq B_i, \quad \forall i \in \mathcal{D},$$

which implies that in expectation, for $\pi_{\widetilde{\pi}^\star}$, we have that for all $i \in \mathcal{D}$:

$$\mathbb{E}^{\pi_{\widetilde{\pi}^\star}} \left[\sum_{h=1}^{H} c_i(s_h, a_h) \right] = \sum_{h=1}^{H} \sum_{(s,a)} \rho_{\pi_{\widetilde{\pi}^\star}}(s, a, h) c_i(s, a) = \sum_{k=1}^{K} \sum_{h=1}^{H} \sum_{(s,a)} \rho_{\widetilde{\pi}^\star}(s, a; h) c_i(s, a) / K \leq B_i / K.$$

This means that $\pi_{\widetilde{\pi}^\star}$ is a feasible solution of the hard-constraint program.

Similarly, we have that the expected per-episode total reward of $\widetilde{\pi}^\star$ is the same as the expected total reward of $\pi_{\widetilde{\pi}^\star}$:

$$\mathbb{E}^{\pi_{\widetilde{\pi}^\star}} \Big[\sum_{h=1}^{H} r_h(s_h, a_h) \Big] = \frac{1}{K} \mathbb{E}^{\widetilde{\pi}^\star} \left[\sum_{k=1}^{K} \sum_{h=1}^{H} r_{k,h} \right].$$

Hence, due to the optimality of $\pi^\star$, we immediately have:

$$\mathbb{E}^{\pi^\star}\Big[\sum_{h=1}^{H} r_h\Big] \geq \mathbb{E}^{\pi_{\widetilde{\pi}^\star}}\Big[\sum_{h=1}^{H} r_h\Big] = \frac{1}{K}\mathbb{E}^{\widetilde{\pi}^\star}\Big[\sum_{k=1}^{K}\sum_{h=1}^{H} r_{k,h}\Big].$$

□

Since our approach incurs sublinear regret with respect to $\pi^\star$, it follows from the above lemma that it incurs sublinear regret with respect to Opt – the total reward across K episodes from the best dynamic policy.

B.5 Experimental details

In the experiments, both ApproPO and RCPO use the same policy gradient algorithm, specifically, Advantage Actor-Critic (A2C) Mnih et al. (2016) as the learning algorithm. We implemented ConRL using two version of LagrConPlanner (see algorithm 10 below) as ConPlanner in which the planner is either value iteration (exact planner) or A2C (approximate planner similar to Dyna model-base RL Sutton (1991)) using fictitious samples. All three algorithms have outer-loop learning rates which we tuned while hyperparameters used for A2C is same across all three methods. Here, we report the result for the best learning rate for each method.

B.5.1 LagrConPlanner

Our theoretical results posit that ConPlanner is solved optimally, which can be indeed achieved via linear programming (see section B.1). However in our experiments it suffices to use a general heuristic for ConPlanner. Our approach is to Lagrangify the constraints, and create a min-max mathematical program with the Lagrangean objective:

$$\min_{\forall i\in\mathcal{D}:\lambda(i)\leq 0}\ \max_{\pi}\ \Big(\mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} r^{(k)}\big(s_h,a_h\big)\Big] + \sum_{i\in\mathcal{D}}\lambda(i)\Big(\mathbb{E}^{\pi,p^{(k)}}\Big[\sum_{h=1}^{H} c^{(k)}\big(s_h,a_h,i\big)\Big] - \xi(i)\Big).$$

Define pseudo-reward $r_\lambda^{(k)}$ as

$$r_\lambda^{(k)}(s,a) = r^{(k)}(s,a) + \sum_{i \in D} \lambda(i)[c^{(k)}(s,a) - \xi(i)]$$

With a fixed choice of Lagrange multipliers $\{\lambda(i)\}_{i \in \mathcal{D}}$, this is an unconstrained *planning* program which we refer to as PLANNER$(p^{(k)}, r_\lambda^{(k)})$ and it can be solved by a planning oracle.

We update Lagrange multipliers via projected gradient descent Zinkevich (2003). The overhead of CONPLANNER is computational, as we do not require new samples. The full procedure is in Algorithm 10. The near-optimality of Algorithm 10 can be proved by leveraging the fact that we are iteratively updating π and λ using no-regret online learning procedure (Best Response for π and OGD for λ) (e.g., Cesa-Bianchi and Lugosi (2006)). We omit the analysis for Algorithm 10 as it is not the main focus of this work.

Algorithm 10 Lagrangean-based Constrained Planner (LAGRCONPLANNER)

hyper-parameters: learning rate η
Input: Estimates $\widehat{p}_k$, $\widehat{r}_k$, $\widehat{\mathbf{c}}_k$ and bonus $\hat{b}_k$
Compute bonus-enhanced model $\mathcal{M}^{(k)} = \left(p^{(k)}, r^{(k)}, \mathbf{c}^{(k)}\right)$

$$p^{(k)}(s'|s,a) = \widehat{p}_k(s'|s,a) \quad \forall s, a, s'$$

$$r^{(k)}(s,a) = \widehat{r}_k(s,a) + \widehat{b}_k(s,a) \quad \forall s, a$$

$$c^{(k)}(s,a,i) = \widehat{c}_k(s,a,i) - \widehat{b}_k(s,a) \quad \forall s, a, i \in \mathcal{D}$$

Initialize Lagrange parameters $\lambda_1(i) \quad 0$ for $i \in \mathcal{D}$
for Iteration k from 1 to N **do**
  Define

$$r_\lambda^{(k)}(s,a) = r^{(k)}(s,a) + \sum_{i \in D} \lambda(i)[c^{(k)}(s,a) - \xi(i)]$$

  $\pi_k \quad$ PLANNER$(p^{(k)}, r_\lambda^{(k)})$
  $\lambda_{k+1}(i) \quad \min\left\{0, \lambda_k(i) - \eta \mathbb{E}^{\pi_k, p^{(k)}}\left[\sum_{h=1}^H [c^{(k)}(s_h, a_h, i)] - \xi(i)\right]\right\} \quad \forall i \in \mathcal{D}$
Return mixture policy $\pi := \frac{1}{N}\sum_{k=1}^N \pi_k$

In our experiments, two versions of PLANNER have been implemented: Value Iteration (exact

planner) and A2C with fictitious samples (approximate planner)

Value Iteration as Planner This program takes p and r as input. Finite horizon value iteration is simply solving the following acyclic dynamic program.

$$Q(s,a,h) = \begin{cases} 0 & h = H+1 \\ r(s,a) + \sum_{s'} \left[p(s'|s,a) \max_{a'} Q(s',a',h+1) \right] & h = 1,\dots,H \end{cases}$$

then the optimal policy for step h is computed as

$$\pi_h(s) = \text{argmax}_a Q(s,a,h)$$

and the algorithm returns the H-step policy

$$\pi = (\pi)_{h=1}^{H}$$

A2C with fictitious samples as Planner This program takes p and r as input, then, using model p and r it generates episodes and use those samples to train our A2C agent. Since we only call this subroutine with our estimated model ($p \leftarrow \hat{p}$ and $r \leftarrow \hat{r}$) those episodes are fictitious (not adding to sample complexity). The algorithm is given Algorithm 11 (Parameterized policy π_θ and value function estimate V_θ)

B.5.2 Hyperparameter Tuning

Both ConRL-A2C and RCPO used the Adam optimizer. For our method we performed a hyperparamter search on both domains over the following values in Table B.1; selected values are given in Table B.2. Note that reset row refers to when using the A2C planner during each call to the planner we tried the following options: (warm-start) reuse previous weights and reset the optimizer (warm -start), or (continue) continue learning using the previous weights (continue) and optimizer, or (none) reset the model weights and optimizer.

Algorithm 11 A2C planner with fictitious samples

hyper-parameters: learning rate η, $\alpha \in [0, 1]$
Input: transitions p, reward function r
Define A2C loss

$$L(\theta) = \mathbb{E}^{\pi_\theta, p}[\sum_{h=1}^{H} -\log \pi_\theta(a_h|s_h)(R(h) - V_\theta(s_h)) + \alpha(R(h) - V_\theta(s_h))^2]$$

$$R(h) = \sum_{h'=h}^{H} r(s_h, a_h)$$

Initialize θ arbitrarily
for Iteration i from 1 to T **do**
    Emulate an episode by running π_θ on MDP with transitions p and reward function r
    update $\theta \leftarrow \theta - \eta \nabla_\theta L(\theta)$
Return π_θ

Table B.1: Considered Hyperparameters

Hyperparameter	Values Considered
A2C learning rate	$10^{-2}, 10^{-3}, 10^{-4}$
lambda learning rate	$10^0, \{1, 2, 5\} \times 10^{-1}, 2 \times 10^{-2}, 10^{-3}, 2 \times 10^{-3}$
reset	warm-start, continue, none
conplanner iterations	$10, 20, 30, 50, 100, 150, 200, 250$
A2C Entropy coeff	10^{-3}
A2C Value loss coeff	0.5

TFW-UCRL2

We used the code provided by the author (with no algorithmic parameter changed). Moreover, TFW-UCRL2 uses weights $(L_0, L_1, \ldots, L_k)$ in the objective function $g(w)$ defined in Equation 1 in Cheung (2019). We only tuned these weights to identify the one maximizing the reward while guaranteeing the constraint satisfaction (for a more fair comparison with the baseline). In our experiments, we have $k = 2$ and you can see the performance of TFW-UCRL2 for $L_0 = 1$ and $L_1 \in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ in Figure B.1.

Table B.2: Selected Hyperparameters

Hyperparameter	Gridworld	Box
A2C learning rate	10^{-3}	10^{-3}
lambda learning rate	2×10^{-1}	10^{-2}
reset	none	none
conplanner iterations	10	10
A2C Entropy coeff	10^{-3}	10^{-3}
A2C Value loss coeff	0.5	0.5

B.6 Concentration tools

This section contains general concentration inequalities that are not tied with the constrained RL setting considered in the chapter.

Lemma B.6.1 (Hoeffding)**.** *Let $\{X_i\}_{i=1}^N$ be a set with each X_i i.i.d sampled from some distribution and $\mathbb{E}[X_i] = 0$ for all i and $\max_i |X_i| \leq b$. Then with probability at least $1-\delta$, it holds that:*

$$\left| \frac{1}{N} \sum_{i=1}^{N} X_i \right| \leq b \sqrt{\frac{2 \ln(2/\delta)}{N}}.$$

Lemma B.6.2 (Anytime version of Hoeffding)**.** *Let $\{X_i\}_{i=1}^\infty$ be a set with each X_i i.i.d sampled from some distribution and $\mathbb{E}[X_i] = 0$ for all i and $\max_i |X_i| \leq b$. Then with probability at least $1-\delta$, for any $N \in \mathbb{N}^+$, it holds that:*

$$\left| \frac{1}{N} \sum_{i=1}^{N} X_i \right| \leq b \sqrt{\frac{2 \ln(4N^2/\delta)}{N}}.$$

Proof. We first fix $N \in \mathbb{N}^+$ and apply standard Hoeffding (Lemma B.6.1) with a failure probability δ/N^2. Then we apply a union bound over $\mathbb{N}^+$ and use the fact that $\sum_{N>0} \frac{\delta}{2N^2} \leq \delta$ to conclude the lemma. □

The following lemma is used when bounding the final regret in the above analysis where we bound the difference between the cumulative Bellman error along the empirical trajectories and the cumulative

Bellman error under the expectation of trajectories (the expectation is taken with respect to the policies generating these trajectories cross episodes).

Lemma B.6.3. *Consider a sequence of episodes $k = 1$ to K, a sequence of policies $\{\pi_k\}_{k=1}^K$, and a sequence of functions $\{f_k\}_{k=1}^K$ with corresponding filtration $\{\mathcal{F}_k\}$ with $\pi_k \in \mathcal{F}_{k-1}$ and $f_k \in \mathcal{F}_{k-1}$. Each policy π_k generates a sequence of trajectory $\{s_{k;h}, s_{k;h}\}_{h=1}^H$. Denote a function $f_k : \mathcal{S} \times \mathcal{A} \to [0, C]$, with $f_k \in \mathcal{F}_{k-1}$. With probability at least $1 - \delta$, for any K, we have:*

$$\left| \sum_{i=1}^{K} \sum_{h=1}^{H} f_k(s_{k;h}, a_{k;h}) - \sum_{k=1}^{K} \mathbb{E}^{\pi_k} \left(\sum_{h=1}^{H} f_k(s(h), a(h)) \right) \right| \leq C \sqrt{2 \ln(4K^2/\delta) KH}.$$

Proof. Denote the random variable $v_{k;h} = f_k\left(s_{k;h}, a_{k;h}\right)$. Denote $\mathbb{E}_{k;h}$ as the conditional expectation that is conditioned on all history from the beginning to time step h (not including step h) at episode k. Note that we have: $\mathbb{E}_{k;h}\left[v_k\right] = \mathbb{E}^{\pi_k}\left(f_k\left(s_{k;h}, a_{k;h}\right)\right)$. Note that $|v_{k;h}| \leq C$ for any k, h by the assumption on f_k. Hence, $\{v_{k;h}\}_{k,h}$ forms a sequence of Martingales. Applying Hoeffding's inequality, we have with probability at least $1 - \delta$,

$$\left| \sum_{k=1}^{K} \sum_{h=1}^{H} v_{k;h} - \sum_{k=1}^{K} \mathbb{E}^{\pi_k} \left(\sum_{h=1}^{H} f_k(s(h), a(h)) \right) \right| \leq C \sqrt{2 \ln(2/\delta) KH} = C \sqrt{2 \ln(2/\delta) HK}.$$

Assigning failure probability δ/k^2 for each episode k and using a union bound over all episodes conclude the proof. □

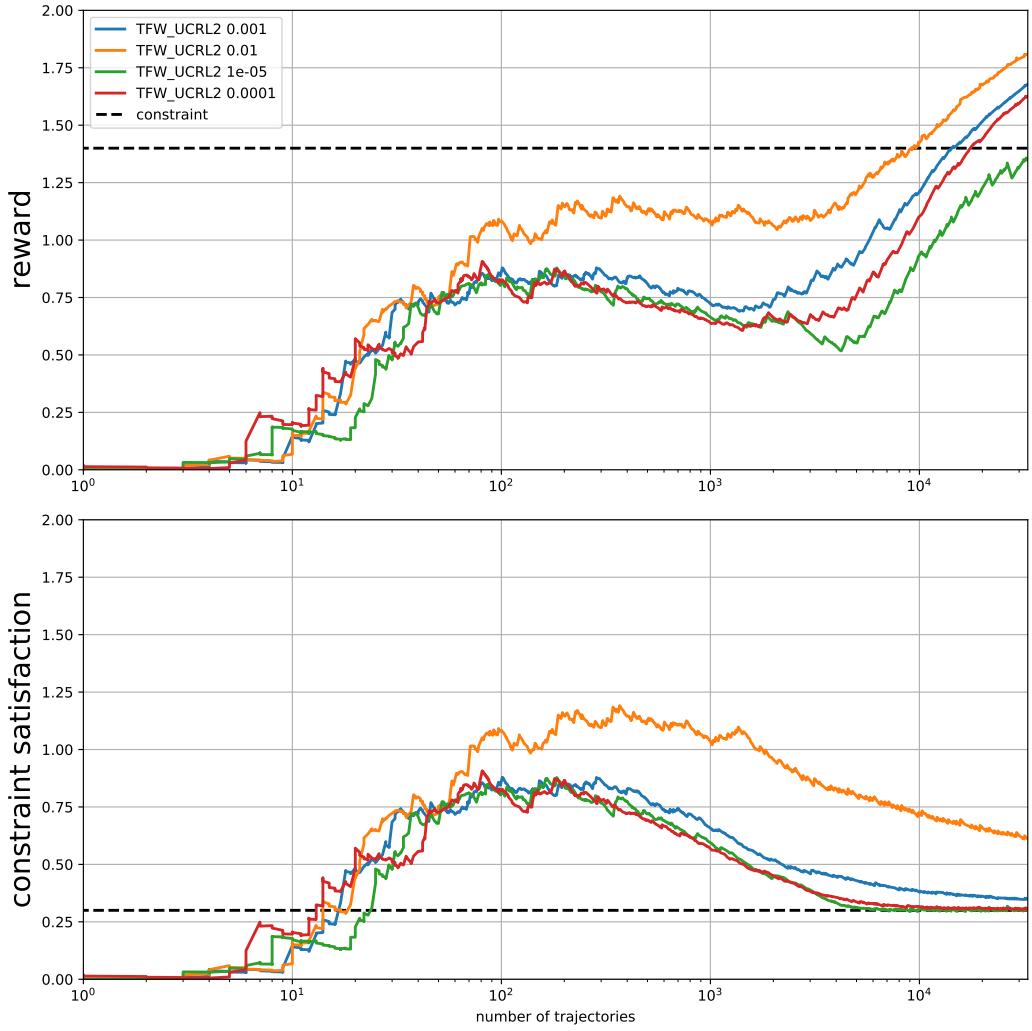


Figure B.1: Performance of TFW-UCRL2 with different choices of L_1 ($L_0 = 1$)

Appendix C

Remaining Proofs of Chapter 4

C.1 Proof for Section 4.2

In this section we provide proofs and missing details for Section 4.2.

C.1.1 Proof of Theorem 4.2.5

Consider the following algorithm which is performing an approximate version of binary search on the optimal cost. We use $\oplus$ to denote vector concatenation.

Theorem C.1.1. *For any choice of approachability algorithm (as in Definiton 4.2.3) and for any $\epsilon, \delta > 0$, if we choose*

$$T = \mathcal{O}\big[\log(H/\epsilon)\big], \quad K_{\text{APP}} = m_{\text{APP}}(\epsilon, \epsilon\delta/(2H)), \quad K_{\text{est}} = \mathcal{O}\big[\frac{H^2 \log(dH/\epsilon\delta)}{\epsilon^2}\big], \quad \epsilon' = \mathcal{O}(\epsilon)$$

Algorithm 12 Solving Constrained RL Using Approachability

Input: approachability algorithm APP
Hyperparameters: $\epsilon' > 0$
Initialize: $L \leftarrow 0$ and $R \leftarrow H$
Define the augmented VMDP model

$$\overline{\mathbf{r}}_h(s,a) = \mathbf{r}_h(s,a) \oplus c_h(s,a) \quad \forall h \in [H]$$
$$\overline{\mathcal{M}} = \{\mathcal{S}, \mathcal{A}, H, \mathbb{P}, \overline{\mathbf{r}}\}$$

for iteration $t = 1, 2, \ldots, T$ **do**
  Set mid $= (R + L)/2$
  Define the target set for approachability

$$\overline{\mathcal{C}}^t = \{\mathbf{x} \oplus y \mid \mathbf{x} \in \mathcal{C}, y \leq \text{mid}\}$$

  $\pi^t \leftarrow$ output of APP algorithm for the model $\overline{\mathcal{M}}$ with target set $\overline{\mathcal{C}}^t$ using K_{APP} episodes.
  $\overline{\mathbf{v}}^t \leftarrow$ estimate $\overline{\mathbf{V}}_1^{\pi^t}(s_1)$ using K_{est} episodes, where $\overline{\mathbf{V}}$ is the value function for $\overline{\mathcal{M}}$.
  if $\text{dist}(\overline{\mathbf{v}}^t, \mathcal{C}) \leq \epsilon'$ **then**
    $R \leftarrow$ mid
  else
    $L \leftarrow$ mid
Return π^T

then, with probability at least $1 - \delta$*, Algorithm 12 satisfies*

$$\begin{cases} C_1^{\pi^T}(s_1) - \min\limits_{\pi: \mathbf{V}_1^\pi(s_1) \in \mathcal{C}} C_1^\pi(s_1) \leq \mathcal{O}(\epsilon), \\ \text{dist}\big(\mathbf{V}_1^{\pi^T}(s_1), \mathcal{C}\big) \leq \mathcal{O}(\epsilon). \end{cases}$$

Proof of Theorem C.1.1. By definition 4.2.3, Lemma C.6.1, and union bound; with probability at least $1 - \delta$, we have for all $t \in [T]$

$$\begin{aligned} &\|\overline{\mathbf{v}}^t - \overline{\mathbf{V}}_1^{\pi^t}(s_1)\| \leq \epsilon, \\ &\text{dist}(\overline{\mathbf{V}}_1^{\pi^t}(s_1), \mathcal{C}) \leq \min_\pi \text{dist}(\overline{\mathbf{V}}_1^\pi(s_1), \mathcal{C}) + \epsilon. \end{aligned} \tag{C.1}$$

We use L^t, R^t, and mid^t to denote values of L, R, and mid during t^{th} iteration. By choice of T we have

$$R^T - L^T \leq \epsilon. \tag{C.2}$$

Define $c^* = \min_{\pi: \mathbf{V}_1^\pi(s_1) \in \mathcal{C}} C_1^\pi(s_1)$ and let $\pi^* = \text{argmin}_{\pi: \mathbf{V}_1^\pi(s_1) \in \mathcal{C}} C_1^\pi(s_1)$. Let's consider these cases

- Case mid $\geq c^*$: It's easy to see that $\min_\pi \operatorname{dist}(\overline{\mathbf{V}}_1^\pi(s_1), \mathcal{C}) = 0$, therefore by second inequality in Equation C.1 we have

$$\operatorname{dist}(\overline{\mathbf{V}}_1^{\pi^t}(s_1), \mathcal{C}) \leq \epsilon.$$

  Since distance function is 1-Lipschitz with respect to Euclidean norm, by first inequality in Equation C.1, we have

$$\operatorname{dist}(\overline{\mathbf{v}}^t, \mathcal{C}) \leq \epsilon + \epsilon = 2\epsilon$$

- Case mid $\leq c^* - 3\epsilon$: It's easy to see that $\min_\pi \operatorname{dist}(\overline{\mathbf{V}}_1^\pi(s_1), \mathcal{C}) \geq 3\epsilon$, therefore by definition of minimum we have

$$\operatorname{dist}(\overline{\mathbf{V}}_1^{\pi^t}(s_1), \mathcal{C}) \geq 3\epsilon.$$

  Since distance function is 1-Lipschitz with respect to Euclidean norm, by first inequality in Equation C.1, we have

$$\operatorname{dist}(\overline{\mathbf{v}}^t, \mathcal{C}) \geq 3\epsilon - \epsilon = 2\epsilon.$$

What we showed above implies that if we set $\epsilon' = 2\epsilon$, in all iterations $t \in [T]$ we have

$$L^t \leq c^*, \quad R^t \geq c^* - 3\epsilon.$$

Combining with Equation C.2, we get

$$c^* - 4\epsilon \leq L^T \leq \text{mid}^T \leq R^T \leq c^* + \epsilon$$

Therefore we have,

$$\begin{aligned}
&\max\{C_1^{\pi^T}(s_1) - \mathrm{mid}^T, \mathrm{dist}\big(\mathbf{V}_1^{\pi^T}(s_1), \mathcal{C}\big)\} \\
&\leq \mathrm{dist}(\overline{\mathbf{V}}_1^{\pi^T}(s_1), \mathcal{C}) \\
&\leq \min_{\pi} \mathrm{dist}(\overline{\mathbf{V}}_1^{\pi}(s_1), \mathcal{C}) + \epsilon \\
&\leq \mathrm{dist}(\overline{\mathbf{V}}_1^{\pi^*}(s_1), \mathcal{C}) + \epsilon \\
&\leq \max\{c^* - \mathrm{mid}^T, 0\} + \epsilon \\
&\leq c^* - (c^* - 4\epsilon) + \epsilon = 5\epsilon
\end{aligned}$$

It implies

$$\begin{cases}
\mathrm{dist}\big(\mathbf{V}_1^{\pi^T}(s_1), \mathcal{C}\big) \leq 5\epsilon \\
C_1^{\pi^T}(s_1) \leq 5\epsilon + \mathrm{mid}^T \leq c^* + 6\epsilon
\end{cases}$$

Rescaling ϵ to $\epsilon/6$ completes the proof. □

Proof of Theorem 4.2.5. Using Theorem C.1.1 the claim follows immediately: total sample complexity of Algorithm 12 is

$$T(K_{\mathrm{APP}} + K_{\mathrm{est}}) \leq \log(1/\epsilon) \cdot \mathcal{O}\Big(m_{\mathrm{APP}}(\epsilon, \epsilon\delta/H) + \frac{H^2 \log[d/\epsilon\delta]}{\epsilon^2}\Big).$$

□

C.2 Proof for Section 4.3

In this section we provide proofs and missing details for Section 4.3.

C.2.1 Fenchel duality

Consider a convex and closed function $f : \mathrm{dom}(f) \to \mathbb{R}$. We define the dual function f^*, called *Fenchel conjugate*, as

$$f^*(\theta) = \max_{\mathbf{x} \in \mathrm{dom}(f)} \Big[\langle \theta, \mathbf{x} \rangle - f(\mathbf{x})\Big].$$

If function f is 1-Lipschitz and $\mathrm{dom}(f) = \mathcal{B}(H)$; then, the conjugate function f^* is H-Lipschitz with $\mathrm{dom}(f^*) = \mathcal{B}(1)$ (Corollary 13.3.3 in Rockafellar 2015). Therefore, Fenchel daulity implies

$$f(\mathbf{x}) = \max_{\theta \in \mathcal{B}(1)} \Big[\langle \theta, \mathbf{x} \rangle - f^*(\theta) \Big].$$

In particular, for closed, convex, and 1-Lipschitz function f defined as

$$\begin{cases} f : \mathcal{B}(H) \to \mathbb{R} \\ f(\mathbf{x}) = \mathrm{dist}(\mathbf{x}, \mathcal{C}) \end{cases}$$

we have

$$f^*(\theta) = \max_{\mathbf{x} \in \mathcal{C}} \langle \theta, \mathbf{x} \rangle.$$

It's easy to verify that $\partial f^*(\theta) = \mathrm{argmax}_{\mathbf{x} \in \mathcal{C}} \langle \theta, \mathbf{x} \rangle$ is a subgradient of f^* at θ. Fenchel duality implies that

$$\mathrm{dist}(x, \mathcal{C}) = \max_{\theta \in \mathcal{B}(1)} \Big[\langle \theta, \mathbf{x} \rangle - \max_{\mathbf{x}' \in \mathcal{C}} \langle \theta, \mathbf{x}' \rangle \Big]. \tag{C.3}$$

C.2.2 Online Convex Optimization (OCO)

We will be using the guarantee of online gradient ascent algorithm (Zinkevich, 2003) in the proof. Therefore, we briefly review the framework of online convex optimization. We can imagine an online game between the leaner and the environment: The learner is given a decision set Θ; at time $t = 1, 2, \ldots, T$, the leaner makes a decision $\theta^t \in \Theta$, the environment reveals a concave utility function $u^t : \Theta \to \mathbb{R}$, and the learner gains utility $u^t(\theta^t)$. The learner's goal is to minimize *regret* defined as

$$\mathrm{Regret}_T \triangleq \max_{\theta \in \Theta} \big[\sum_{t=1}^{T} u^t(\theta) \big] - \big[\sum_{t=1}^{T} u^t(\theta^t) \big].$$

An OCO algorithm is *no-regret* if $\mathrm{Regret}_T = o(T)$, meaning its average utility approaches to best in hindsight. The *online gradient ascent* (OGA) is an example of such algorithm (Algorithm 13). In Theorem C.2.1 we formally state the theoretical guarantee of this algorithm.

Algorithm 13 Online gradient ascent (OGA)

input: projection operator Γ_Θ where $\Gamma_\Theta(\theta) = \operatorname{argmin}_{\theta\in\Theta}\|\theta - \theta'\|$
init: θ^1 arbitrarily
parameters: step size η_t
for $t = 1$ **to** T **do**
  observe concave utility function $u^t : \Theta \to \mathbb{R}$
  $\theta^{t+1} = \Gamma_\Theta\big(\theta^t + \eta_t \partial u^t(\theta^t)\big)$ {where $\partial u^t(\theta^t)$ is a subgradient of u^t at θ^t}

Theorem C.2.1 (Zinkevich 2003)**.** *Assume that for any $\theta, \theta' \in \Theta$ we have $\|\theta - \theta'\| \le D$ and $u^1, \dots, u^T$ are concave and G-Lipschitz. By setting $\eta_t = \frac{D}{G\sqrt{t}}$, Algorithm 13 satisfies*

$$\text{Regret}_T \le \mathcal{O}(DG\sqrt{T}).$$

C.2.3 Proof of Theorem 4.3.1

We use the following choice for parameters:

$$K \ge m_{\text{RFE}}(\epsilon/2, \delta/2), \quad T \ge c \cdot (H^2\iota/\epsilon^2). \tag{C.4}$$

We denote $\mathbf{v}^t := \mathbf{V}_1^{\pi^t}(s_1)$ and start with the following lemma.

Lemma C.2.2. *Define even E_0 to be:*

$$\begin{cases} \|\frac{1}{T}\sum_{t=1}^T \mathbf{v}^t - \widehat{\mathbf{v}}^t\| \le \mathcal{O}(\sqrt{H^2\iota/T}), \\ \mathbf{V}_1^*(s_1; -\theta^t) \le \mathbf{V}_1^{\pi^t}(s_1; -\theta^t) + \epsilon/2 \quad \forall t \in [T]. \end{cases}$$

where $\iota = \log(d/\delta)$. We have $\mathbb{P}(E_0) \ge 1 - \delta$.

Proof of Lemma C.2.2. We show that each claim holds with probability at least $1 - \delta/2$; applying a union bound completes the proof.

First claim. Let $\mathcal{F}_t$ be the filtration capturing all the randomness in the algorithm before iteration t. We have $\mathbb{E}[\widehat{\mathbf{v}}^t \mid \mathcal{F}_t] = \mathbf{v}^t$ and we also know that $\|\widehat{\mathbf{v}}^t\| \leq H$ almost surely. By applying Lemma C.6.1, with probability at least $1-\delta$ we have

$$\|\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t - \widehat{\mathbf{v}}^t\| \leq \mathcal{O}(\sqrt{H^2\log[d/\delta]/T}),$$

which completes the proof.

Second claim. Choice of parameters in Equation C.4 along with Definition 4.2.1 immediately implies that with probability at least $1-\delta/2$ we have

$$\mathbf{V}_1^*(s_1;-\theta^t) \leq \mathbf{V}_1^{\pi^t}(s_1;-\theta^t) + \epsilon/2 \quad \forall t \in [T].$$

Note that in Algorithm 3, π^t is the output of the planning phase of the RFE algorithm for the vector $-\theta^t$ as input. □

The following lemma states that if $\alpha = \min_\pi \text{dist}(\mathbf{V}_1^\pi(s_1), \mathcal{C}) \geq 0$ is the closest achievable distance to target set $\mathcal{C}$, then any halfspace containing $\mathcal{C}$ is reachable up to error α.

Lemma C.2.3. *For any $\theta \in \mathcal{B}(1)$, we have*

$$\min_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle \leq \min_\pi \text{dist}(\mathbf{V}_1^\pi(s_1),\mathcal{C}) + \mathbf{V}_1^*(s_1;\theta).$$

Proof of Lemma C.2.3. Let $\overline{\pi} = \text{argmin}_\pi \text{dist}(\mathbf{V}_1^\pi(s_1),\mathcal{C})$ and define $\overline{\mathbf{v}} = \mathbf{V}_1^{\overline{\pi}}(s_1)$. Let $\tilde{\mathbf{v}} = \Gamma_{\mathcal{C}}(\overline{\mathbf{v}})$ be

the orthogonal projection of $\overline{\mathbf{v}}$ into $\mathcal{C}$. We have

$$\begin{aligned}
\mathbf{V}_1^*(s_1;\theta) &\geq \mathbf{V}_1^{\overline{\pi}}(s_1;\theta) \\
&= \langle \theta, \overline{\mathbf{v}} \rangle \\
&= \langle \theta, \overline{\mathbf{v}} - \tilde{\mathbf{v}} \rangle + \langle \theta, \tilde{\mathbf{v}} \rangle \\
&\geq -\|\overline{\mathbf{v}} - \tilde{\mathbf{v}}\| + \min_{\mathbf{x} \in \mathcal{C}} \langle \theta, \mathbf{x} \rangle \\
&\geq -\min_{\pi} \operatorname{dist}(\mathbf{V}_1^{\pi}(s_1), \mathcal{C}) + \min_{\mathbf{x} \in \mathcal{C}} \langle \theta, \mathbf{x} \rangle
\end{aligned}$$

□

Now we are ready to proceed with proof of Theorem 4.3.1.

Proof of Theorem 4.3.1. With probability at least $1-\delta$ event E_0 holds and we have

$$\begin{aligned}
\operatorname{dist}(\mathbf{V}_1^{\pi^{\text{out}}}(s_1),\mathcal{C}) &= \operatorname{dist}\Big(\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t,\mathcal{C}\Big)\\
&\overset{(i)}{=} \max_{\theta\in\mathcal{B}(1)}\Big[\langle\theta,\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t)\rangle - \max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle\Big]\\
&= \max_{\theta\in\mathcal{B}(1)}\Big[\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta,\widehat{\mathbf{v}}^t\rangle - \max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle\big) + \langle\theta,\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t-\widehat{\mathbf{v}}^t\rangle\Big]\\
&\overset{(ii)}{\leq} \max_{\theta\in\mathcal{B}(1)}\Big[\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta,\widehat{\mathbf{v}}^t\rangle - \max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle\big)\Big] + \mathcal{O}(\sqrt{H^2\iota/T})\\
&\overset{(iii)}{\leq} \frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta^t,\widehat{\mathbf{v}}^t\rangle - \max_{\mathbf{x}\in\mathcal{C}}\langle\theta^t,\mathbf{x}\rangle\big) + \mathcal{O}(\sqrt{H^2/T}) + \mathcal{O}(\sqrt{H^2\iota/T})\\
&\overset{(iv)}{\leq} \min_{\pi}\operatorname{dist}(\mathbf{V}_1^{\pi}(s_1),\mathcal{C}) + \frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta^t,\widehat{\mathbf{v}}^t\rangle + \mathbf{V}_1^{*}(s_1;-\theta^t)\big) + \mathcal{O}(\sqrt{H^2\iota/T})\\
&\overset{(v)}{\leq} \min_{\pi}\operatorname{dist}(\mathbf{V}_1^{\pi}(s_1),\mathcal{C}) + \epsilon/2 + \frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta^t,\widehat{\mathbf{v}}^t\rangle + \mathbf{V}_1^{\pi^t}(s_1;-\theta^t)\big) + \mathcal{O}(\sqrt{H^2\iota/T})\\
&= \min_{\pi}\operatorname{dist}(\mathbf{V}_1^{\pi}(s_1),\mathcal{C}) + \epsilon/2 + \frac{1}{T}\sum_{t=1}^{T}\langle\theta^t,\widehat{\mathbf{v}}^t-\mathbf{v}^t\rangle + \mathcal{O}(\sqrt{H^2\iota/T})\\
&\overset{(vi)}{\leq} \min_{\pi}\operatorname{dist}(\mathbf{V}_1^{\pi}(s_1),\mathcal{C}) + \epsilon/2 + \mathcal{O}(\sqrt{H^2\iota/T})\\
&\overset{(vii)}{\leq} \min_{\pi}\operatorname{dist}(\mathbf{V}_1^{\pi}(s_1),\mathcal{C}) + \epsilon
\end{aligned}$$

where (i) is by Equation C.3, (ii) is by first inequality in event E_0 together with Cauchy-Schwarz, (iii) is by guarantee of OGA in Theorem C.2.1, (iv) is by Lemma C.2.3, (v) is by second inequality in event E_0, (vi) is by first inequality in event E_0 together with Cauchy-Schwarz, and finally (vii) is by setting $T \geq c\big(H^2\iota/\epsilon^2\big)$ for large enough constant c, completing the proof. □

C.3 Proof for Section 4.4

In this section we provide proofs and missing details for Section 4.4.

Algorithm 14 VI-Zero: Exploration Phase

Hyperparameters: Bonus β_t.
Initialize: for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$: $\widetilde{Q}_h(s,a) \leftarrow H$ and $N_h(s,a) \leftarrow 0$,
  for all $(s,a,h,s') \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{S}$: $N_h(s,a,s') \leftarrow 0$,
  $\Delta \leftarrow 0$.
for episode $k = 1, 2, \ldots, K$ **do**
  for step $h = H, H-1, \ldots, 1$ **do**
    for state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
      $t \leftarrow N_h(s,a)$.
      if $t > 0$ **then**
        $\widetilde{Q}_h(s,a) \leftarrow \min\{[\widehat{\mathbb{P}}_h \widetilde{V}_{h+1}](s,a) + \beta_t, H\}$.
    for state $s \in \mathcal{S}$ **do**
      $\widetilde{V}_h(s) \leftarrow \max_{a \in \mathcal{A}} \widetilde{Q}_h(s,a)$ and $\pi_h(s) \leftarrow \operatorname{argmax}_{a \in \mathcal{A}} \widetilde{Q}_h(s,a)$
  if $\widetilde{V}(s_1) \le \Delta$ **then**
    $\Delta \leftarrow \widetilde{V}(s_1)$ and $\widehat{\mathbb{P}}^{\text{out}} \leftarrow \widehat{\mathbb{P}}_h$
  for step $h = 1, 2, \ldots, H$ **do**
    Take action $a_h \leftarrow \pi_h(s_h)$ and observe next state s_{h+1}
    Update $N_h(s_h, a_h) \leftarrow N_h(s_h, a_h) + 1$ and $N_h(s_h, a_h, s_{h+1}) \leftarrow N_h(s_h, a_h, s_{h+1}) + 1$
    $\widehat{\mathbb{P}}_h(\cdot \mid s_h, a_h) \leftarrow N_h(s_h, a_h, \cdot)/N_h(s_h, a_h)$
Return $\widehat{\mathbb{P}}^{\text{out}}$

C.3.1 Reward-free Algorithm for Tabular VMDPs

In the exploration phase, we use VI-Zero (Liu et al., 2020) with modified choice of hyperparameters. The pseudocode is displayed in Algorithm 14. Intuitively, the value function $\widetilde{Q}_h(s,a)$ computed in the algorithm measures the level of uncertainty that agent may suffer if it takes action a at state s in step h. It incentivize the greedy policy to visit underexplored states improving our empirical estimate $\widehat{\mathbb{P}}$.

In the planning phase, given $\theta \in \mathcal{B}(1)$ as input we can use any planning algorithm (such as value iteration) for $\widehat{\mathcal{M}}_\theta = (\mathcal{S}, \mathcal{A}, H, \widehat{\mathbb{P}}^{\text{out}}, \langle \theta, \widehat{\mathbf{r}} \rangle)$ where $\widehat{\mathbf{r}}$ is empirical estimate of $\mathbf{r}$ using collected samples $\{\mathbf{r}_h^k\}$.

C.3.2 Proof of Theorem 4.4.1

In this section, we prove Theorem C.3.1 which implies the first claim in Theorem 4.4.1. Second and third claims in Theorem 4.4.1 immediately follow due to Theorem 4.3.1 and Theorem 4.2.5.

Let $\widehat{\mathbb{P}}^k$ and $\widehat{\mathbf{r}}^k$ be our empirical estimates of the transition and the return vectors at the beginning of the k^{th} episode in Algorithm 14 and define $\widehat{\mathcal{M}}^k = (\mathcal{S}, \mathcal{A}, H, \widehat{\mathbb{P}}^k, \widehat{\mathbf{r}}^k)$. We use $N_h^k(s,a)$ to denote the number of times we have visited state-action (s,a) in step h before k^{th} episode in Algorithm 14. We use superscript k to denote variable corresponding to episode k; in particular, $(s_1^k, a_1^k, \ldots, s_H^k, a_H^k)$ is the trajectory we have visited in the k^{th} episode.

For any $\theta \in \mathcal{B}(1)$, let $\widehat{\mathcal{M}}_\theta^k$ be the scalarized MDP using vector θ (defined in Section 4.2). We use $\widehat{V}^k(\cdot;\theta)$, $\widehat{Q}^k(\cdot,\cdot;\theta)$, and $\widehat{\pi}_\theta^k = \widehat{\pi}^k(\cdot;\theta)$ to denote the optimal value function, optimal Q-value function, and optimal policy of $\widehat{\mathcal{M}}_\theta^k$ respectively. Therefore, we have

$$\begin{aligned}
\widehat{Q}_h^k(s,a;\theta) &= [\widehat{\mathbb{P}}_h^k \widehat{V}_{h+1}^k](s,a;\theta) + \widehat{r}_h^k(s,a;\theta), \\
\widehat{V}_h^k(s;\theta) &= \max_{a\in\mathcal{A}} \widehat{Q}_h^k(s,a;\theta), \\
\widehat{\pi}_h^k(s;\theta) &= \operatorname{argmax}_{a\in\mathcal{A}} \widehat{Q}_h^k(s,a;\theta).
\end{aligned} \tag{C.5}$$

Theorem C.3.1. *There exist absolute constants c_β and c_K, such that for any $\epsilon \in (0,H]$, $\delta \in (0,1]$, if we choose bonus $\beta_t = c_\beta\big(\sqrt{\min\{d,S\}H^2\iota/t} + H^2S\iota/t\big)$ where $\iota = \log[dSAKH/\delta]$, and run the exploration phase (Algorithm 14) for $K \geq c_K\big(\min\{d,S\}H^4SA\iota'/\epsilon^2 + H^3S^2A(\iota')^2/\epsilon\big)$ episodes where $\iota' = \log[dSAH/(\epsilon\delta)]$, then with probability at least $1-\delta$, the algorithm satisfies*

$$\forall \theta \in \mathcal{B}(1): \quad V_1^\star(s_1;\theta) - V_1^{\pi_\theta}(s_1;\theta) \leq \epsilon,$$

where π_θ is the output of the any planning algorithm (e.g., value iteration) for the MDP $\widehat{\mathcal{M}}_\theta^{\text{out}}$. Therefore, we have

$$m_{\text{RFE}}(\epsilon,\delta) \leq \mathcal{O}\Big(\frac{\min\{d,S\}H^4SA\iota'}{\epsilon^2} + \frac{H^3S^2A(\iota')^2}{\epsilon}\Big).$$

The bonus for episode k can be written as

$$\beta_h^k(s,a) = c_\beta\Big(\sqrt{\frac{\min\{d,S\}H^2\iota}{\max\{N_h^k(s,a),1\}}} + \frac{H^2S\iota}{\max\{N_h^k(s,a),1\}}\Big), \tag{C.6}$$

where $\iota = \log[dSAKH/\delta]$ and c_β is some large absolute constant.

We begin with the following lemma showing that the value function for a fixed π and also the optimal value function is H-Lipschitz with respect to θ.

Lemma C.3.2. *For all $(s,h) \in \mathcal{S} \times [H]$, for all policies π, and for any two vectors $\theta, \theta' \in \mathcal{B}(1)$, we have*

$$|V_h^\star(s;\theta) - V_h^\star(s;\theta')| \le (H-h+1)\|\theta - \theta'\|$$
$$|V_h^\pi(s;\theta) - V_h^\pi(s;\theta')| \le (H-h+1)\|\theta - \theta'\|$$

Proof of Lemma C.3.2. We prove each claim separately.

First claim. We prove the lemma by backward induction on h. For $h = H+1$ we have $V_h^\star(s;\theta) = V_h^\star(s;\theta') = 0$ and the inequality holds. Now assume that $|V_{h+1}^\star(s;\theta) - V_{h+1}^\star(s;\theta')| \le (H-h)\|\theta - \theta'\|$ holds, we want to show that the claim also holds for h. We have

$$\begin{aligned}
|V_h^\star(s;\theta) - V_h^\star(s;\theta')| &= |\max_{a\in\mathcal{A}} Q_h^\star(s,a;\theta) - \max_{a'\in\mathcal{A}} Q^\star(s,a';\theta')| \\
&\le \max_{a\in\mathcal{A}} |Q_h^\star(s,a;\theta) - Q_h^\star(s,a;\theta')| \\
&= \max_{a\in\mathcal{A}} |\langle \theta - \theta', \mathbf{r}_h(s,a)\rangle + \sum_{s'\in\mathcal{S}} \mathbb{P}(s' \mid s,a)(V_{h+1}^\star(s';\theta) - V_{h+1}^\star(s';\theta'))| \\
&\le \max_{a\in\mathcal{A}} ||\langle \theta - \theta', \mathbf{r}_h(s,a)\rangle| + \max_{a\in\mathcal{A}} |\sum_{s'\in\mathcal{S}} \mathbb{P}(s' \mid s,a)(V_{h+1}^\star(s';\theta) - V_{h+1}^\star(s';\theta'))| \\
&\le \|\theta - \theta'\| + (H-h)\|\theta - \theta'\| \\
&= (H-h+1)\|\theta - \theta'\|.
\end{aligned}$$

It completes the proof of the lemma.

Second claim. The second claim is much easier to prove, since we have

$$\begin{aligned}
|V_h^\pi(s;\theta) - V_h^\pi(s;\theta')| &= \Big|\mathbb{E}_\pi\big[\sum_{h'=1}^{H}\langle\theta-\theta', \mathbf{r}_h(s'_h,a'_h)\rangle\big]\Big| \\
&\le \mathbb{E}_\pi\big[\sum_{h'=1}^{H}|\langle\theta-\theta', \mathbf{r}_h(s'_h,a'_h)\rangle|\big] \\
&\le E_\pi\big[\sum_{h'=1}^{H}\|\theta-\theta'\|\big] \\
&= (H-h+1)\|\theta-\theta'\|
\end{aligned}$$

where the first inequality uses Jensen, and second inequality uses Cauchy-Schwarz. □

Lemma C.3.3. *Let c be some large absolute constant such that $2c+12c^2 \le c_\beta$. Define event E_1 to be: for all $(s,a,s',h) \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}\times[H]$, $k\in[K]$, and $\theta\in\mathcal{B}(1)$,*

$$\begin{cases}
|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| & \le c\sqrt{\frac{\min\{d,S\}H^2\iota}{\max\{N_h^k(s,a),1\}}}, \\
|(\widehat{r}_h^k - r_h)(s,a;\theta)| & \le c\sqrt{\frac{\iota}{\max\{N_h^k(s,a),1\}}}, \\
|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(s' \mid s,a)| & \le c\Big(\sqrt{\frac{\widehat{\mathbb{P}}_h^k(s'|s,a)\iota}{\max\{N_h^k(s,a),1\}}} + \frac{\iota}{\max\{N_h^k(s,a),1\}}\Big),
\end{cases} \tag{C.7}$$

where $\iota = \log[dSAKH/\delta]$. We have $\mathbb{P}(E_1)\ge 1-\delta$.

Proof of Lemma C.3.3. The proof is by applying concentration and covering arguments together with union bounds. The following shows that each claim holds with probability at least $1-\delta$; rescaling δ to $\delta/3$ and applying a union bound completes the proof.

First claim: For a fixed $(s,a,k,h,\theta)\in\mathcal{S}\times\mathcal{A}\times[K]\times[H]\times\mathcal{B}(1)$, using Azuma-Hoeffding inequality, with probability at least $1-\delta'$ we have

$$|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| \le \mathcal{O}\Big(\sqrt{\frac{H^2\log(1/\delta')}{N_h^k(s,a)}}\Big).$$

Now consider an ϵ'-covering $\mathcal{B}_{\epsilon'}$ for the unit Euclidean ball $\mathcal{B}(1)$ with $\log|\mathcal{B}_{\epsilon'}| \leq \mathcal{O}(d\log(1/\epsilon'))$. For any $\theta \in \mathcal{B}(1)$, there exists $\theta' \in \mathcal{B}_{\epsilon'}$ satisfying $\|\theta - \theta\| \leq \epsilon'$. The concentration inequality above along with a union bound implies that with probability at least $1-\delta$ for any $(s,a,k,h,\theta') \in \mathcal{S}\times\mathcal{A}\times[K]\times[H]\times\mathcal{B}_{\epsilon'}$ we have

$$|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta')| \leq \mathcal{O}\Big(\sqrt{\frac{dH^2}{N_h^k(s,a)}\log(\frac{SAKH}{\epsilon'\delta})}\Big).$$

Now consider an arbitrary $(s,a,k,h,\theta) \in \mathcal{S}\times\mathcal{A}\times[K]\times[H]\times\mathcal{B}(1)$. Let $\theta' \in \mathcal{B}_{\epsilon'}$ be such that $\|\theta-\theta'\| \leq \epsilon'$; we have

$$\begin{aligned}
&|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| \\
&\overset{(i)}{\leq} |[\widehat{\mathbb{P}}_h^k(V_{h+1}^\star(\cdot;\theta) - V_{h+1}^\star(\cdot;\theta'))](s,a)]| + |[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta')| \\
&+ |[\mathbb{P}_h(V_{h+1}^\star(\cdot;\theta') - V_{h+1}^\star(\cdot;\theta))](s,a)]| \\
&\overset{(ii)}{\leq} 2H\|\theta-\theta'\| + \mathcal{O}\Big(\sqrt{\frac{dH^2}{N_h^k(s,a)}\log(\frac{SAKH}{\epsilon'\delta})}\Big) \\
&\leq 2H\epsilon' + \mathcal{O}\Big(\sqrt{\frac{dH^2}{N_h^k(s,a)}\log(\frac{SAKH}{\epsilon'\delta})}\Big),
\end{aligned}$$

where (i) is by adding and subtracting the term $[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta')$ along with triangle inequality, and (ii) is by Lemma C.3.2. Setting $\epsilon' = \frac{1}{HN_h^k(s,a)} \geq \frac{1}{HK}$ results in

$$[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| \leq \mathcal{O}\Big(\sqrt{\frac{dH^2}{N_h^k(s,a)}\log(\frac{SAKH}{\delta})}\Big).$$

On the other hand, consider an ϵ'-cover $\mathcal{V}_{\epsilon'}$ for the ℓ_∞ ball of radius H in dimension S, i.e. $\{\mathbf{v}\in\mathbb{R}^S \mid \|\mathbf{v}\|_\infty \leq H\}$. For a fixed $(s,a,k,h,\mathbf{v}) \in \mathcal{S}\times\mathcal{A}\times[K]\times[H]\times\mathcal{V}_{\epsilon'}$, using Azuma-Hoeffding inequality, with probability at least $1-\delta'$ we have

$$|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)\mathbf{v}](s,a)| \leq \mathcal{O}\Big(\sqrt{\frac{H^2\log(1/\delta')}{N_h^k(s,a)}}\Big).$$

Note that $|\mathcal{V}_{\epsilon'}| \leq (3H/\epsilon')^d$, therefore by putting $\delta' = \delta/(SAKH|\mathcal{V}_{\epsilon'}|)$ we get for all $(s, a, k, h, \mathbf{v}) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H] \times \mathcal{V}_{\epsilon'}$

$$|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)\mathbf{v}](s,a)| \leq \mathcal{O}\Big(\sqrt{\frac{SH^2 \log(SAKH/(\epsilon'\delta))}{N_h^k(s,a)}}\Big).$$

Now consider an arbitrary $(s, a, k, h, \theta) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H] \times \mathcal{B}(1)$, and let $\mathbf{v} \in \mathcal{V}_{\epsilon'}$ be such that $\|V_{h+1}^\star(\cdot;\theta) - \mathbf{v}\|_\infty \leq \epsilon'$. We have

$$\begin{aligned}
|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| &\leq |[\widehat{\mathbb{P}}_h^k(V_{h+1}^\star(\cdot;\theta) - \mathbf{v})](s,a)| + |[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)\mathbf{v}](s,a)| \\
&\quad + |[\mathbb{P}_h(V_{h+1}^\star(\cdot;\theta) - \mathbf{v})](s,a)| \\
&\leq 2\epsilon' + \mathcal{O}\Big(\sqrt{\frac{SH^2 \log(SAKH/(\epsilon'\delta))}{N_h^k(s,a)}}\Big).
\end{aligned}$$

Setting $\epsilon' = \frac{1}{N_h^k(s,a)} \geq \frac{1}{K}$ results in

$$|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| \leq \mathcal{O}\Big(\sqrt{\frac{SH^2}{N_h^k(s,a)} \log(\frac{SAKH}{\delta})}\Big)$$

The two bounds together complete the proof for the first claim.

Second claim: We have $\|\mathbf{r}_h^k\| \leq 1$ almost surely and $\mathbb{E}[\mathbf{r}_h^k \mid \mathcal{F}_h^k] = \mathbf{r}_h(s_h^k, a_h^k)$. For a fixed $(s, a, k, h) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H]$, applying Lemma C.6.1 implies that with probability at least $1 - \delta'$ we have

$$\|(\widehat{\mathbf{r}}_h^k - \mathbf{r}_h)(s,a)\| \leq \mathcal{O}\Big(\sqrt{\frac{\log(d/\delta')}{N_h^k(s,a)}}\Big).$$

Setting $\delta' = \delta/(SAKH)$ and applying a union bound, for all $(s, a, k, h) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H]$, we have

$$\|(\widehat{\mathbf{r}}_h^k - \mathbf{r}_h)(s,a)\| \leq \mathcal{O}\Big(\sqrt{\frac{\log(dSAKH/\delta)}{N_h^k(s,a)}}\Big).$$

Now consider an arbitrary $(s, a, k, h, \theta) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H] \times \mathcal{B}(1)$, we have (by Cauchy-Schwarz)

$$\begin{aligned}|(\widehat{r}_h^k - r_h)(s, a; \theta)| &= |\langle \theta, (\widehat{\mathbf{r}}_h^k - \mathbf{r}_h)(s, a)| \\ &\leq \|\theta\| \|(\widehat{\mathbf{r}}_h^k - \mathbf{r}_h)(s, a)\| \\ &\leq \mathcal{O}\Big(\sqrt{\frac{\log(dSAKH/\delta)}{N_h^k(s, a)}}\Big),\end{aligned}$$

completing proof of this claim.

Third claim: For a fixed $(s, a, s', k, h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times [K] \times [H]$, using empirical Bernstein inequality, with probability at least $1 - \delta'$ we have

$$|(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(s' \mid s, a)| \leq \mathcal{O}\Big(\sqrt{\frac{\widehat{\mathbb{P}}_h^k(s' \mid s, a) \log(1/\delta')}{N_h^k(s, a)}} + \frac{\log(1/\delta')}{N_h^k(s, a)}\Big)$$

Applying a union bound and setting $\delta' = \delta / S^2 AKH$ completes the proof. □

The following lemma shows that the optimal value functions of $\widehat{\mathcal{M}}_\theta^k$ are close to the optimal value functions of $\mathcal{M}_\theta$ and their difference is controlled by $\widetilde{Q}$ and $\widetilde{V}$ computed in Algorithm 14.

Lemma C.3.4. *Suppose event E_1 holds (defined in Lemma C.3.3); then, for all $(s, a, k, h, \theta) \in \mathcal{S} \times \mathcal{A} \times [K] \times [H] \times \mathcal{B}(1)$ we have*

$$\begin{aligned}|\widehat{Q}_h^k(s, a; \theta) - Q_h^\star(s, a; \theta)| &\leq \widetilde{Q}_h^k(s, a), \\ |\widehat{V}_h^k(s; \theta) - V_h^\star(s; \theta)| &\leq \widetilde{V}_h^k(s).\end{aligned} \tag{C.8}$$

Proof of Lemma C.3.4. We prove the lemma by backward induction on h. For $h = H + 1$ the claim holds trivially. Now suppose that the claim is true for $(h + 1)^{\text{th}}$ step, we want to show that the

claim is also true for h^{th} step. For the Q-value function we have

$$
\begin{aligned}
&|\widehat{Q}_h^k(s,a;\theta) - Q_h^\star(s,a;\theta)| \\
&\le \min\Big\{ \underbrace{|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| + |(\widehat{r}_h^k - r_h)(s,a;\theta)|}_{(T_1)} + \underbrace{|[\widehat{\mathbb{P}}_h^k(\widehat{V}_{h+1}^k - V_{h+1}^\star)](s,a;\theta)|}_{(T_2)}, H \Big\} \\
&\overset{(i)}{\le} \min\Big\{ \beta_h^k(s,a) + [\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a), H \Big\} \overset{(ii)}{=} \widetilde{Q}_h^k(s,a),
\end{aligned}
$$

where (i) follows from $T_1 \le \beta_h^k(s,a)$ (event E_1) and $T_2 \le [\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a)$ (induction hypothesis), and (ii) is due to definition of $\widetilde{Q}_h^k$ in Algorithm 14. Now for the value function we have

$$
\begin{aligned}
&|\widehat{V}_h^k(s;\theta) - V_h^\star(s;\theta)| \\
&= |\max_{a\in\mathcal{A}} \widehat{Q}_h^k(s,a;\theta) - \max_{a'\in\mathcal{A}} \widehat{Q}^\star(s,a';\theta)| \\
&\le \max_{a\in\mathcal{A}} |\widehat{Q}_h^k(s,a;\theta) - \widehat{Q}^\star(s,a;\theta)| \\
&\le \max_{a\in\mathcal{A}} \widetilde{Q}_h^k(s,a) = \widetilde{V}_h^k(s),
\end{aligned}
$$

which completes the induction step and consequently the proof. □

Now we are ready to introduce the main lemma that shows value of $\widehat{\pi}_\theta^k$ under the true model is close to its value under empirical model. The difference is controlled by $\widetilde{Q}$ and $\widetilde{V}$ computed in Algorithm 14.

Lemma C.3.5. *Suppose event E_1 holds (defined in Lemma C.3.3); then, for all $(s,a,k,h,\theta) \in \mathcal{S}\times\mathcal{A}\times[K]\times[H]\times\mathcal{B}(1)$ we have*

$$
\begin{aligned}
|\widehat{Q}_h^k(s,a;\theta) - Q_h^{\widehat{\pi}_\theta^k}(s,a;\theta)| &\le \alpha_h \widetilde{Q}_h^k(s,a), \\
|\widehat{V}_h^k(s;\theta) - V_h^{\widehat{\pi}_\theta^k}(s;\theta)| &\le \alpha_h \widetilde{V}_h^k(s),
\end{aligned}
\tag{C.9}
$$

where $\alpha_{H+1} = 1$ and $\alpha_h = [(1+\frac{1}{H})\alpha_{h+1} + \frac{1}{H}]$; we have $1 \le \alpha_h \le 5$ for $h \in [H]$.

Proof of Lemma C.3.5. We prove the claim by backward induction on h. For $h = H+1$ the claim

trivially holds. Now suppose that the claim is true for step $h+1$ and we want to show that it also holds for step h.

$$\begin{aligned}
&|\widehat{Q}_h^k(s,a;\theta) - Q_h^{\widehat{\pi}_\theta^k}(s,a;\theta)| \\
&\leq \min\Big\{ \underbrace{|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(V_{h+1}^{\widehat{\pi}_\theta^k} - V_{h+1}^\star)](s,a;\theta)|}_{(T_1)} \\
&+ \underbrace{|[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a;\theta)| + |(\widehat{r}_h^k - r_h)(s,a;\theta)|}_{(T_2)} \\
&+ \underbrace{|[\widehat{\mathbb{P}}_h^k(\widehat{V}_{h+1}^k - V_{h+1}^{\widehat{\pi}_\theta^k})](s,a;\theta)|}_{(T_3)}, H\Big\}
\end{aligned} \tag{C.10}$$

For the term (T_3), by applying induction hypothesis we have

$$(T_3) \leq \alpha_{h+1}[\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a). \tag{C.11}$$

Using event E_1, for the term (T_2) we have

$$(T_2) \leq 2c\sqrt{\frac{\min\{d,S\}H^2\iota}{\max\{N_h^k(s,a),1\}}}. \tag{C.12}$$

It only remains to bound the term (T_1); we have

$$\begin{aligned}
(T_1) &\leq \sum_{s'\in\mathcal{S}} |\widehat{\mathbb{P}}_h^k(s' \mid s,a) - \mathbb{P}_h(s' \mid s,a)||(V_{h+1}^{\widehat{\pi}_\theta^k} - V_{h+1}^\star)(s')| \\
&\leq \sum_{s'\in\mathcal{S}} |\widehat{\mathbb{P}}_h^k(s' \mid s,a) - \mathbb{P}_h(s' \mid s,a)|\Big[|(V_{h+1}^{\widehat{\pi}_\theta^k} - \widehat{V}_{h+1}^k)(s')| + |(\widehat{V}_{h+1}^k - V_{h+1}^\star)(s')|\Big] \\
&\overset{(i)}{\leq} \sum_{s'\in\mathcal{S}} |\widehat{\mathbb{P}}_h^k(s' \mid s,a) - \mathbb{P}_h(s' \mid s,a)|(\alpha_{h+1}+1)\widetilde{V}_{h+1}^k(s') \\
&\overset{(ii)}{\leq} \sum_{s'\in\mathcal{S}} \Big[c(\sqrt{\frac{\widehat{\mathbb{P}}_h^k(s' \mid s,a)\iota}{\max\{N_h^k(s,a),1\}}} + \frac{\iota}{\max\{N_h^k(s,a),1\}})\Big](\alpha_{h+1}+1)\widetilde{V}_{h+1}^k(s') \\
&\overset{(iii)}{\leq} \sum_{s'\in\mathcal{S}} \Big[\frac{\widehat{\mathbb{P}}_h^k(s' \mid s,a)}{H} + \frac{c^2H\iota + c\iota}{\max\{N_h^k(s,a),1\}}\Big](\alpha_{h+1}+1)\widetilde{V}_{h+1}^k(s') \\
&\leq \frac{\alpha_{h+1}+1}{H}[\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a) + 2c^2(\alpha_{h+1}+1)\frac{H^2S\iota}{\max\{N_h^k(s,a),1\}},
\end{aligned} \tag{C.13}$$

where (i) is due Lemma C.3.4 along with induction hypothesis, (ii) is due to event E_1, and (iii) is by AM-GM. Plugging equation C.11, C.12, and C.13 back in C.10, we get

$$
\begin{aligned}
&|\widehat{Q}_h^k(s,a;\theta) - Q_h^{\widehat{\pi}_\theta^k}(s,a;\theta)| \\
&\leq \min\Big\{[(1+\frac{1}{H})\alpha_{h+1} + \frac{1}{h}][\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a) + 2c\sqrt{\frac{\min\{d,S\}H^2\iota}{\max\{N_h^k(s,a)+,1\}}} \\
&+ 2c^2(\alpha_{h+1}+1)\frac{H^2 S\iota}{\max\{N_h^k(s,a),1\}}, H\Big\} \\
&\overset{(i)}{\leq} \min\Big\{[(1+\frac{1}{H})\alpha_{h+1} + \frac{1}{h}][\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a) + \beta_h^k(s,a), H\Big\} \\
&\overset{(ii)}{\leq} \alpha_h \min\{[\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s,a) + \beta_h^k(s,a), H\} \\
&\overset{(iii)}{=} \alpha_h \widetilde{Q}_h^k(s,a),
\end{aligned}
\tag{C.14}
$$

where (i) is by the definition of the bonus β_h^k (we have $2c + 12c^2 \leq C$ and $(\alpha_{h+1}+1) \leq 6$), (ii) is by the definition of α_h (note that $1 \leq \alpha_h$), and (iii) is by the definition of $\widetilde{Q}_h^k$ in Algorihtm 14. The inequality for value function follows immediately since we have

$$
\begin{aligned}
&|\widehat{V}_h^k(s;\theta) - V_h^{\widehat{\pi}_\theta^k}(s;\theta)| \\
&= |[\mathbb{D}_{\widehat{\pi}_\theta^k} \widehat{Q}_h^{\widehat{\pi}_\theta^k}](s;\theta) - [\mathbb{D}_{\widehat{\pi}_\theta^k} Q_h^k](s;\theta)| \\
&\leq \alpha_h [\mathbb{D}_{\widehat{\pi}_\theta^k} \widetilde{Q}_h^k](s) \\
&\leq \alpha_h \max_{a\in\mathcal{A}} \widetilde{Q}_h^k(s,a) \\
&= \alpha_h \widetilde{V}_h^k(s).
\end{aligned}
$$

It completes the induction step and consequently the proof of the lemma. □

Theorem C.3.6 (Similar to guarantee for UCB-VI from Azar et al. 2017)**.** *For any $\delta \in (0,1]$, if we choose β_t^k in Algorithm 14 as in Equation C.6; then, with probability at least $1-\delta$, we have*

$$
\sum_{k=1}^{K} \widetilde{V}_1^k(s_1) \leq \mathcal{O}(\sqrt{\min\{d,S\}H^4 SAK\iota} + H^3 S^2 A\iota^2).
$$

Proof of Theorem C.3.6. For a fixed k, by definition of $\widetilde{V}$ we have

$$\widetilde{V}_1^k(s_1) \le \sum_{h=1}^{H} \big(\beta_h^k(s_h^k, a_h^k) + \zeta_h^k\big),$$

where $\zeta_h^k = [\widehat{\mathbb{P}}_h^k \widetilde{V}_{h+1}^k](s_h^k, a_h^k) - \widetilde{V}_{h+1}^k(s_{h+1}^k)$. Summing over k gives us,

$$\sum_{k=1}^{K} \widetilde{V}_1^k(s_1) \le \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H} \beta_h^k(s_h^k, a_h^k)}_{(T_1)} + \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H} \zeta_h^k}_{(T_2)}.$$

Now we bound each term separately. For the term (T_1), using standard pigeonhole argument, we have

$$\begin{aligned}
(T_1) &= C\Big[\sum_{k=1}^{K}\sum_{h=1}^{H} \sqrt{\frac{\min\{d,S\}H^2\iota}{N_h^k(s_h^k,a_h^k)}} + \sum_{k=1}^{K}\sum_{h=1}^{H} \frac{H^2S\iota}{N_h^k(s_h^k,a_h^k)}\Big] \\
&= C\Big[\sqrt{\min\{d,S\}H^2\iota} \sum_{h,s,a} \sum_{i=1}^{N_h^K(s,a)} \sqrt{\frac{1}{i}} + H^2S\iota \sum_{h,s,a} \sum_{i=1}^{N_h^K(s,a)} \frac{1}{i}\Big] \\
&\le C'\Big[\sqrt{\min\{d,S\}H^2\iota} \sum_{h,s,a} \sqrt{N_h^K(s,a)} + H^2S\iota \sum_{h,s,a} \log(KH)\Big] \\
&\le C'\Big[\sqrt{\min\{d,S\}H^2\iota}\sqrt{HSA}\sqrt{KH} + H^3S^2A\iota^2\Big] \\
&\le \mathcal{O}(\sqrt{\min\{d,S\}H^4SAK\iota} + H^3S^2A\iota^2).
\end{aligned}$$

For the second term, note that ζ_h^k forms a martingale difference sequence; therefore, by Azuma-Hoeffding's inequality, with probability at least $1-\delta$, we have

$$(T_2) \le \mathcal{O}(H\sqrt{(KH)\log(1/\delta)}) = \mathcal{O}(\sqrt{H^3K\log(1/\delta)}),$$

resulting in a lower order term and completing the proof. □

Proof of Theorem C.3.1. By Algorithm 14, we have out $= \operatorname{argmin}_{k\in[K]} \widetilde{V}_1^k(s_1)$, resulting in $\widetilde{V}_1^{\text{out}}(s_1) \le$

$\frac{1}{K}\sum_{k=1}^{K}\widetilde{V}_1^k(s_1)$. Therefore, with probability at least $1-2\delta$, for any vector $\theta \in \mathcal{B}(1)$ we have

$$\begin{aligned}
V_1^\star(s_1;\theta) - V_1^{\widehat{\pi}_\theta^{\text{out}}}(s_1;\theta) &\le |V_1^\star(s_1;\theta) - \widehat{V}_1^{\text{out}}(s_1;\theta)| + |\widehat{V}_1^{\text{out}}(s_1;\theta) - V_1^{\widehat{\pi}_\theta^{\text{out}}}(s_1;\theta)| \\
&\overset{(i)}{\le} (1+\alpha_1)\widetilde{V}_1^{\text{out}}(s_1) \\
&\le 6\widetilde{V}_1^{\text{out}}(s_1) \\
&\le \frac{6}{K}\sum_{k=1}^{K}\widetilde{V}_1^k(s_1) \\
&\overset{(ii)}{\le} \mathcal{O}(\sqrt{\min\{d,S\}H^4SA\iota/K} + H^3S^2A\iota^2/K) \\
&\overset{(iii)}{\le} \epsilon,
\end{aligned}$$

where (i) is due to Lemma C.3.4 and Lemma C.3.5, (ii) is due to Theorem C.3.6, and (iii) is due to $K \ge c_K(\min\{d,S\}H^4SA\iota'/\epsilon^2 + H^3S^2A(\iota')^2/\epsilon)$ with a sufficiently large constant c_K. Rescaling δ completes the proof. □

C.4 Proof for Section 4.5

In this section we provide proofs and missing details for Section 4.5.

C.4.1 Reward-free algorithm for linear VMDPs

We use slightly modified version of the reward-free algorithm introduced by Wang et al. (2020a). The exploration phase and planning phase are displayed in Algorithm 15 and 16, respectively.

C.4.2 Proof of Theorem 4.5.2

In this section, we prove Theorem C.4.1 which implies the first claim in Theorem 4.5.2. Second and third claims in Theorem 4.5.2 immediately follow due to Theorem 4.3.1 and Theorem 4.2.5.

Theorem C.4.1. *There exist absolute constants c_β and c_K, such that for any $\epsilon \in (0,H]$ and $\delta \in (0,1]$, if we choose bonus coefficient $\beta = c_\beta \cdot d_{\text{lin}}H\sqrt{\iota}$ with $\iota = \log[d_{\text{lin}}dKH/\delta]$, and run the*

Algorithm 15 Reward-Free RL for Linear VMDPs: Exploration Phase

Hyperparameters: Bonus coefficient β.
for episode $k = 1, 2, \ldots, K$ **do**
  for step $h = H, H-1, \ldots, 1$ **do**
    $\widetilde{\Lambda}_h^k = \sum_{i=1}^{k-1} \boldsymbol{\phi}(s_h^i, a_h^i)\boldsymbol{\phi}(s_h^i, a_h^i)^\top + I$
    $\widetilde{u}_h^k(\cdot,\cdot) \leftarrow \min\{\beta \cdot \sqrt{\boldsymbol{\phi}(\cdot,\cdot)^\top (\widetilde{\Lambda}_h^k)^{-1} \boldsymbol{\phi}(\cdot,\cdot)}, H\}$
    Define $\widetilde{r}_h^k(\cdot,\cdot) \leftarrow \widetilde{u}_h^k(\cdot,\cdot)/H$
    $\widetilde{\mathbf{w}}_h^k \leftarrow (\widetilde{\Lambda}_h^k)^{-1} \sum_{i=1}^{k-1} \boldsymbol{\phi}(s_h^i, a_k^i)\widetilde{V}_{h+1}^k(s_{h+1}^i)$
    $\widetilde{Q}_h^k(\cdot,\cdot) \leftarrow \min\{(\widetilde{\mathbf{w}}_h^k)^\top \boldsymbol{\phi}(\cdot,\cdot) + \widetilde{r}_h^k(\cdot,\cdot) + \widetilde{u}_h^k(\cdot,\cdot), H\}$
    $\widetilde{V}_h^k(\cdot) = \max_{a\in\mathcal{A}} \widetilde{Q}_h^k(\cdot, a)$ and $\widetilde{\pi}_h^k(\cdot) \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} \widetilde{Q}_h^k(\cdot, a)$
  Observe initial state $s_1^k \leftarrow s_1$
  for step $h = 1, 2, \ldots, H$ **do**
    Take action $a_h^k \leftarrow \widetilde{\pi}_h^k(s_h^k)$ and observe next state s_{h+1}^k
Return $\mathcal{D} \leftarrow \{(s_h^k, a_h^k)\}_{(h,k)\in[H]\times[K]}$

Algorithm 16 Reward-Free RL for Linear VMDPs: Planning Phase

Hyperparameters: Bonus coefficient β.
Input: Dataset $\mathcal{D} = \{(s_h^k, a_h^k)\}_{(k,h)\in[K]\times[H]}$, vector $\theta \in \mathcal{B}(1)$
  samples of return function $\{\mathbf{r}_h^k\}_{(k,h)\in[K]\times[H]}$
for step $h = H, H-1, \ldots, 1$ **do**
  $\widehat{\Lambda}_h = \sum_{i=1}^{K} \boldsymbol{\phi}(s_h^i, a_h^i)\boldsymbol{\phi}(s_h^i, a_h^i)^\top + I$
  $\widehat{u}_h(\cdot,\cdot) \leftarrow \min\{\beta \cdot \sqrt{\boldsymbol{\phi}(\cdot,\cdot)^\top (\widehat{\Lambda}_h)^{-1} \boldsymbol{\phi}(\cdot,\cdot)}, H\}$
  $\widehat{\mathbf{w}}_h \leftarrow (\widehat{\Lambda}_h)^{-1} \sum_{i=1}^{K} \boldsymbol{\phi}(s_h^i, a_h^i)[\widehat{V}_{h+1}(s_{h+1}^i) + \theta^\top \mathbf{r}_h^i]$
  $\widehat{Q}_h(\cdot,\cdot) \leftarrow \min\{(\widehat{\mathbf{w}}_h)^\top \boldsymbol{\phi}(\cdot,\cdot) + \widehat{u}_h(\cdot,\cdot), H\}$
  $\widehat{V}_h(\cdot) = \max_{a\in\mathcal{A}} \widehat{Q}_h(\cdot, a)$ and $\widehat{\pi}_h(\cdot) \leftarrow \operatorname{argmax}_{a\in\mathcal{A}} \widehat{Q}_h(\cdot, a)$
Return $\pi_\theta = \{\widehat{\pi}_h\}_{h=1}^H$

exploration algorithm (Algorithm 15) for $K \geq c_K[d_{\text{lin}}^3 H^6(\iota')^2/\epsilon^2]$ *episodes where* $\iota' = \log[d_{\text{lin}}dH/(\epsilon\delta)]$*, then with probability at least* $1-\delta$*, for any* $\theta \in \mathcal{B}(1)$*, the output of the planning phase satisfies:*

$$V_1^\star(s_1;\theta) - V_1^{\pi_\theta}(s_1;\theta) \leq \epsilon,$$

where π_θ *is the output of the planning algorithm (Algorithm 16) given* θ *as input. Therefore, in this case we have*

$$m_{\text{RFE}}(\epsilon,\delta) \leq \mathcal{O}\Big(d_{\text{lin}}^3 H^6(\iota')^2/\epsilon^2\Big).$$

In this section, we denote $\boldsymbol{\phi}_h^k := \boldsymbol{\phi}(s_h^k, a_h^k)$ for $(k,h) \in [K]\times[H]$. For a scalar reward function $r' : \mathcal{S}\times\mathcal{A} \to [-1,1]$ and a policy π, we use $V_h^\pi(\cdot \mid r')$ and $Q_h^\pi(\cdot,\cdot \mid r')$ to denote the value function and Q-value function for the MDP $(\mathcal{S},\mathcal{A},H,\mathbb{P},r')$. Similarly we define the optimal value function and Q-value function denoted by $V_h^\star(\cdot \mid r')$ and $Q_h^\star(\cdot \mid r')$.

The bonus coefficient is defined to be

$$\beta = c_\beta \cdot d_{\text{lin}} H\sqrt{\iota} \tag{C.15}$$

where $\iota = \log[d_{\text{lin}}dHK/\delta]$.

We start with the following concentration lemma.

Lemma C.4.2. *Suppose Assumption 4.5.1 holds. Let* c *be some large absolute constant. Define event* E_2 *to be: for all* $(k,h,\theta) \in [K]\times[H]\times\mathcal{B}(1)$*,*

$$\begin{cases} \Big\|\sum_{i=1}^{k-1}\boldsymbol{\phi}_h^i\Big(\widetilde{V}_{h+1}^k(s_{h+1}^i) - [\mathbb{P}_h\widetilde{V}_{h+1}^k](s_h^i,a_h^i)\Big)\Big\|_{(\widetilde{\Lambda}_h^k)^{-1}} & \leq c\cdot H\sqrt{d_{\text{lin}}^2\iota}, \\ \Big\|\sum_{i=1}^{K}\boldsymbol{\phi}_h^i\Big(\widehat{V}_{h+1}(s_{h+1}^i) - [\mathbb{P}_h\widehat{V}_{h+1}](s_h^i,a_h^i)\Big)\Big\|_{(\widehat{\Lambda}_h)^{-1}} & \leq c\cdot H\sqrt{d_{\text{lin}}^2\iota}, \\ \Big\|\sum_{i=1}^{K}\boldsymbol{\phi}_h^i\Big(\theta^\top(\widehat{\mathbf{r}}_h - \mathbf{r}_h)(s_h^i,a_h^i)\Big)\Big\|_{(\widehat{\Lambda}_h)^{-1}} & \leq c\cdot\sqrt{d_{\text{lin}}\iota}, \\ |\sum_{k=1}^{K}\sum_{h=1}^{H}[\mathbb{P}_h\widetilde{V}_{h+1}^k](s_h^k,a_h^k) - \widetilde{V}_{h+1}^k(s_h^k)| & \leq c\cdot H^2\sqrt{K\iota}, \end{cases} \tag{C.16}$$

where $\iota = \log[d_{\text{lin}} dHK/\delta]$. *We have* $\mathbb{P}(E_2) \geq 1 - \delta$.

Proof of Lemma C.4.2. The first three inequalities follow from the standard concentration inequalities of the self-normalized process, a covering argument over the value functions or θ, and union bound. We refer readers to the proofs of Lemma B.3 in Jin et al. (2020c) or Lemma A.1 in Wang et al. (2020a) for details. The last inequality follows immediately from Azuma-Hoeffding's inequality since for a fixed h, $\{[\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^k, a_h^k) - \widetilde{V}_{h+1}^k(s_h^k)\}_{k\in[K]}$ is a martingale difference sequence bounded by H. □

The following lemma shows that $\widetilde{V}_1^k$ (defined in Algorithm 15) is optimistic with respect to reward function $\widetilde{r}^k$. In addition, it shows its sum over k can be controlled by $\widetilde{\mathcal{O}}(\sqrt{d_{\text{lin}}^3 H^4 K})$.

Lemma C.4.3. *Suppose Assumption 4.5.1 and event* E_2 *(defined in Lemma C.4.2) hold; we have*

$$V_1^\star(s_1 \mid \widetilde{r}^k) \leq \widetilde{V}_1^k(s_1) \quad \forall k \in [K]$$
$$\sum_{k=1}^{k} \widetilde{V}_1^k(s_1) \leq \mathcal{O}\Big(\sqrt{d_{\text{lin}}^3 H^4 K \iota^2}\Big)$$

Proof of Lemma C.4.3. Let $\overline{\mathbf{w}}_h^k = \int \widetilde{V}_{h+1}^k(s') \mathrm{d}\boldsymbol{\mu}_h(s')$; by Assumption 4.5.1, we have

$$\begin{aligned} &\|\overline{\mathbf{w}}_h^k\| \leq H\|\boldsymbol{\mu}_h(\mathcal{S})\| \leq H\sqrt{d_{\text{lin}}} \\ &[\mathbb{P}_h \widetilde{V}_{h+1}^k](s,a) = \boldsymbol{\phi}(s,a)^\top \overline{\mathbf{w}}_h^k \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A} \end{aligned} \tag{C.17}$$

For all $k, h, s, a \in [K] \times [H] \times \mathcal{S} \times \mathcal{A}$, we have

$$\begin{aligned} &\boldsymbol{\phi}(s,a)^\top \widetilde{\mathbf{w}}_h^k - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s,a) \\ &= \boldsymbol{\phi}(s,a)^\top [\widetilde{\mathbf{w}}_h^k - \overline{\mathbf{w}}_h^k] \\ &= \boldsymbol{\phi}(s,a)^\top (\widetilde{\Lambda}_h^k)^{-1} \Big(\sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i \widetilde{V}_{h+1}^k(s_{h+1}^i) - \widetilde{\Lambda}_h^k \overline{\mathbf{w}}_h^k \Big) \\ &= \boldsymbol{\phi}(s,a)^\top (\widetilde{\Lambda}_h^k)^{-1} \Big(\sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i \widetilde{V}_{h+1}^k(s_{h+1}^i) - \sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i (\boldsymbol{\phi}_h^i)^\top \overline{\mathbf{w}}_h^k - \overline{\mathbf{w}}_h^k \Big). \end{aligned}$$

Note that $(\boldsymbol{\phi}_h^i)^\top \overline{\mathbf{w}}_h^k = [\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^i, a_h^i)$. Therefore, we have

$$
\begin{aligned}
&|\boldsymbol{\phi}(s,a)^\top \widetilde{\mathbf{w}}_h^k - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s,a)| \\
&= \Big|\boldsymbol{\phi}(s,a)^\top (\widetilde{\Lambda}_h^k)^{-1}\Big[\sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i \Big(\widetilde{V}_{h+1}^k(s_{h+1}^i) - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^i, a_h^i)\Big) - \overline{\mathbf{w}}_h^k\Big]\Big| \\
&\le \Big|\boldsymbol{\phi}(s,a)^\top (\widetilde{\Lambda}_h^k)^{-1}\Big[\sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i \Big(\widetilde{V}_{h+1}^k(s_{h+1}^i) - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^i, a_h^i)\Big)\Big]\Big| + |\boldsymbol{\phi}(s,a)^\top (\widetilde{\Lambda}_h^k)^{-1} \overline{\mathbf{w}}_h^k| \\
&\le \Big\|\sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i \Big(\widetilde{V}_{h+1}^k(s_{h+1}^i) - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^i, a_h^i)\Big)\Big\|_{(\widetilde{\Lambda}_h^k)^{-1}} \cdot \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}} + \|\overline{\mathbf{w}}_h^k\|_{(\widetilde{\Lambda}_h^k)^{-1}} \cdot \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}}.
\end{aligned}
$$

Note that $\|\overline{\mathbf{w}}_h^k\|_{(\widetilde{\Lambda}_h^k)^{-1}} \le \|\overline{\mathbf{w}}_h^k\| \le H\sqrt{d_{\text{lin}}}$ since $\widetilde{\Lambda}_h^k \succeq I$.
By event E_2 we have $\Big\|\sum_{i=1}^{k-1} \boldsymbol{\phi}_h^i \Big(\widetilde{V}_{h+1}^k(s_{h+1}^i) - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^i, a_h^i)\Big)\Big\|_{(\widetilde{\Lambda}_h^k)^{-1}} \le c \cdot H\sqrt{d_{\text{lin}}^2 \iota}$. Plugging back, results in

$$
\begin{aligned}
&|\boldsymbol{\phi}(s,a)^\top \widetilde{\mathbf{w}}_h^k - [\mathbb{P}_h \widetilde{V}_{h+1}^k](s,a)| \\
&\le (H\sqrt{d_{\text{lin}}} + c \cdot H\sqrt{d_{\text{lin}}^2 \iota}) \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}} \\
&\le (c_\beta \cdot H\sqrt{d_{\text{lin}}^2 \iota}) \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}} \\
&= \beta \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}}
\end{aligned}
\tag{C.18}
$$

Now we are ready to complete the proof:

First claim: we prove the claim

$$V_h^\star(s \mid \widetilde{r}^k) \le \widetilde{V}_h^k(s) \quad \forall s \in \mathcal{S},$$

by backward induction on h. For $h = H+1$ the claim is trivial since both LHS and RHS are zero. Now suppose that we have

$$V_{h+1}^\star(s \mid \widetilde{r}^k) \le \widetilde{V}_{h+1}^k(s) \quad \forall s \in \mathcal{S}.$$

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

$$\begin{aligned}
V_h^\star(s \mid \widetilde{r}^k) &= \max_{a\in\mathcal{A}} Q_h^\star(s \mid \widetilde{r}^k) \\
&= \max_{a\in\mathcal{A}}\{\min\{\widetilde{r}_h^k(s,a) + [\mathbb{P}_h V_{h+1}^\star](s,a \mid \widetilde{r}^k), H\}\} \\
&\le \max_{a\in\mathcal{A}}\{\min\{\widetilde{r}_h^k(s,a) + [\mathbb{P}_h \widetilde{V}_{h+1}^k](s,a), H\}\} \\
&\le \max_{a\in\mathcal{A}}\{\min\{\widetilde{r}_h^k(s,a) + \boldsymbol{\phi}(s,a)^\top \widetilde{\mathbf{w}}_h^k + \beta\|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}}, H\}\} \\
&\le \max_{a\in\mathcal{A}} \widetilde{Q}_h^k(s,a) = \widetilde{V}_h^k(s),
\end{aligned}$$

where the first inequality is due to induction hypothesis and the second inequality is due to Equation C.18. It proves the induction step and completes the induction.

Second claim: Let

$$\zeta_h^k = [\mathbb{P}_h \widetilde{V}_{h+1}^k](s_h^k, a_h^k) - \widetilde{V}_{h+1}^k(s_h^k) \quad \forall (k,h) \in [K]\times[H]$$

we have

$$\begin{aligned}
\sum_{k=1}^K \widetilde{V}_1^k(s_1^k) &\le \sum_{k=1}^K \big((\widetilde{r}_1^k + u_1^k)(s_1^k, a_1^k) + (\boldsymbol{\phi}_1^k)^\top \widetilde{\mathbf{w}}_1^k\big) \\
&= \sum_{k=1}^K \big((1+1/H)\beta \cdot \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_1^k)^{-1}} + (\boldsymbol{\phi}_1^k)^\top \widetilde{\mathbf{w}}_1^k\big) \\
&\le \sum_{k=1}^K \big((2+1/H)\beta \cdot \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_1^k)^{-1}} + [\mathbb{P}_1 \widetilde{V}_2^k][s_1^k, a_1^k]\big) \\
&\le \sum_{k=1}^K \big(\widetilde{V}_2^k(s_2^k) + (2+1/H)\beta \cdot \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_1^k)^{-1}} + \zeta_1^k\big)
\end{aligned}$$

By repeatedly applying the same argument we get

$$\sum_{k=1}^K \widetilde{V}_1^k(s_1^k) \le (2+1/H)\beta \underbrace{\sum_{k=1}^K \sum_{h=1}^H \|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_h^k)^{-1}}}_{(T_1)} + \underbrace{\sum_{k=1}^K \sum_{h=1}^H \zeta_h^k}_{(T_2)}.$$

For the term (T_1) we have

$$\begin{aligned} T_1 &= \sum_{k=1}^{K}\sum_{h=1}^{H}\|\boldsymbol{\phi}(s,a)\|_{(\widetilde{\Lambda}_1^k)^{-1}} \\ &\overset{(i)}{\leq} \sqrt{KH\sum_{k=1}^{K}\sum_{h=1}^{H}(\phi_h^k)^\top(\widetilde{\Lambda}_h^k)(\phi_h^k)} \\ &\overset{(ii)}{\leq} \sqrt{KH(2d_{\text{lin}}H\log(K))}, \end{aligned}$$

where (i) uses Cauchy-Schwarz, and (ii) uses Lemma D.2 in Jin et al. (2020c) that implies $\sum_{k=1}^{K}\sum_{h=1}^{H}(\phi_h^k)^\top(\widetilde{\Lambda}_h^k)(\phi_h^k) \leq 2d_{\text{lin}}H\log(K)$.

For the term (T_2), by the third inequality in event E_2, we have

$$T_2 \leq c\cdot H^2\sqrt{K\iota}.$$

Plugging back in the original equation gives us

$$\begin{aligned} &\sum_{k=1}^{K}\widetilde{V}_1^k(s_1^k) \\ &\leq (2+1/H)\beta\cdot\sqrt{KH(2d_{\text{lin}}H\log(K))} + c\cdot H^2\sqrt{K\iota} \\ &\leq c'\sqrt{d_{\text{lin}}^3H^4K\iota^2}, \end{aligned}$$

for some absolute constant c', which completes the proof of the lemma. □

Lemma C.4.4. *Suppose Assumption 4.5.1 and event E_2 (defined in Lemma C.4.2) hold; Let $\widehat{u} = \{\widehat{u}_h\}_{h=1}^H$ (as defined in Line 5 of Algorithm 16), we have*

$$V_1^\star(s_1 \mid \widehat{u}/H) \leq \mathcal{O}\Big(\sqrt{d_{\text{lin}}^3H^4\iota^2/K}\Big)$$

Proof of Lemma C.4.4. Note that $\widehat{\Lambda}_h \succeq \widetilde{\Lambda}_h^k$ for all $k\in[K]$. Therefore for all $h\in[H]$ and $(s,a)\in\mathcal{S}\times\mathcal{A}$, we have

$$\widehat{u}_h(s,a)/H \leq \widetilde{u}_h^k(s,a)/H = \widetilde{r}_h^k(s,a)$$

Using Lemma C.4.3 we have

$$\begin{aligned} KV_1^\star(s_1 \mid \widehat{u}/H) &\le \sum_{k=1}^K V_1^*(s_1 \mid \widetilde{r}^k) \\ &\le \sum_{k=1}^K \widetilde{V}_1^k(s_1) \\ &\le \mathcal{O}\Big(\sqrt{d_{\text{lin}}^3 H^4 K \iota^2}\Big). \end{aligned}$$

Dividing both sides by K completes the proof. □

Lemma C.4.5. *Suppose Assumption 4.5.1 and event E_2 (defined in Lemma C.4.2) hold. For all $(s, a, h, \theta) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{B}(1)$ we have*

$$Q_h^*(s, a; \theta) \le \widehat{Q}_h(s, a) \le \theta^\top \mathbf{r}_h(s, a) + [\mathbb{P}_h \widehat{V}_{h+1}](s, a) + 2\widehat{u}_h(s, a).$$

Proof of Lemma C.4.5. First note that by Assumption 4.5.1, we have $\mathbf{r}_h(s, a) = W_h \boldsymbol{\phi}(s, a)$. Define

$$\overline{\mathbf{w}}_h = \int \widehat{V}_{h+1}(s') \mathrm{d}\boldsymbol{\mu}_h(s') + \theta^\top W_h.$$

By Assumption 4.5.1, we have

$$\begin{aligned} \|\overline{\mathbf{w}}_h\| &\le \Big\| \int \widehat{V}_{h+1}(s') \mathrm{d}\boldsymbol{\mu}_h(s') \Big\| + \|\theta^\top W_h\| \\ &\le H \|\boldsymbol{\mu}_h(\mathcal{S})\| + \|\theta\| \|W_h\| \\ &\le H \cdot \sqrt{d_{\text{lin}}} + \sqrt{d_{\text{lin}}} \le 2H\sqrt{d_{\text{lin}}}. \end{aligned}$$

Therefore we have

$$\begin{aligned} &\|\overline{\mathbf{w}}_h\| \le 2H \cdot \sqrt{d_{\text{lin}}} \\ &[\mathbb{P}_h \widehat{V}_{h+1}](s, a) + \theta^\top \mathbf{r}_h(s, a) = \boldsymbol{\phi}(s, a)^\top \overline{\mathbf{w}}_h \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A} \end{aligned} \tag{C.19}$$

Now using similar argument in Lemma C.4.3, for all $(s, a, h, \theta) \in \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{B}(1)$ we can have

$$
\begin{aligned}
&\left|\boldsymbol{\phi}(s,a)^\top \widehat{\mathbf{w}}_h - [\mathbb{P}_h \widehat{V}_{h+1}](s,a) - \theta^\top \mathbf{r}_h(s,a)\right| \\
&\leq \underbrace{\left\|\sum_{i=1}^K \boldsymbol{\phi}_h^i \left(\widehat{V}_{h+1}(s_{h+1}^i) - [\mathbb{P}_h \widehat{V}_{h+1}](s_h^i, a_h^i)\right)\right\|_{(\widehat{\Lambda}_h)^{-1}}}_{(T_1)} \cdot \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}} \\
&+ \underbrace{\left\|\sum_{i=1}^K \boldsymbol{\phi}_h^i \left(\theta^\top (\widehat{\mathbf{r}}_h - \mathbf{r}_h)(s_h^i, a_h^i)\right)\right\|_{(\widehat{\Lambda}_h)^{-1}}}_{(T_2)} \cdot \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}} \\
&+ \underbrace{\|\overline{\mathbf{w}}_h^k\|_{(\widehat{\Lambda}_h)^{-1}}}_{(T_3)} \cdot \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}}
\end{aligned}
$$

Note that $(T_3) = \|\overline{\mathbf{w}}_h\|_{(\widehat{\Lambda}_h)^{-1}} \leq \|\overline{\mathbf{w}}_h\| \leq 2H\sqrt{d_{\text{lin}}}$ since $\widehat{\Lambda}_h \succeq I$. The other two terms (T_1) and (T_2) are both upper-bounded by $c \cdot H\sqrt{d_{\text{lin}}^2 \iota}$ due to event E_2. Plugging back results in

$$
\begin{aligned}
&\left|\boldsymbol{\phi}(s,a)^\top \widehat{\mathbf{w}}_h - [\mathbb{P}_h \widehat{V}_{h+1}](s,a) - \theta^\top \mathbf{r}_h(s,a)\right| \\
&\leq \left[2H\sqrt{d_{\text{lin}}} + 2cH\sqrt{d_{\text{lin}}^2 \iota}\right] \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}} \\
&\leq [c_\beta \cdot H\sqrt{d_{\text{lin}}^2 \iota}] \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}} \\
&= \beta \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}}.
\end{aligned}
\tag{C.20}
$$

Now we are ready to complete the proof of the lemma. For all $(s, a, h, \theta) \leq \mathcal{S} \times \mathcal{A} \times [H] \times \mathcal{B}(1)$, we have

$$
\begin{aligned}
\widehat{Q}_h(s,a) &= \min\{\boldsymbol{\phi}(s,a)^\top \widehat{\mathbf{w}}_h + \widehat{u}_h(s,a), H\} \\
&\leq \min\{[\mathbb{P}_h \widehat{V}_{h+1}](s,a) + \theta^\top \mathbf{r}_h(s,a) + 2\beta \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}}, H\} \\
&\leq [\mathbb{P}_h \widehat{V}_{h+1}](s,a) + \theta^\top \mathbf{r}_h(s,a) + 2\min\{\beta \|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}}, H\} \\
&= [\mathbb{P}_h \widehat{V}_{h+1}](s,a) + \theta^\top \mathbf{r}_h(s,a) + 2\widehat{u}_h(s,a),
\end{aligned}
$$

where the first inequality uses Equation C.20. It completes the proof for one side of the inequality in Lemma C.4.5. For the other side we prove the claim by backward induction on h. For $h = H + 1$

we the claim is trivial. Now suppose that

$$Q^*_{h+1}(s,a;\theta) \le \widehat{Q}_{h+1}(s,a),$$

we want to prove the claim for h. We have

$$\begin{aligned}
Q^*_h(s,a;\theta) &= \min\{\theta^\top \mathbf{r}_h(s,a) + [\mathbb{P}_h V^*_{h+1}](s,a;\theta), H\} \\
&\overset{(i)}{\le} \min\{\theta^\top \mathbf{r}_h(s,a) + [\mathbb{P}_h \widehat{V}_{h+1}](s,a;\theta), H\} \\
&\overset{(ii)}{\le} \min\{\boldsymbol{\phi}(s,a)^\top \widehat{\mathbf{w}}_h + \beta\|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}}, H\} \\
&\le \min\{\boldsymbol{\phi}(s,a)^\top \widehat{\mathbf{w}}_h + \min\{\beta\|\phi(s,a)\|_{(\widehat{\Lambda}_h)^{-1}}, H\}, H\} \\
&= \min\{\boldsymbol{\phi}(s,a)^\top \widehat{\mathbf{w}}_h + \widehat{u}_h(s,a), H\} = \widehat{Q}_h(s,a),
\end{aligned}$$

where (i) uses induction hypothesis, and (ii) uses Equation C.20. It completes the proof of the lemma. □

Proof of Theorem C.4.1. With probability at least $1-\delta$, event E_2 holds and we have

$$\begin{aligned}
&\widehat{V}_1(s_1) - V^{\widehat{\pi}}_1(s_1;\theta) \\
&= \widehat{Q}_1(s_1, \widehat{\pi}_1(s_1)) - Q^{\widehat{\pi}}_1(s_1, \widehat{\pi}_1(s_1);\theta) \\
&\overset{(i)}{\le} \Big([\mathbb{P}_1 \widehat{V}_2](s_1, \widehat{\pi}_1(s_1)) + \theta^\top \mathbf{r}_1(s_1, \widehat{\pi}_1(s_1)) + 2\widehat{u}_1(s_1, \widehat{\pi}_1(s_1))\Big) \\
&\quad - \Big(\theta^\top \mathbf{r}_1(s_1, \widehat{\pi}_1(s_1)) + [\mathbb{P}_1 V^{\widehat{\pi}}_2](s_1, \widehat{\pi}_1(s_1);\theta)\Big) \\
&= 2\widehat{u}_1(s_1, \widehat{\pi}_1(s_1)) + \Big([\mathbb{P}_1 \widehat{V}_2](s_1, \widehat{\pi}_1(s_1)) - [\mathbb{P}_1 V^{\widehat{\pi}}_2](s_1, \widehat{\pi}_1(s_1);\theta)\Big) \\
&= 2\widehat{u}_1(s_1, \widehat{\pi}_1(s_1)) + \mathbb{E}_{s_2 \sim \widehat{\pi}}[\widehat{V}_2(s_2) - V^{\widehat{\pi}}_2(s_2;\theta)] \\
&= \dots \\
&= 2\mathbb{E}_{\widehat{\pi}}[\sum_{h=1}^{H} \widehat{u}_h(s_h, a_h)] \\
&= 2V^{\widehat{\pi}}_1(s_1 \mid \widehat{u}),
\end{aligned} \tag{C.21}$$

where (i) is uses Lemma C.4.5. Therefore we have

$$\begin{aligned}
&V_1^\star(s_1;\theta) - V_1^{\widehat{\pi}}(s_1;\theta) \\
&\overset{(i)}{\leq} \widehat{V}_1(s_1) - V_1^{\widehat{\pi}}(s_1;\theta) \\
&\overset{(ii)}{\leq} 2V_1^{\widehat{\pi}}(s_1 \mid \widehat{u}) \\
&\overset{(iii)}{\leq} 2V_1^\star(s_1 \mid \widehat{u}) \\
&= 2H \cdot V_1^\star(s_1 \mid \widehat{u}/H) \\
&\overset{(iv)}{\leq} \mathcal{O}\Big(\sqrt{d_{\text{lin}}^3 H^6 \iota^2 / K}\Big) \\
&\overset{(v)}{\leq} \epsilon,
\end{aligned}$$

where (i) uses Lemma C.4.5, (ii) uses Equation C.21, (iii) uses definition of optimal value function, (iv) uses Lemma C.4.4, and (v) is due to $K \geq c_K[d_{\text{lin}}^3 H^6(\iota')^2/\epsilon^2]$ with a sufficiently large constant c_K; It completes the proof.

□

C.5 Proof for Section 4.6

In this section we provide proofs and missing details for Section 4.6.

C.5.1 Proof of Theorem 4.6.3

Define $\mathbf{v}^t = V_1^{\mu^t,\nu^t}(s_1)$ and note that $\mathbb{E}[\widehat{\mathbf{v}}^t] = \mathbf{v}^t$.

Lemma C.5.1. *Define even E_3 to be:*

$$\begin{cases}
\|\frac{1}{T}\sum_{t=1}^T \mathbf{v}^t - \widehat{\mathbf{v}}^t\| \leq \mathcal{O}(\sqrt{dH^2\iota/T}), \\
V_1^{\mu^t,\nu^t}(s_1;\theta^t) - V_1^\star(s_1;\theta^t) \leq \epsilon/2 \quad \forall t \in [T].
\end{cases}$$

where $\iota = \log(d/\delta)$. We have $\mathbb{P}(E_0) \geq 1-\delta$.

Proof of Lemma C.5.1. We prove each claim holds with probability at least $1-\delta/2$; applying union bound completes the proof.

First claim. Let $\mathcal{F}_t$ be the filtration capturing all the randomness in the algorithm before iteration t. We have $\mathbb{E}[\widehat{\mathbf{v}}^t \mid \mathcal{F}_t] = \mathbf{v}^t$ and we also know that $\|\widehat{\mathbf{v}}^t\| \le H$ almost surely. By applying Lemma C.6.1, with probability at least $1-\delta$ we have

$$\|\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t - \widehat{\mathbf{v}}^t\| \le \mathcal{O}(\sqrt{H^2\log[d/\delta]/T}),$$

which completes the proof.

Second claim. We have $K \ge m_{\mathrm{RFE}}(\epsilon/2, \delta/2)$, therefore by probability at least $1-\delta/2$ (Definition 4.6.1) we have

$$V_1^{\mu^t,\dagger}(s_1;\theta^t) - V_1^{\dagger,\omega^t}(s_1;\theta^t) \le \epsilon/2,$$

Since (μ^t, ω^t) is the output of the planning phase. By definition of $V^\star$, $V^{\cdot,\dagger}$, and $V^{\dagger,\cdot}$, we further know that

$$V_1^\star(s_1;\theta^t) = \max_\nu V_1^{\dagger,\nu}(s_1;\theta^t) \ge V_1^{\dagger,\omega^t}(s_1;\theta^t)$$
$$V_1^{\mu^t,\dagger}(s_1;\theta^t) = \max_\nu V_1^{\mu^t,\nu}(s_1;\theta^t) \ge V_1^{\mu^t,\nu^t}(s_1;\theta^t)$$

Combining the three equations gives us,

$$V_1^{\mu^t,\nu^t}(s_1;\theta^t) - V_1^\star(s_1;\theta^t) \le V_1^{\mu^t,\dagger}(s_1;\theta^t) - V_1^{\dagger,\omega^t}(s_1;\theta^t) \le \epsilon/2,$$

and completes the proof. □

Lemma C.5.2. *For any $\theta \in \mathcal{B}(1)$, we have*

$$V_1^*(s_1;\theta) \le \max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle + \max_\nu \min_\mu \mathrm{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})$$

Proof of Lemma C.5.2. Let $\alpha = \max_\nu \min_\mu \mathrm{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})$; therefore, for every max-player policy

ν there exist a min-player policy $\overline{\mu}(\nu)$ such that $\mathrm{dist}(\mathbf{V}_1^{\overline{\mu}(\nu),\nu}(s_1), \mathcal{C}) \leq \alpha$. Let $\Gamma_{\mathcal{C}}$ be the (Euclidean) projection operator into $\mathcal{C}$. We have

$$\begin{aligned}
V_1^*(s_1;\theta) &= V_1^{\mu^\star,\nu^*}(s_1;\theta) \\
&\leq V_1^{\overline{\mu}(\nu^*),\nu^*}(s_1;\theta) \\
&= \langle \theta, \mathbf{V}_1^{\overline{\mu}(\nu^*),\nu^*}(s_1)\rangle \\
&= \langle \theta, \mathbf{V}_1^{\overline{\mu}(\nu^*),\nu^*}(s_1) - \Gamma_{\mathcal{C}}\left[\mathbf{V}_1^{\overline{\mu}(\nu^*),\nu^*}(s_1)\right]\rangle + \langle \theta, \Gamma_{\mathcal{C}}\left[\mathbf{V}_1^{\overline{\mu}(\nu^*),\nu^*}(s_1)\right]\rangle \\
&\leq \|\theta\| \mathrm{dist}(\mathbf{V}_1^{\overline{\mu}(\nu^*),\nu^*}(s_1), \mathcal{C}) + \max_{\mathbf{x}\in\mathcal{C}} \langle \theta, \mathbf{x}\rangle \\
&\leq \alpha + \max_{\mathbf{x}\in\mathcal{C}} \langle \theta, \mathbf{x}\rangle,
\end{aligned}$$

Recalling that $\alpha = \max_\nu \min_\mu \mathrm{dist}(\mathbf{V}_1^{\mu,\nu}(s_1), \mathcal{C})$ completes the proof. □

Proof of Theorem 4.6.3. With probability at least $1-\delta$, event E_3 (as in Definition C.5.1) holds and

we have

$$
\begin{aligned}
&\operatorname{dist}\Big(\frac{1}{T}\sum_{t=1}^{T}\mathbf{V}_1^{\mu^t,\nu^t}(s_1),\mathcal{C}\Big)\\
&=\operatorname{dist}\Big(\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t,\mathcal{C}\Big)\\
&\overset{(i)}{=}\max_{\theta\in\mathcal{B}(1)}\Big[\langle\theta,\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t)\rangle-\max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle\Big]\\
&=\max_{\theta\in\mathcal{B}(1)}\Big[\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta,\widehat{\mathbf{v}}^t\rangle-\max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle\big)+\langle\theta,\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}^t-\widehat{\mathbf{v}}^t\rangle\Big]\\
&\overset{(ii)}{\leq}\max_{\theta\in\mathcal{B}(1)}\Big[\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta,\widehat{\mathbf{v}}^t\rangle-\max_{\mathbf{x}\in\mathcal{C}}\langle\theta,\mathbf{x}\rangle\big)\Big]+\mathcal{O}(\sqrt{dH^2\iota/T})\\
&\overset{(iii)}{\leq}\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta^t,\widehat{\mathbf{v}}^t\rangle-\max_{\mathbf{x}\in\mathcal{C}}\langle\theta^t,\mathbf{x}\rangle\big)+\mathcal{O}(\sqrt{H^2/T})+\mathcal{O}(\sqrt{dH^2\iota/T})\\
&\overset{(iv)}{\leq}\max_{\nu}\min_{\mu}\operatorname{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})+\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta^t,\widehat{\mathbf{v}}^t\rangle-\mathbf{V}_1^{*}(s_1;\theta^t)\big)+\mathcal{O}(\sqrt{dH^2\iota/T})\\
&\overset{(v)}{\leq}\max_{\nu}\min_{\mu}\operatorname{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})+\epsilon/2+\frac{1}{T}\sum_{t=1}^{T}\big(\langle\theta^t,\widehat{\mathbf{v}}^t\rangle-\mathbf{V}_1^{\mu^t,\nu^t}(s_1;\theta^t)\big)+\mathcal{O}(\sqrt{dH^2\iota/T})\\
&=\max_{\nu}\min_{\mu}\operatorname{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})+\epsilon/2+\frac{1}{T}\sum_{t=1}^{T}\langle\theta^t,\widehat{\mathbf{v}}^t-\mathbf{v}^t\rangle+\mathcal{O}(\sqrt{dH^2\iota/T})\\
&\overset{(vi)}{\leq}\max_{\nu}\min_{\mu}\operatorname{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})+\epsilon/2+\mathcal{O}(\sqrt{dH^2\iota/T})\\
&\overset{(vii)}{\leq}\max_{\nu}\min_{\mu}\operatorname{dist}(\mathbf{V}_1^{\mu,\nu}(s_1),\mathcal{C})+\epsilon
\end{aligned}
$$

where (i) is by Equation C.3, (ii) is by first inequality in event E_3 together with Cauchy-Schwarz, (iii) is by guarantee of OGA in Theorem C.2.1, (iv) is by Lemma C.5.2, (v) is by second inequality in event E_3, (vi) is by first inequality in event E_3 together with Cauchy-Schwarz, and finally (vii) is by setting $T\geq c\big(dH^2\iota/\epsilon^2\big)$ for large enough constant c, completing the proof. □

C.5.2 Proof of Theorem 4.6.4

Algorithm

Exploration phase. Similar to Algorithm 14, we use VI-Zero proposed by Liu et al. (2020) with different choice of hyperparameters. The pseudo-code is provided in Algorithm 17.

Planning phase. In the planning phase, given $\theta \in \mathcal{B}(1)$ as input we can use any planning algorithm for $\widehat{\mathcal{G}}_\theta = (\mathcal{S}, \mathcal{A}, \mathcal{B}, H, \widehat{\mathbb{P}}^{\text{out}}, \langle \theta, \widehat{\mathbf{r}} \rangle)$ where $\widehat{\mathbf{r}}$ is empirical estimate of $\mathbf{r}$ using collected samples $\{\mathbf{r}_h^k\}$. One such algorithm could be Nash value iteration (e.g. see Algorithm 5 in Liu et al. 2020) that computes Nash equilibrium policy for a *known* model.

Algorithm 17 VI-Zero for VMGs: Exploration Phase

Hyperparameters: Bonus β_t.
Initialize: for all $(s,a,b,h) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H]$: $\widetilde{Q}_h(s,a,b) \leftarrow H$ and $N_h(s,a,b) \leftarrow 0$,
  for all $(s,a,b,h,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{B} \times [H] \times \mathcal{S}$: $N_h(s,a,b,s') \leftarrow 0$,
  $\Delta \leftarrow 0$.
for episode $k = 1, 2, \ldots, K$ **do**
  for step $h = H, H-1, \ldots, 1$ **do**
    for state-action pair $(s,a,b) \in \mathcal{S} \times \mathcal{A} \times \mathcal{B}$ **do**
      $t \leftarrow N_h(s,a,b)$.
      if $t > 0$ **then**
        $\widetilde{Q}_h(s,a,b) \leftarrow \min\{[\widehat{\mathbb{P}}_h \widetilde{V}_{h+1}](s,a,b) + \beta_t, H\}$.
    for state $s \in \mathcal{S}$ **do**
      $\widetilde{V}_h(s) \leftarrow \max_{(a,b) \in \mathcal{A} \times \mathcal{B}} \widetilde{Q}_h(s,a,b)$ and $\pi_h(s) \leftarrow \operatorname{argmax}_{(a,b) \in \mathcal{A} \times \mathcal{B}} \widetilde{Q}_h(s,a,b)$
  if $\widetilde{V}(s_1) \le \Delta$ **then**
    $\Delta \leftarrow \widetilde{V}(s_1)$ and $\widehat{\mathbb{P}}^{\text{out}} \leftarrow \widehat{\mathbb{P}}_h$
  for step $h = 1, 2, \ldots, H$ **do**
    Take action $(a_h, b_h) \leftarrow \pi_h(s_h)$ and observe next state s_{h+1}
    Update $N_h(s_h, a_h, b_h) \leftarrow N_h(s_h, a_h, b_h) + 1$
    Update $N_h(s_h, a_h, b_h, s_{h+1}) \leftarrow N_h(s_h, a_h, b_h, s_{h+1}) + 1$
    $\widehat{\mathbb{P}}_h(\cdot \mid s_h, a_h, b_h) \leftarrow N_h(s_h, a_h, b_h, \cdot)/N_h(s_h, a_h, b_h)$
Return $\widehat{\mathbb{P}}^{\text{out}}$

Proof of Theorem 4.6.4

Proof is almost identical to proof of Theorem **??** provided in Appendix C.3; therefore, we only provide the statement for the main lemmas without proof.

Let $\widehat{\mathbb{P}}^k$ and $\widehat{\mathbf{r}}^k$ be our empirical estimates of the transition and the return vectors at the beginning of the k^{th} episode in Algorithm 17 and define $\widehat{\mathcal{G}}^k = (\mathcal{S}, \mathcal{A}, \mathcal{B}, H, \widehat{\mathbb{P}}^k, \widehat{\mathbf{r}}^k)$. We use $N_h^k(s,a,b)$ to denote the number of times we have visited state-action (s,a,b) in step h before k^{th} episode in Algorithm 17. We use superscript k to denote variable corresponding to episode k; in particular, $(s_1^k, a_1^k, b_1^k, \ldots, s_H^k, a_H^k, b_H^k)$ is the trajectory we have visited in the k^{th} episode.

For any $\theta \in \mathcal{B}(1)$, let $\widehat{\mathcal{G}}_\theta^k$ be the scalarized VMG using vector θ (defined in Section 4.6). We use $\widehat{V}^k(\cdot;\theta)$, $\widehat{Q}^k(\cdot,\cdot,\cdot;\theta)$, and $(\widehat{\mu}_\theta^k, \widehat{\nu}_\theta^k) = (\widehat{\mu}^k(\cdot;\theta), \widehat{\nu}^k(\cdot;\theta))$ to denote the optimal value function, optimal Q-value function, and Nash equilibrium policy of $\widehat{\mathcal{G}}_\theta^k$ respectively. Therefore, we have

$$\begin{aligned}
\widehat{Q}_h^k(s,a,b;\theta) &= [\widehat{\mathbb{P}}_h^k \widehat{V}_{h+1}^k](s,a,b;\theta) + \widehat{r}_h^k(s,a,b;\theta), \\
\widehat{V}_h^k(s;\theta) &= \min_{\mu} \max_{\nu} [\mathbb{D}_{\mu\times\nu} \widehat{Q}_h^k](s;\theta), \\
\widehat{V}_h^k(s;\theta) &= [\mathbb{D}_{\widehat{\mu}_\theta^k \times \widehat{\nu}_\theta^k} \widehat{Q}_h^k](s;\theta).
\end{aligned} \tag{C.22}$$

Theorem C.5.3 (restatement of Theorem 4.6.4)**.** *There exist absolute constants c_β and c_K, such that for any $\epsilon \in (0,H]$, $\delta \in (0,1]$, if we choose bonus $\beta_t = c_\beta\big(\sqrt{\min\{d,S\}H^2\iota/t} + H^2S\iota/t\big)$ where $\iota = \log[dSABKH/\delta]$, and run the exploration phase (Algorithm 17) for $K \geq c_K\big(\min\{d,S\}H^4SAB\iota'/\epsilon^2 + H^3S^2AB(\iota')^2/\epsilon\big)$ episodes where $\iota' = \log[dSABH/(\epsilon\delta)]$, then with probability at least $1-\delta$, the algorithm satisfies for all $\theta \in \mathcal{B}(1)$*

$$V_1^{\mu_\theta,\dagger}(s_1;\theta) - V_1^{\dagger,\nu_\theta}(s_1;\theta) = [V_1^{\mu_\theta,\dagger}(s_1;\theta) - V_1^\star(s_1;\theta)] + [V_1^\star(s_1;\theta) - V_1^{\dagger,\nu_\theta}(s_1;\theta)] \leq \epsilon,$$

where (μ_θ, ν_θ) is the output of any planning algorithm (e.g., Nash value iteration) for the Markov game $\widehat{\mathcal{G}}_\theta^{\text{out}}$. Therefore, we have

$$m_{\text{RFE}}(\epsilon,\delta) \leq \mathcal{O}\Big(\frac{\min\{d,S\}H^4SAB\iota'}{\epsilon^2} + \frac{H^3S^2AB(\iota')^2}{\epsilon}\Big).$$

The bonus for episode k can be written as

$$\beta_h^k(s,a,b) = c_\beta\Big(\sqrt{\frac{\min\{d,S\}H^2\iota}{\max\{N_h^k(s,a,b),1\}}} + \frac{H^2S\iota}{\max\{N_h^k(s,a,b),1\}}\Big), \tag{C.23}$$

where $\iota = \log[dSABKH/\delta]$ and c_β is some large absolute constant.

We start with the concentration lemma similar to Lemma C.3.3.

Lemma C.5.4. *Let c be some large absolute constant. Define event E_4 to be: for all $(s,a,b,s',h) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\times\mathcal{S}\times[H]$, $k\in[K]$, and $\theta\in\mathcal{B}(1)$,*

$$\begin{cases} |[(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)V_{h+1}^\star](s,a,b;\theta)| & \le c\sqrt{\frac{\min\{d,S\}H^2\iota}{\max\{N_h^k(s,a,b),1\}}}, \\ |(\widehat{r}_h^k - r_h)(s,a,b;\theta)| & \le c\sqrt{\frac{\iota}{\max\{N_h^k(s,a,b),1\}}}, \\ |(\widehat{\mathbb{P}}_h^k - \mathbb{P}_h)(s' \mid s,a,b)| & \le c\Big(\sqrt{\frac{\widehat{\mathbb{P}}_h^k(s'|s,a,b)\iota}{\max\{N_h^k(s,a,b),1\}}} + \frac{\iota}{\max\{N_h^k(s,a,b),1\}}\Big), \end{cases} \tag{C.24}$$

where $\iota = \log[dSABKH/\delta]$. We have $\mathbb{P}(E_4)\ge 1-\delta$.

Similar to Lemma C.3.4, the following lemma shows that the optimal value functions of $\widehat{\mathcal{G}}_\theta^k$ are close to the optimal value functions of $\mathcal{G}_\theta$ and their difference is controlled by $\widetilde{Q}$ and $\widetilde{V}$ computed in Algorithm 17.

Lemma C.5.5. *Suppose event E_4 holds (defined in Lemma C.5.4); then, for all $(s,a,b,k,h,\theta)\in\mathcal{S}\times\mathcal{A}\times\mathcal{B}\times[K]\times[H]\times\mathcal{B}(1)$ we have*

$$\begin{aligned} &|\widehat{Q}_h^k(s,a,b;\theta) - Q_h^\star(s,a,b;\theta)| \le \widetilde{Q}_h^k(s,a,b), \\ &|\widehat{V}_h^k(s;\theta) - V_h^\star(s;\theta)| \le \widetilde{V}_h^k(s). \end{aligned} \tag{C.25}$$

Similar to Lemma C.3.5, now we are ready to introduce the main lemma that shows value of $\widehat{\pi}_\theta^k$ under the true model is close to its value under empirical model. The difference is controlled by $\widetilde{Q}$ and $\widetilde{V}$ computed in Algorithm 17.

Lemma C.5.6. *Suppose event E_4 holds (defined in Lemma C.5.4); then, for all $(s,a,b,k,h,\theta) \in \mathcal{S}\times\mathcal{A}\times\mathcal{B}\times[K]\times[H]\times\mathcal{B}(1)$ we have*

$$\begin{aligned}|\widehat{Q}_h^k(s,a,b;\theta) - Q_h^{\dagger,\widehat{\nu}_\theta^k}(s,a,b;\theta)| &\le \alpha_h \widetilde{Q}_h^k(s,a,b),\\ |\widehat{V}_h^k(s;\theta) - V_h^{\dagger,\widehat{\nu}_\theta^k}(s;\theta)| &\le \alpha_h \widetilde{V}_h^k(s),\end{aligned} \tag{C.26}$$

and

$$\begin{aligned}|\widehat{Q}_h^k(s,a,b;\theta) - Q_h^{\widehat{\mu}_\theta^k,\dagger}(s,a,b;\theta)| &\le \alpha_h \widetilde{Q}_h^k(s,a,b),\\ |\widehat{V}_h^k(s;\theta) - V_h^{\widehat{\mu}_\theta^k,\dagger}(s;\theta)| &\le \alpha_h \widetilde{V}_h^k(s),\end{aligned} \tag{C.27}$$

where $\alpha_{H+1}=1$ and $\alpha_h = [(1+\frac{1}{H})\alpha_{h+1}+\frac{1}{H}]$; we have $1\le\alpha_h\le 5$ for $h\in[H]$.

Similar to Lemma C.3.6, we can bound the uncertainty using the following lemma.

Theorem C.5.7. *For any $\delta\in(0,1]$, if we choose β_t^k in Algorithm 17 as in Equation C.23; then, with probability at least $1-\delta$, we have*

$$\sum_{k=1}^K \widetilde{V}_1^k(s_1) \le \mathcal{O}(\sqrt{\min\{d,S\}H^4SABK\iota} + H^3S^2AB\iota^2).$$

Proof of Theorem C.5.3 (restatement of Theorem 4.6.4). By Algorithm 17, we have $\text{out} = \text{argmin}_{k\in[K]}\widetilde{V}_1^k(s_1)$, resulting in $\widetilde{V}_1^{\text{out}}(s_1) \le \frac{1}{K}\sum_{k=1}^K \widetilde{V}_1^k(s_1)$. Therefore, with probability at least $1-2\delta$, for any vector $\theta\in\mathcal{B}(1)$ we have

$$\begin{aligned}V_1^{\widehat{\mu}_\theta^{\text{out}},\dagger}(s_1;\theta) - V_1^{\dagger,\widehat{\nu}_\theta^{\text{out}}}(s_1;\theta) &\le |V_1^{\widehat{\mu}_\theta^{\text{out}},\dagger}(s_1;\theta) - \widehat{V}_1^{\text{out}}(s_1;\theta)| + |\widehat{V}_1^{\text{out}}(s_1;\theta) - V_1^{\dagger,\widehat{\nu}_\theta^{\text{out}}}(s_1;\theta)|\\ &\overset{(i)}{\le} 2\alpha_1\widetilde{V}_1^{\text{out}}(s_1)\\ &\le 10\widetilde{V}_1^{\text{out}}(s_1)\\ &\le \frac{10}{K}\sum_{k=1}^K \widetilde{V}_1^k(s_1)\\ &\overset{(ii)}{\le} \mathcal{O}(\sqrt{\min\{d,S\}H^4SAB\iota/K} + H^3S^2AB\iota^2/K)\\ &\overset{(iii)}{\le} \epsilon,\end{aligned}$$

where (i) is due to Lemma C.5.6, (ii) is due to Theorem C.5.7, and (iii) is due to $K \geq c_K(\min\{d, S\}H^4SAB\iota'/\epsilon^2 + H^3S^2AB(\iota')^2/\epsilon)$ with a sufficiently large constant c_K. Rescaling δ completes the proof. □

C.6 Auxiliary tools

Lemma C.6.1 (Hoeffding type inequality for norm-subGaussian, Corollary 7 in Jin et al. 2019)**.** *Let $\{\mathbf{X}_t\}_{t\in[T]}$ be a d-dimensional vector-valued random variable. Consider filtration $\{\mathcal{F}_t\}_{t\in[T]}$ and define $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_t]$. If $\|\mathbf{X}_t\| \leq R$ almost surely, then it holds with probability at least $1-\delta$,*

$$\|\sum_{t=1}^{T} \mathbf{X}_t - \mathbb{E}_{t-1}[\mathbf{X}_t]\| \leq \mathcal{O}(R\sqrt{T\log[d/\delta]}).$$

Appendix D

Remaining Proofs of Chapter 5

D.1 Algorithm Olive

In this section, we analyze algorithm Olive proposed in Jiang et al. (2017), which is based on hypothesis elimination. We prove that, despite Olive was originally designed for solving low Bellman rank problems, it naturally learns RL problems with low BE dimension as well.

The main advantage of Olive comparing to Golf is that Olive does not require the completeness assumption. In return, Olive has several disadvantages including worse sample complexity, and no sublinear regret.

The pseudocode of Olive is presented in Algorithm 18, where in each phase the algorithm contains the following three main components:

- Line 3 (Optimistic planning): compute the most optimistic value function f^k from the candidate set $\mathcal{B}^{k-1}$, and choose π^k to be its greedy policy.
- Line 4-7 (Estimate Bellman error): estimate the Bellman error of f^k under π^k; output π^k if the estimated error is small, and otherwise activate the elimination procedure.

Algorithm 18 OLIVE $(\mathcal{F}, \zeta_{\text{act}}, \zeta_{\text{elim}}, n_{\text{act}}, n_{\text{elim}})$

1: **Initialize**: $\mathcal{B}^0 \quad \mathcal{F}$, $\mathcal{D}_h \quad \emptyset$ for all h, k.
2: **for phase** $k = 1, 2, \ldots$ **do**
3: **Choose policy** $\pi^k = \pi_{f^k}$, where $f^k = \operatorname{argmax}_{f \in \mathcal{B}^{k-1}} f(s_1, \pi_f(s_1))$.
4: **Execute** π^k for n_{act} episodes and *refresh* $\mathcal{D}_h$ to include the fresh (s_h, a_h, r_h, s_{h+1}) tuples.
5: **Estimate** $\hat{\mathcal{E}}(f^k, \pi^k, h)$ for all $h \in [H]$, where

$$\hat{\mathcal{E}}(g, \pi^k, h) = \frac{1}{|\mathcal{D}_h|} \sum_{(s,a,r,s') \in \mathcal{D}_h} \left(g_h(s,a) - r - \max_{a' \in \mathcal{A}} g_{h+1}(s', a') \right).$$

6: **if** $\sum_{h=1}^{H} \hat{\mathcal{E}}(f^k, \pi^k, h) \le H\zeta_{\text{act}}$ **then**
7: Terminate and output π^k.
8: Pick any $t \in [H]$ for which $\hat{\mathcal{E}}(f^k, \pi^k, t) \ge \zeta_{\text{act}}$.
9: **Execute** π^k for n_{elim} episodes and *refresh* $\mathcal{D}_h$ to include the fresh (s_h, a_h, r_h, s_{h+1}) tuples.
10: **Estimate** $\hat{\mathcal{E}}(f, \pi^k, t)$ for all $f \in \mathcal{F}$.
11: **Update** $\mathcal{B}^k = \left\{ f \in \mathcal{B}^{k-1} : \left| \hat{\mathcal{E}}(f, \pi^k, t) \right| \le \zeta_{\text{elim}} \right\}$.

- Line 8-11 (Eliminate functions with large Bellman error): pick a step $t \in [H]$ where the estimated Bellman error exceeds the activation threshold ζ_{act}; eliminate all functions in the candidate set whose Bellman error at step t exceeds the elimination threshold ζ_{elim}.

We comment that OLIVE is computationally inefficient in general because implementing the optimistic planning part requires solving an NP-hard problem in the worst case (Theorem 4, Dann et al., 2018).

D.1.1 Theoretical guarantees

Now, we are ready to present the theoretical guarantee for OLIVE.

Theorem D.1.1 (OLIVE). *Under Assumption 5.2.1, there exists absolute constant c such that if we choose*

$$\zeta_{act} = \frac{2\epsilon}{H}, \ \zeta_{elim} = \frac{\epsilon}{2H\sqrt{d}}, \ n_{act} = \frac{H^2\iota}{\epsilon^2}, \ \text{and } n_{elim} = \frac{H^2 d \log(\mathcal{N}_{\mathcal{F}}(\zeta_{elim}/8)) \cdot \iota}{\epsilon^2}$$

where $d = \dim_{\text{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon/H)$ and $\iota = c\log(Hd/\delta\epsilon)$, then with probability at least $1-\delta$, Algorithm 18 will output an $\mathcal{O}(\epsilon)$-optimal policy using at most $\mathcal{O}(H^3 d^2 \log[\mathcal{N}_{\mathcal{F}}(\zeta_{elim}/8)] \cdot \iota/\epsilon^2)$ episodes.

Theorem D.1.1 claims that OLIVE learns an ϵ-optimal policy of an MDP with BE dimension d within $\tilde{\mathcal{O}}(H^3d^2\log(\mathcal{N}_{\mathcal{F}})/\epsilon^2)$ episodes. When specialized to low Bellman rank problems, our sample complexity has the same quadratic dependence on Bellman rank d as in Jiang et al. (2017).

Comparing to GOLF, the major advantage of OLIVE is that OLIVE does not require completeness assumption (Assumption 5.2.2) to work. Nevertheless, OLIVE only learns the RL problems that have low BE dimension with respect to distribution family $\mathcal{D}_{\mathcal{F}}$, not $\mathcal{D}_{\Delta}$. The sample complexity of OLIVE is also worse than the sample complexity GOLF (as presented in Corollary 5.4.3).

Finally, we comment that interpreting OLIVE through the lens of BE dimension, makes the proof of Theorem D.1.1 surprisingly natural, which follows from the definition of BE dimension along with some standard concentration arguments.

D.1.2 Interpret Olive with BE dimension

In this subsection, we explain the key idea behind OLIVE through the lens of BE dimension.

To provide a clean high-level view, let us assume all estimates are accurate for now, and the activation threshold ζ_{act} and the elimination threshold ζ_{elim} satisfy $\zeta_{\mathrm{elim}}\sqrt{d} \le \zeta_{\mathrm{act}}$, where $d = \dim_{\mathrm{BE}}\big(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \zeta_{\mathrm{act}}\big)$. Since $\mathcal{E}(Q^\star, \pi, h) \equiv 0$ for any (π, h), $Q^\star$ is always in the candidate set. Therefore, the optimistic planning (Line 3) guarantees $\max_a f_1^k(s_1, a) \ge V_1^\star(s_1)$.

If the Bellman error summation is small (Line 6) i.e., $\sum_{h=1}^H \mathcal{E}(f^k, \pi^k, h) \le H\zeta_{\mathrm{act}}$, then by simple policy loss decomposition (e.g., Lemma 1 in Jiang et al. (2017)) and the optimism of f^k, π^k is $H\zeta_{\mathrm{act}}$-optimal. Otherwise, the elimination procedure is activated at some step t satisfying $\mathcal{E}(f^k, \pi^k, t) \ge \zeta_{\mathrm{act}}$ and all f with $\mathcal{E}(f, \pi^k, t) \ge \zeta_{\mathrm{elim}}$ get eliminated. The *key* observation here is:

> *If the elimination procedure is activated at step h in phase $k_1 < \ldots < k_m$, then the roll-in distribution of $\pi^{k_1}, \ldots, \pi^{k_m}$ at step h is an ζ_{act}-independent sequence with respect to the class of Bellman residuals $(I - \mathcal{T}_h)\mathcal{F}$ at step h. Therefore, we should have $m \le d$.*

For the sake of contradiction, assume $m \ge d+1$. Let us prove $\pi^{k_1}, \ldots, \pi^{k_{d+1}}$ is a ζ_{act}-independent

sequence. Firstly, for any $j \in [d+1]$, since f^{k_j} is not eliminated in phase $k_1, \ldots, k_{j-1}$, we have

$$\sqrt{\sum_{i=1}^{j-1} \big(\mathcal{E}(f^{k_j}, \pi^{k_i}, h)\big)^2} \le \sqrt{d} \times \zeta_{\mathrm{elim}} \le \zeta_{\mathrm{act}}.$$

Besides, because the elimination procedure is activated at step h in phase k_j, we have $\mathcal{E}(f^{k_j}, \pi^{k_j}, h) \ge \zeta_{\mathrm{act}}$. By Definition 5.3.1, we obtain that the roll-in distribution of π^{k_j} at step h is ζ_{act}-independent of those of $\pi^{k_1}, \ldots, \pi^{k_{j-1}}$ for $j \in [d+1]$, which contradicts the definition $d = \dim_{\mathrm{BE}}\big(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \zeta_{\mathrm{act}}\big)$. As a result, the elimination procedure can happen at most d times for each $h \in [H]$, which means the algorithm should terminate within $dH + 1$ phases and output an $H\zeta_{\mathrm{act}}$-optimal policy.

D.2 V-type BE Dimension and Algorithms

The definition of Bellman rank, mentioned in Definition 5.3.5 and Proposition 5.3.6, is slightly different from the original definition in Jiang et al. (2017). We denote the former by **Q-type** and the latter (the original definition) by **V-type**. In this section we introduce V-type BE Dimension as well as V-type variants of GOLF and OLIVE. We show that similar results also hold for the V-type variants.

Definition D.2.1 (V-type Bellman rank)**.** The V-type Bellman rank is the minimum integer d so that there exists $\phi_h : \mathcal{F} \to \mathbb{R}^d$ and $\psi_h : \mathcal{F} \to \mathbb{R}^d$ for each $h \in [H]$, such that for any $f, f' \in \mathcal{F}$, the average V-type Bellman error

$$\mathcal{E}_{\mathrm{V}}(f, \pi_{f'}, h) := \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(s_h, a_h) \mid s_h \sim \pi_{f'}, a_h \sim \pi_f] = \langle \phi_h(f), \psi_h(f') \rangle,$$

where $\|\phi_h(f)\|_2 \cdot \|\psi_h(f')\|_2 \le \zeta$, and ζ is the normalization parameter.

The only difference between these two definitions is how we sample a_h. In the Q-type definition we have $a_h \sim \pi_{f'}$ (the roll-in policy), however in the V-type definition we have $a_h \sim \pi_f$ (the greedy policy of the function evaluated in the Bellman error) instead. It is worth mentioning that the

Q-type and V-type bellman error coincide whenever $f = f'$; namely, $\mathcal{E}(f, \pi_f, h) = \mathcal{E}_{\mathrm{V}}(f, \pi_f, h)$ for all $f \in \mathcal{F}$.

We can similarly define the V-type variant of BE Dimension. At a high level, **V-type BE dimension** $\dim_{\mathrm{VBE}}(\mathcal{F}, \Pi, \epsilon)$ measures the complexity of finding a function in $\mathcal{F}$ such that its expected Bellman error under any state distribution in Π is smaller than ϵ.

Definition D.2.2 (V-type BE dimension)**.** Let $(I - \mathcal{T}_h)V_{\mathcal{F}} \subseteq (\mathcal{S} \to \mathbb{R})$ be the state-wise Bellman residual class of $\mathcal{F}$ at step h which is defined as

$$(I - \mathcal{T}_h)V_{\mathcal{F}} := \big\{ s \mapsto (f_h - \mathcal{T}_h f_{h+1})(s, \pi_{f_h}(s)) : f \in \mathcal{F} \big\}.$$

Let $\Pi = \{\Pi_h\}_{h=1}^H$ be a collection of H probability measure families over $\mathcal{S}$. The **V-type ϵ-BE dimension** of $\mathcal{F}$ with respect to Π is defined as

$$\dim_{\mathrm{VBE}}(\mathcal{F}, \Pi, \epsilon) := \max_{h \in [H]} \dim_{\mathrm{DE}} \big((I - \mathcal{T}_h)V_{\mathcal{F}}, \Pi_h, \epsilon \big).$$

Relation with low V-type Bellman rank With slight abuse of notation, denote by $\mathcal{D}_{\mathcal{F},h}$ the collection of all probability measures over $\mathcal{S}$ at the h^{th} step, which can be generated by rolling in with a greedy policy π_f with $f \in \mathcal{F}$. Similar to Proposition 5.3.6, the following proposition claims that the V-type BE dimension of $\mathcal{F}$ with respect to $\mathcal{D}_{\mathcal{F}} := \{\mathcal{D}_{\mathcal{F},h}\}_{h \in [H]}$ is always upper bounded by its V-type Bellman rank up to some logarithmic factor.

Proposition D.2.3 (low V-type Bellman rank $\subset$ low V-type BE dimension)**.** *If an MDP with function class $\mathcal{F}$ has V-type Bellman rank d with normalization parameter ζ, then*

$$\dim_{\mathrm{VBE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \le \mathcal{O}(1 + d \log(1 + \zeta/\epsilon)).$$

The proof of Proposition D.2.3 is almost the same as that of Proposition 5.3.6 in Appendix D.4.1. We omit it here since the only modification is to replace Q-type Bellman rank with its V-type variant

Algorithm 19 V-type GOLF $(\mathcal{F}, K, \beta)$

1: **Initialize**: $\mathcal{D}_1, \ldots, \mathcal{D}_H \leftarrow \emptyset$, $\mathcal{B}^0 \leftarrow \mathcal{F}$.
2: **for epoch** k from 1 to K **do**
3: **Choose policy** $\pi^k = \pi_{f^k}$, where $f^k = \operatorname{argmax}_{f \in \mathcal{B}^{k-1}} f(s_1, \pi_f(s_1))$.
4: **for step** h from 1 to H **do**
5: **Collect** a tuple (s_h, a_h, r_h, s_{h+1}) by executing π^k at step $1, \ldots, h-1$ and taking action uniformly at random at step h.
6: **Augment** $\mathcal{D}_h = \mathcal{D}_h \cup \{(s_h, a_h, r_h, s_{h+1})\}$ for all $h \in [H]$.
7: **Update**

$$\mathcal{B}^k = \left\{ f \in \mathcal{F} : \ \mathcal{L}_{\mathcal{D}_h}(f_h, f_{h+1}) \le \inf_{g \in \mathcal{G}_h} \mathcal{L}_{\mathcal{D}_h}(g, f_{h+1}) + \beta \text{ for all } h \in [H] \right\},$$

$$\text{where } \mathcal{L}_{\mathcal{D}_h}(\xi_h, \zeta_{h+1}) = \sum_{(s,a,r,s') \in \mathcal{D}_h} [\xi_h(s,a) - r - \max_{a' \in \mathcal{A}} \zeta_{h+1}(s', a')]^2.$$

8: **Output** π^{out} sampled uniformly at random from $\{\pi^k\}_{k=1}^K$.

wherever it is used.

D.2.1 Algorithm V-type Golf

In this section we describe the V-type variant of GOLF. The pseudocode is provided in Algorithm 19. Its only difference from the Q-type analogue is in Line 5: for each $h \in [H]$, we roll in with policy π^k to sample s_h, and then instead of continuing following π^k we take random action at step h.

Now we present the theoretical guarantee for Algorithm 19. Its proof is almost the same as that of Corollary 5.4.3 and can be found in appendix D.7.2.

Theorem D.2.4 (V-type GOLF)**.** *Under Assumption 5.2.1, 5.4.1, there exists an absolute constant c such that for any given $\epsilon > 0$, if we choose $\beta = c \log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\epsilon^2/(d|\mathcal{A}|H^2))]$, then with probability at least 0.99, π^{out} is $\mathcal{O}(\epsilon)$-optimal, if*

$$K \ge \Omega\left(\frac{H^2 d|\mathcal{A}|}{\epsilon^2} \cdot \log\left[\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}\left(\frac{\epsilon^2}{H^2 d|\mathcal{A}|} \right) \cdot \frac{Hd|\mathcal{A}|}{\epsilon} \right] \right),$$

where $d = \min_{\Pi \in \{\mathcal{D}_\Delta, \mathcal{D}_\mathcal{F}\}} \dim_{\text{VBE}}(\mathcal{F}, \Pi, \epsilon/H)$.

Compared with Theorem D.2.5 (V-type OLIVE), Theorem D.2.4 (V-type GOLF) has the following

two advantages.

- The sample complexity in Theorem D.2.4 depends linearly on the V-type BE-dimension while the dependence in Theorem D.2.5 is quadratic.
- Theorem D.2.4 applies to RL problems of finite V-type BE dimension with respect to either $\mathcal{D}_{\mathcal{F}}$ or $\mathcal{D}_{\Delta}$. In comparison, Theorem D.2.5 provides no guarantee for the $\mathcal{D}_{\Delta}$ case.

Finally, we comment that for the low Q-type BE dimension family, we provide both regret and sample complexity guarantees while for the low V-type counterpart, we only derive sample complexity result due to the need of taking actions uniformly at random in Algorithm 20 and Algorithm 19. Dong et al. (2020) propose an algorithm that can achieve $\sqrt{T}$-regret for problems of low V-type Bellman rank. It is an interesting open problem to study whether similar techniques can be adapted to the low V-type BE dimension setting so that we can also obtain $\sqrt{T}$-regret.

D.2.2 Algorithm V-type Olive

In this section, we describe the original OLIVE (i.e., V-type OLIVE) proposed by Jiang et al. (2017), and its theoretical guarantee in terms of V-type BE dimension.

The pseudocode is provided in Algorithm 20. Its only difference from Algorithm 18 is Line 9-10: note that V-type Bellman rank needs the action at step t to be greedy with respect to the function f instead of being picked by the roll-in policy π^k, so we choose action a_t uniformly at random and use the importance-weighted estimator to estimate the Bellman error for each f.

We have the following similar theoretical guarantee for Algorithm 20. Its proof is almost the same as that of Theorem D.1.1 and can be found in Appendix D.7.1.

Theorem D.2.5 (V-type OLIVE)**.** *Assume realizability (Assumption 5.2.1) holds and $\mathcal{F}$ is finite. There exists absolute constant c such that if we choose*

$$\zeta_{\text{act}} = \frac{2\epsilon}{H},\ \zeta_{elim} = \frac{\epsilon}{2H\sqrt{d}},\ n_{act} = \frac{H^2\iota}{\epsilon^2},\ \textit{and}\ n_{elim} = \frac{H^2 d|\mathcal{A}|\log(|\mathcal{F}|)\cdot\iota}{\epsilon^2}$$

Algorithm 20 V-type OLIVE $(\mathcal{F}, \zeta_{\mathrm{act}}, \zeta_{\mathrm{elim}}, n_{\mathrm{act}}, n_{\mathrm{elim}})$

Initialize: $\mathcal{B}^0 \leftarrow \mathcal{F}$, $\mathcal{D}_h \leftarrow \emptyset$ for all h, k.
for phase $k = 1, 2, \ldots$ **do**
  Choose policy $\pi^k = \pi_{f^k}$, where $f^k = \mathrm{argmax}_{f \in \mathcal{B}^{k-1}} f(s_1, \pi_f(s_1))$.
  Execute π^k for n_{act} episodes and *refresh* $\mathcal{D}_h$ to include the fresh (s_h, a_h, r_h, s_{h+1}) tuples.
  Estimate $\tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h)$ for all $h \in [H]$, where

$$\tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) = \frac{1}{|\mathcal{D}_h|} \sum_{(s,a,r,s') \in \mathcal{D}_h} \left(f_h^k(s,a) - r - \max_{a' \in \mathcal{A}} f_{h+1}^k(s', a') \right).$$

  if $\sum_{h=1}^{H} \tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) \le H\zeta_{\mathrm{act}}$ **then**
    Terminate and output π^k.
  Pick any $t \in [H]$ for which $\tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, t) > \zeta_{\mathrm{act}}$.
  Collect n_{elim} episodes by executing π^k for step $1, \ldots, t-1$ and picking action uniform at random for step t. *Refresh* $\mathcal{D}_h$ to include the fresh (s_h, a_h, r_h, s_{h+1}) tuples.
  Estimate $\hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, t)$ for all $f \in \mathcal{F}$, where

$$\hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, h) = \frac{1}{|\mathcal{D}_h|} \sum_{(s,a,r,s') \in \mathcal{D}_h} \frac{\mathbf{1}[a = \pi_f(s)]}{1/|\mathcal{A}|} \left(f_h(s,a) - r - \max_{a' \in \mathcal{A}} f_{h+1}(s', a') \right).$$

  Update $\mathcal{B}^k = \left\{ f \in \mathcal{B}^{k-1} : \left| \hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, t) \right| \le \zeta_{\mathrm{elim}} \right\}$.

where $d = \dim_{\mathrm{VBE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon/H)$ *and* $\iota = c \log[Hd|\mathcal{A}|/\delta\epsilon]$, *then with probability at least* $1 - \delta$, *Algorithm 20 will output an* $\mathcal{O}(\epsilon)$*-optimal policy using at most* $\mathcal{O}(H^3 d^2 |\mathcal{A}| \log(|\mathcal{F}|) \cdot \iota / \epsilon^2)$ *episodes.*

For problems with Bellman rank d and finite function class $\mathcal{F}$, Theorem D.2.5 together with Proposition D.2.3 guarantees $\tilde{\mathcal{O}}(H^3 d^2 |\mathcal{A}| \log(|\mathcal{F}|)/\epsilon^2)$ samples suffice for finding an ϵ-optimal policy, which matches the result in Jiang et al. (2017). For function class $\mathcal{F}$ of infinite cardinality but with finite covering number, we can first compute an $\mathcal{O}(\zeta_{\mathrm{elim}})$-cover of $\mathcal{F}$, which we denote as $\mathcal{Z}_\rho$, and then run Algorithm 20 on $\mathcal{Z}_\rho$. By following almost the same arguments in the proof of Theorem D.2.5 (the only difference is to replace $Q^\star$ by its proxy in $\mathcal{Z}_\rho$), we can show Algorithm 20 will output an $\mathcal{O}(\epsilon)$-optimal policy using at most $\tilde{\Omega}(H^3 d^2 |\mathcal{A}| \log(N)/\epsilon^2)$ episodes where $N = \mathcal{N}_{\mathcal{F}}(\mathcal{O}(\zeta_{\mathrm{elim}}))$.

D.2.3 Discussions on Q-type versus V-type

In this work, we have introduced two complementary definitions of Bellman rank: Q-type Bellman rank and V-type Bellman rank. And we prove they are upper bounds for Q-type and V-type BE dimension, respectively. Here, we want to emphasize that both Q-type and V-type Bellman rank

have their own advantages. Specifically, the Q-type version has the following strengths.

1. There are natural RL problems whose Q-type Bellman rank is small, while their V-type Bellman rank is very large, e.g., the linear function approximation setting studied in in Zanette et al. (2020a).

2. All the existing sample complexity results for the V-type cases scale linearly with respect to the number of actions, while those for the Q-type cases are independent of the number of actions. Therefore, for control problems such as Linear Quadratic Regulator (LQR), which has both small Q-type and V-type Bellman rank but infinite number of actions, the notion of Q-type is more suitable.

On the other hand, there are problems that naturally induce low V-type Bellman rank but have large Q-type Bellman rank, e.g., reactive POMDPs.

D.3 Examples

In this section, we introduce examples with low BE dimension. We will start with linear models and their variants, then introduce kernel MDPs, and finally present kernel reactive POMDPs which have low BE dimension, but possibly large Bellman rank and large Eluder dimension. All the proofs for this section are deferred to Appendix D.8.

D.3.1 Linear models and their variants

In this subsection, we review problems with linear structure in ascending order of generality. We start with the definition of linear MDPs (e.g., Jin et al., 2020c).

Definition D.3.1 (Linear MDPs)**.** We say an MDP is linear of dimension d if for each $h \in [H]$, there exists feature mappings $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, and d unknown signed measures $\psi_h = (\psi_h^{(1)}, \ldots, \psi_h^{(d)})$ over $\mathcal{S}$, and an unknown vector $\theta_h^r \in \mathbb{R}^d$, such that $\mathbb{P}_h(\cdot \mid s, a) = \phi_h(s,a)^\top \psi_h(\cdot)$ and $r_h(s,a) = \phi_h(s,a)^\top \theta_h^r$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

We remark that existing works (e.g., Jin et al., 2020c) usually assumxe ϕ is *known* to the learner. Next, we review a more general setting—the linear completeness setting (e.g., Zanette et al., 2020a).

Definition D.3.2 (Linear completeness setting)**.** We say an MDP is in the linear completeness setting of dimension d, if there exists a feature mapping $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, such that for the linear function class $\mathcal{F}_h = \{\phi_h(\cdot)^\top \theta \mid \theta \in \mathbb{R}^d\}$, both Assumption 5.2.1 and 5.2.2 are satisfied.

We make three comments here. Firstly, we note that linear MDPs automatically satisfy both linear realizability and linear completeness assumptions, therefore are special cases of the linear completeness setting with the same ambient dimension. Secondly, only assuming linear realizability but without completeness is insufficient for sample-efficient learning (see exponential lower bounds in Weisz et al. (2021)). Finally, as mentioned in Appendix D.2.3, though MDPs in the linear completeness setting have low Q-type Bellman rank, their V-type Bellman rank can be arbitrarily large.

Finally, we review the generalized linear completeness setting (Wang et al., 2019), which generalizes the linear completeness setting by adding nonlinearity.

Definition D.3.3 (Generalized linear completeness setting)**.** We say an MDP is in the generalized linear completeness setting of dimension d, if there exists a feature mapping $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, and a link function σ, such that for the generalized linear function class $\mathcal{F}_h = \{\sigma(\phi_h(\cdot)^\top \theta) \mid \theta \in \mathbb{R}^d\}$, both Assumption 5.2.1 and 5.2.2 are satisfied, and the link function is strictly monotone, i.e., there exist $0 < c_1 < c_2 < \infty$ such that $\sigma'(x) \in [c_1, c_2]$ for all x.

One can directly verify by definition that when we choose link function $\sigma(x) = x$ in the generalized linear completeness setting, it will reduce to the standard linear version. Besides, it is known (Russo and Van Roy, 2013) the generalized linear completeness setting is a special case of low Eluder dimension, thus belonging to the low BE dimension family. Finally, we comment that despite the

linear completeness setting belongs to the low Bellman rank family, the generalized version does not because of the possible nonlinearity of the link function.

D.3.2 Effective dimension and kernel MDPs

In this subsection, we introduce the notion of effective dimension. With this notion, we prove a useful proposition that any linear kernel function class with low effective dimension also has low Eluder dimension. This proposition directly implies that kernel MDPs are special cases of low Eluder dimension, which are also special cases of low BE dimension.

Effective dimension We start with the definition of effective dimension for a set, which is also known as critical information gain in Du et al. (2021).

Definition D.3.4 (ϵ-effective dimension of a set)**.** The ϵ-effective dimension of a set $\mathcal{X}$ is the minimum integer $d_{\text{eff}}(\mathcal{X}, \epsilon) = n$ such that

$$\sup_{x_1,\ldots,x_n \in \mathcal{X}} \frac{1}{n} \log\det\left(\mathrm{I} + \frac{1}{\epsilon^2}\sum_{i=1}^{n} x_i x_i^\top\right) \leq e^{-1}. \tag{D.1}$$

Based on this definition, we can also define the effective dimension of a function class.

Definition D.3.5 (ϵ-effective dimension of a function class)**.** Given a function class $\mathcal{F}$ defined on $\mathcal{X}$, its ϵ-effective dimension $d_{\text{eff}}(\mathcal{F}, \epsilon) = n$ is the minimum integer n such that there exists a separable Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$ so that

- for every $f \in \mathcal{F}$ there exists $\theta_f \in B_{\mathcal{H}}(1)$ satisfying $f(x) = \langle \theta_f, \phi(x) \rangle_{\mathcal{H}}$ for all $x \in \mathcal{X}$,
- $d_{\text{eff}}(\phi(\mathcal{X}), \epsilon) = n$ where $\phi(\mathcal{X}) = \{\phi(x) : \ x \in \mathcal{X}\}$.

The following proposition shows that the Eluder dimension of any function class is always upper bounded by its effective dimension.

Proposition D.3.6 (low effective dimension $\subset$ low Eluder dimension)**.** *For any function class $\mathcal{F}$ and domain $\mathcal{X}$, we have*

$$\dim_{\mathrm{E}}(\mathcal{F}, \epsilon) \leq \dim_{\mathrm{eff}}(\mathcal{F}, \epsilon/2).$$

On the other hand, we remark that effective dimension requires the existence of a benign linear structure in certain Hilbert spaces. In constrast, Eluder dimension does not require such conditions. Therefore, the function class of low Eluder dimension is more general than the function class of low effective dimension.

Kernel MDPs Now, we are ready to define kernel MDPs and prove it is a subclass of low Eluder dimension.

Definition D.3.7 (Kernel MDPs)**.** In a kernel MDP of effective dimension $d(\epsilon)$, for each step $h \in [H]$, there exist feature mappings $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathcal{H}$ and $\psi_h : \mathcal{S} \to \mathcal{H}$ where $\mathcal{H}$ is a separable Hilbert space, so that the transition measure can be represented as the inner product of features, i.e., $\mathbb{P}_h(s' \mid s, a) = \langle \phi_h(s, a), \psi_h(s') \rangle_{\mathcal{H}}$. Besides, the reward function is linear in ϕ, i.e., $r_h(s, a) = \langle \phi_h(s, a), \theta_h^r \rangle_{\mathcal{H}}$ for some $\theta_h^r \in \mathcal{H}$. Here, ϕ is known to the learner while ψ and θ^r are *unknown*. Moreover, a kernel MDP satisfies the following regularization conditions: for all h

- $\|\theta_h^r\|_{\mathcal{H}} \leq 1$ and $\|\phi_h(s, a)\|_{\mathcal{H}} \leq 1$ for all s, a.
- $\| \sum_{s \in \mathcal{S}} \mathcal{V}(s) \psi_h(s) \|_{\mathcal{H}} \leq 1$ for any function $\mathcal{V} : \mathcal{S} \to [0, 1]$.
- $\dim_{\mathrm{eff}}(\mathcal{X}_h, \epsilon) \leq d(\epsilon)$ for all h and ϵ, where $\mathcal{X}_h = \{\phi_h(s, a) : \ (s, a) \in \mathcal{S} \times \mathcal{A}\}$.

In order to learn kernel MDPs, we need to construct a proper function class $\mathcal{F}$. Formally, for each $h \in [H]$, we choose $\mathcal{F}_h = \{\phi_h(\cdot, \cdot)^\top \theta \ \mid \ \theta \in B_{\mathcal{H}}(H + 1 - h)\}$. One can easily verify $\mathcal{F}$ satisfies both realizability and completeness by following the same arguments as in linear MDPs (Jin et al., 2020c). In order to apply GOLF or OLIVE, we also need to show it has low BE dimension and bounded log-covering number. Below, we prove in sequence that $\mathcal{F}$ has low Eluder dimension and low log-covering number. Therefore, kernel MDPs fall into our low BE dimension framework.

Proposition D.3.8 (kernel MDPs $\subset$ low Eluder dimension)**.** *Let $\mathcal{M}$ be a kernel MDP of effective dimension $d(\epsilon)$, then*

$$\dim_{\mathrm{E}}(\mathcal{F}, \epsilon) \leq d(\epsilon/2H).$$

Proposition D.3.8 follows directly from Proposition D.3.6 by rescaling the parameters. Utilizing Proposition D.3.8, we can further prove the log-covering number of $\mathcal{F}$ is also upper bounded by the effective dimension of the kernel MDP up to some logarithmic factor.

Proposition D.3.9 (bounded covering number)**.** *Let $\mathcal{M}$ be a kernel MDP of effective dimension $d(\epsilon)$, then*

$$\log \mathcal{N}_{\mathcal{F}}(\epsilon) \leq \mathcal{O}\big(Hd(\epsilon) \cdot \log(1 + d(\epsilon)H/\epsilon)\big).$$

D.3.3 Effective Bellman rank and kernel reactive POMDPs

To begin with, we introduce the definition of effective Bellman rank and prove that it is always an upper bound for BE dimension. We will see effective Bellman rank serves as a useful tool for controlling the BE dimension of the example discussed in this section—kernel reactive POMDPs.

Q-type effective Bellman rank We start with Q-type ϵ-effective Bellman rank which is simply the ϵ-effective dimension of a special feature set.

Definition D.3.10 (Q-type ϵ-effective Bellman rank)**.** The Q-type ϵ-effective Bellman rank is the minimum integer d so that

- There exists $\phi_h : \mathcal{F} \to \mathcal{H}$ and $\psi_h : \mathcal{F} \to \mathcal{H}$ for each $h \in [H]$ where $\mathcal{H}$ is a separable Hilbert space, such that for any $f, f' \in \mathcal{F}$, the average Bellman error

$$\mathcal{E}(f, \pi_{f'}, h) := \mathbb{E}_{\pi_{f'}}[(f_h - \mathcal{T}_h f_{h+1})(s_h, a_h)] = \langle \phi_h(f), \psi_h(f') \rangle_{\mathcal{H}}$$

  where $\|\phi_h(f)\|_{\mathcal{H}} \leq \zeta$, and ζ is the normalization parameter.

- $d = \max_{h \in [H]} d_{\mathrm{eff}}(\mathcal{X}_h(\psi, \mathcal{F}), \epsilon/\zeta)$ where $\mathcal{X}_h(\psi, \mathcal{F}) = \{\psi_h(f_h) : \ f_h \in \mathcal{F}_h\}$.

One can easily verify that when $\mathcal{H}$ is a finite-dimensional Euclidean space, the ϵ-effective Bellman rank is always upper bounded by the original Bellman rank up to a logarithmic factor in ζ and ϵ^{-1}. Moreover, the effective Bellman rank can be much smaller than the original Bellman rank if the induced feature set $\{\mathcal{X}_h(\psi, \mathcal{F})\}_{h\in[H]}$ approximately lies in a low-dimensional linear subspace. Therefore, effective Bellman rank can be viewed as a strict generalization of the original version.

Proposition D.3.11 (low Q-type effective Bellman rank $\subset$ low Q-type BE dimension)**.** *Suppose function class $\mathcal{F}$ has Q-type ϵ-effective Bellman rank d, then*

$$\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \le d.$$

Proposition D.3.11 claims that problems with low Q-type effective Bellman rank also have low Q-type BE dimension.

V-type effective Bellman rank We can similarly define the V-type variant of effective Bellman rank, and prove it is always an upper bound for V-type BE dimension.

Definition D.3.12 (V-type ϵ-effective Bellman rank)**.** The V-type ϵ-effective Bellman rank is the minimum integer d so that

- There exists $\phi_h : \mathcal{F} \to \mathcal{H}$ and $\psi_h : \mathcal{F} \to \mathcal{H}$ for each $h \in [H]$ where $\mathcal{H}$ is a separable Hilbert space, such that for any $f, f' \in \mathcal{F}$, the average Bellman error

  $$\mathcal{E}_{\mathrm{V}}(f, \pi_{f'}, h) := \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(s_h, a_h) \mid s_h \sim \pi_{f'}, a_h \sim \pi_f] = \langle \phi_h(f), \psi_h(f') \rangle_{\mathcal{H}}$$

  where $\|\phi_h(f)\|_{\mathcal{H}} \le \zeta$, and ζ is the normalization parameter.
- $d = \max_{h\in[H]} d_{\mathrm{eff}}(\mathcal{X}_h(\psi, \mathcal{F}), \epsilon/\zeta)$ where $\mathcal{X}_h(\psi, \mathcal{F}) = \{\psi_h(f_h) : \ f_h \in \mathcal{F}_h\}$.

Proposition D.3.13 (low V-type effective Bellman rank $\subset$ low V-type BE dimension)**.** *Suppose*

function class $\mathcal{F}$ has V-type ϵ-effective Bellman rank d, then

$$\dim_{\mathrm{VBE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \leq d.$$

The proof of Proposition D.3.13 is almost the same as that of Proposition D.3.11. We omit it since the only modification is to replace Q-type effective Bellman rank with its V-type variant wherever it is used.

We want to briefly comment that the majority of examples introduced in Du et al. (2021) have low effective Bellman rank. For example, low occupancy complexity, linear Q^*/V^*, linear Bellman complete and Q^* state aggregation have low Q-type effective Bellman rank. And the feature selection problem has low V-type Bellman rank.

Kernel reactive POMDPs We start with the definition of POMDPs. A POMDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathbb{T}, \mathbb{O}, r, H)$ where $\mathcal{S}$ denotes the set of hidden states, $\mathcal{A}$ denotes the set of actions, $\mathcal{O}$ denotes the set of observations, $\mathbb{T}$ denotes the transition measure, $\mathbb{O}$ denotes the emission measure, $r = \{r_h\}_{h=1}^H$ denotes the collections of reward functions, and H denotes the length of each episode. At the beginning of each episode, the agent always starts from a fixed initial state. At each step $h \in [H]$, after reaching s_h, the agent will observe $o_h \sim \mathbb{O}_h(\cdot \mid s_h)$. Then the agent picks action a_h, receives $r_h(o_h, a_h)$ and transits to $s_{h+1} \sim \mathbb{T}_h(\cdot \mid s_h, a_h)$. In POMDPs, the agent can never directly observe the states $s_{1:H}$. It can only observe $o_{1:H}$ and $r_{1:H}$. Now we are ready to formally define kernel reactive POMDPs.

Definition D.3.14 (Kernel reactive POMDPs)**.** A kernel reactive POMDP is a POMDP that additionally satisfies the following two conditions

- For each $h \in [H]$, there exist mappings $\phi_h : \mathcal{S} \times \mathcal{A} \to \mathcal{H}$ and $\psi_h : \mathcal{S} \to \mathcal{H}$ where $\mathcal{H}$ is a separable Hilbert space, such that $\mathbb{T}_h(s' \mid s, a) = \langle \phi_h(s, a), \psi_h(s') \rangle_{\mathcal{H}}$ for all s', a, s. Moreover, for any function $\mathcal{V} : \mathcal{S} \to [0, 1]$, $\| \sum_{s' \in \mathcal{S}} \mathcal{V}(s') \psi_h(s') \|_{\mathcal{H}} \leq 1$.
- (Reactiveness) The optimal action-value function Q^* only depends on the current observation

and action, i.e., for each $h \in [H]$, there exists function $f_h^* : \mathcal{O} \times \mathcal{A} \to [0,1]$ such that for all $\tau_h = [o_1, a_1, r_1, \ldots, o_h]$ and a_h

$$Q_h^*(\tau_h, a_h) = f_h^*(o_h, a_h).$$

The following proposition shows that when a kernel reactive POMDP has low effective dimension, it also has low V-type BE dimension.

Proposition D.3.15 (kernel reactive POMDPs $\subset$ low V-type BE dimension)**.** *Any kernel reactive POMDP and function class $\mathcal{F} \subseteq (\mathcal{O} \times \mathcal{A} \to [0,1])$ satisfy*

$$\dim_{\mathrm{VBE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \leq \max_{h \in [H]} d_{\mathrm{eff}}(\mathcal{X}_h, \epsilon/2),$$

where $\mathcal{X}_h = \{\mathbb{E}_{\pi_f}[\phi_h(s_h, a_h)] : \ f \in \mathcal{F}\}$.

We comment that when $\mathcal{H}$ *approximately* aligns with a low-dimensional linear subspace, the V-type effective Bellman rank in Proposition D.3.15 will also be low. However, the Eluder dimension of $\mathcal{F}$ can be arbitrarily large because we basically pose no structural assumption on $\mathcal{F}$. Besides, its V/Q-type original Bellman rank can also be arbitrarily large, because $\mathcal{H}$ may be infinite-dimensional and the observation set $\mathcal{O}$ may be exponentially large. If we additionally assume $\mathcal{F}$ satisfies realizability ($f^* \in \mathcal{F}$), then we can apply V-type OLIVE and obtain polynomial sample-complexity guarantee.

D.4 Proofs for BE Dimension

In this section, we provide formal proofs for the results stated in Section 5.3.

D.4.1 Proof of Proposition 5.3.6

The proof is basically the same as that of Example 3 in Russo and Van Roy (2013) with minor modification.

Proof. Without loss of generality, assume $\max\{\|\phi_h(f)\|_2, \|\psi_h(f)\|_2\} \leq \sqrt{\zeta}$, otherwise we can satisfy this assumption by rescaling the feature mappings. Assume there exists $h \in [H]$ such that $\dim_{\mathrm{DE}}((I - \mathcal{T}_h)\mathcal{F}, \mathcal{D}_{\mathcal{F},h}, \epsilon) \geq m$. Let $\mu_1, \ldots, \mu_m \in \mathcal{D}_{\mathcal{F},h}$ be a an ϵ-independent sequence with respect to $(I - \mathcal{T}_h)\mathcal{F}$. By Definition 5.3.1, there exists $f^1, \ldots, f^m$ such that for all $i \in [m]$, $\sqrt{\sum_{t=1}^{i-1}(\mathbb{E}_{\mu_t}[f_h^i - \mathcal{T}_h f_{h+1}^i])^2} \leq \epsilon$ and $|\mathbb{E}_{\mu_i}[f_h^i - \mathcal{T}_h f_{h+1}^i]| > \epsilon$. Since $\mu_1, \ldots, \mu_n \in \mathcal{D}_{\mathcal{F},h}$, there exist $g^1, \ldots, g^n \in \mathcal{F}$ so that μ_i is generated by executing π_{g^i} for all $i \in [n]$.

By the definition of Bellman rank, this is equivalent to: for all $i \in [m]$, $\sqrt{\sum_{t=1}^{i-1}(\langle \phi_h(g^i), \psi_h(f^t)\rangle)^2} \leq \epsilon$ and $|\langle \phi_h(g^i), \psi_h(f^i)\rangle| > \epsilon$.

For notational simplicity, define $\mathbf{x}_i = \phi_h(g^i)$, $\mathbf{z}_i = \psi_h(f^i)$ and $\mathbf{V}_i = \sum_{t=1}^{i-1} \mathbf{z}_t \mathbf{z}_t^\top + \frac{\epsilon^2}{\zeta} \cdot \mathbf{I}$. The previous argument directly implies: for all $i \in [m]$, $\|\mathbf{x}_i\|_{\mathbf{V}_i} \leq \sqrt{2}\epsilon$ and $\|\mathbf{x}_i\|_{\mathbf{V}_i} \cdot \|\mathbf{z}_i\|_{\mathbf{V}_i^{-1}} > \epsilon$. Therefore, we have $\|\mathbf{z}_i\|_{\mathbf{V}_i^{-1}} \geq \frac{1}{\sqrt{2}}$.

By the matrix determinant lemma,

$$\det[\mathbf{V}_m] = \det[\mathbf{V}_{m-1}](1 + \|\mathbf{z}_m\|_{\mathbf{V}_m^{-1}}^2) \geq \frac{3}{2}\det[\mathbf{V}_{m-1}] \geq \ldots \geq \det[\frac{\epsilon^2}{\zeta} \cdot \mathbf{I}](\frac{3}{2})^{m-1} = (\frac{\epsilon^2}{\zeta})^d(\frac{3}{2})^{m-1}.$$

On the other hand,

$$\det[\mathbf{V}_m] \leq (\frac{\mathrm{trace}[\mathbf{V}_m]}{d})^d \leq (\frac{\zeta(m-1)}{d} + \frac{\epsilon^2}{\zeta})^d.$$

Therefore, we obtain

$$(\frac{3}{2})^{m-1} \leq (\frac{\zeta^2(m-1)}{d\epsilon^2} + 1)^d.$$

Take logarithm on both sides,

$$m \leq 4\left[1 + d\log(\frac{\zeta^2(m-1)}{d\epsilon^2} + 1)\right],$$

which, by simple calculation, implies

$$m \leq \mathcal{O}\left(1 + d\log(\frac{\zeta^2}{\epsilon^2} + 1)\right).$$

□

D.4.2 Proof of Proposition 5.3.7

Proof. Assume $\delta_{z_1}, \dots, \delta_{z_m}$ is an ϵ-independent sequence of distributions with respect to $(I - \mathcal{T}_h)\mathcal{F}$, where $\delta_{z_i} \in \mathcal{D}_\Delta$. By Definition 5.3.1, there exist functions $f^1, \dots, f^m \in \mathcal{F}$ such that for all $i \in [m]$, we have $|(f_h^i - \mathcal{T}_h f_{h+1}^i)(z_i)| > \epsilon$ and $\sqrt{\sum_{t=1}^{i-1} |(f_h^i - \mathcal{T}_h f_{h+1}^i)(z_t)|^2} \le \epsilon$. Define $g_h^i = \mathcal{T}_h f_{h+1}^i$. Note that $g_h^i \in \mathcal{F}_h$ because $\mathcal{T}_h \mathcal{F}_{h+1} \subset \mathcal{F}_h$. Therefore, we have for all $i \in [m]$, $|(f_h^i - g_h^i)(z_i)| > \epsilon$ and $\sqrt{\sum_{t=1}^{i-1} |(f_h^i - g_h^i)(z_t)|^2} \le \epsilon$ with $f_h^i, g_h^i \in \mathcal{F}_h$. By Definition 5.2.4 and 5.2.5, this implies $\dim_{\mathrm{E}}(\mathcal{F}_h, \epsilon) \ge m$, which completes the proof. □

D.4.3 Proof of Proposition 5.3.8

Proof. For any $m \in \mathbb{N}^+$, denote by $e_1, \dots, e_m$ the basis vectors in $\mathbb{R}^m$, and consider the following linear bandits ($|\mathcal{S}| = H = 1$) problem.

- The action set $\mathcal{A} = \{a_i = (1; e_i) \in \mathbb{R}^{m+1} : \ i \in [m]\}$.
- The function set $\mathcal{F}_1 = \{f_{\theta_i}(a) = a^\top \theta_i : \ \theta_i = (1; e_i), \ i \in [m]\}$.
- The reward function is always zero, i.e., $r \equiv 0$.

Eluder dimension For any $\epsilon \in (0, 1]$, $a_1, \dots, a_{m-1}$ is an ϵ-independent sequence of points because: (a) for any $t \in [m-1]$, $\sum_{i=1}^{t-1} (f_{\theta_t}(a_i) - f_{\theta_{t+1}}(a_i))^2 = 0$; (b) for any $t \in [m-1]$, $f_{\theta_t}(a_t) - f_{\theta_{t+1}}(a_t) = 1 \ge \epsilon$. Therefore, $\min_{h \in [H]} \dim_{\mathrm{E}}(\mathcal{F}_h, \epsilon) = \dim_{\mathrm{E}}(\mathcal{F}_1, \epsilon) \ge m - 1$.

Bellman rank It is direct to see the Bellman residual matrix is $\mathcal{E} := \Theta^\top \Theta \in \mathbb{R}^{m \times m}$ with rank m, where $\Theta = [\theta_1, \theta_2, \dots, \theta_m]$. As a result, the Bellman rank is at least m.

BE dimension First, note in this setting $(I - \mathcal{T}_1)\mathcal{F}$ is simply $\mathcal{F}_1$ (because $\mathcal{F}_2 = \{0\}$ and $r \equiv 0$), and $\mathcal{D}_{\mathcal{F}}$ coincides with $\mathcal{D}_\Delta$, so it suffices to show $\dim_{\mathrm{DE}}(\mathcal{F}_1, \mathcal{D}_\Delta, \epsilon) \le 5$.

Assume $\dim_{\mathrm{DE}}(\mathcal{F}_1, \mathcal{D}_\Delta, \epsilon) = k$. Then there exist $q_1, \dots, q_k \in \mathcal{A}$ and $w_1, \dots, w_k \in \mathcal{A}$ such that for all $t \in [k]$, $\sqrt{\sum_{i=1}^{t-1} (\langle q_t, w_i \rangle)^2} \le \epsilon$ and $|\langle q_t, w_t \rangle| > \epsilon$. By simple calculation, we have $q_i^\top w_j \in [1, 2]$ for

all $i, j \in [k]$. Therefore, if $\epsilon > 2$, then $k = 0$ because $|\langle q_t, w_t \rangle| \le 2$; if $\epsilon \le 2$, then $k \le 5$ because $\sqrt{k-1} \le \sqrt{\sum_{i=1}^{k-1} (\langle q_k, w_i \rangle)^2} \le \epsilon$. □

D.5 Proofs for Golf

In this section, we provide formal proofs for the results stated in Section 5.4.

D.5.1 Proof of Theorem 5.4.2

We start the proof with the following two lemmas. The first lemma shows that with high probability any function in the confidence set has low Bellman-error over the collected datasets $\mathcal{D}_1, \ldots, \mathcal{D}_H$ as well as the distributions from which $\mathcal{D}_1, \ldots, \mathcal{D}_H$ are sampled.

Lemma D.5.1. *Let $\rho > 0$ be an arbitrary fixed number. If we choose $\beta = c\big(\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)/\delta] + K\rho\big)$ with some large absolute constant c in Algorithm 5, then with probability at least $1-\delta$, for all $(k,h) \in [K] \times [H]$, we have*

(a) $\sum_{i=1}^{k-1} \mathbb{E}[\big(f_h^k(s_h,a_h) - (\mathcal{T} f_{h+1}^k)(s_h,a_h)\big)^2 \mid s_h, a_h \sim \pi^i] \le \mathcal{O}(\beta)$.

(b) $\sum_{i=1}^{k-1} \big(f_h^k(s_h^i,a_h^i) - (\mathcal{T} f_{h+1}^k)(s_h^i,a_h^i)\big)^2 \le \mathcal{O}(\beta)$,

where $(s_1^i, a_1^i, \ldots, s_H^i, a_H^i, s_{H+1}^i)$ denotes the trajectory sampled by following π^i in the i^{th} episode.

The second lemma guarantees that the optimal value function is inside the confidence with high probability. As a result, the selected value function f^k in each iteration shall be an upper bound of $Q^\star$ with high probability.

Lemma D.5.2. *Under the same condition of Lemma D.5.1, with probability at least $1-\delta$, we have $Q^\star \in \mathcal{B}^k$ for all $k \in [K]$.*

The proof of Lemma D.5.1 and D.5.2 relies on standard martingale concentration (e.g. Freedman's inequality) and can be found in Appendix D.5.3.

Step 1. Bounding the regret by Bellman error By Lemma D.5.2, we can upper bound the cumulative regret by the summation of Bellman error with probability at least $1-\delta$:

$$\sum_{k=1}^{K}\left(V_1^\star(s_1)-V_1^{\pi^k}(s_1)\right)\le\sum_{k=1}^{K}\left(\max_a f_1^k(s_1,a)-V_1^{\pi^k}(s_1)\right)\overset{(i)}{=}\sum_{k=1}^{K}\sum_{h=1}^{H}\mathcal{E}(f^k,\pi^k,h), \tag{D.2}$$

where (i) follows from standard policy loss decomposition (e.g. Lemma 1 in Jiang et al. (2017)).

Step 2. Bounding cumulative Bellman error using DE dimension Next, we focus on a fixed step h and bound the cumulative Bellman error $\sum_{k=1}^{K}\mathcal{E}(f^k,\pi^k,h)$ using Lemma D.5.1. To proceed, we need the following lemma to control the accumulating rate of Bellman error.

Lemma D.5.3. *Given a function class Φ defined on $\mathcal{X}$ with $|\phi(x)|\le C$ for all $(g,x)\in\Phi\times\mathcal{X}$, and a family of probability measures Π over $\mathcal{X}$. Suppose sequence $\{\phi_k\}_{k=1}^K\subset\Phi$ and $\{\mu_k\}_{k=1}^K\subset\Pi$ satisfy that for all $k\in[K]$, $\sum_{t=1}^{k-1}(\mathbb{E}_{\mu_t}[\phi_k])^2\le\beta$. Then for all $k\in[K]$ and $\omega>0$,*

$$\sum_{t=1}^{k}|\mathbb{E}_{\mu_t}[\phi_t]|\le\mathcal{O}\left(\sqrt{\dim_{\mathrm{DE}}(\Phi,\Pi,\omega)\beta k}+\min\{k,\dim_{\mathrm{DE}}(\Phi,\Pi,\omega)\}C+k\omega\right).$$

Lemma D.5.3 is a simple modification of Lemma 2 in Russo and Van Roy (2013) and its proof can be found in Appendix D.5.4. We provide two ways to apply Lemma D.5.3, which can produce regret bounds in term of two different complexity measures. If we invoke Lemma D.5.1 (a) and Lemma D.5.3 with

$$\begin{cases}\rho=\dfrac{1}{K},\ \omega=\sqrt{\dfrac{1}{K}},\ C=1,\\ \mathcal{X}=\mathcal{S}\times\mathcal{A},\ \Phi=(I-\mathcal{T}_h)\mathcal{F},\ \Pi=\mathcal{D}_{\mathcal{F},h},\\ \phi_k=f_h^k-\mathcal{T}_hf_{h+1}^k\text{ and }\mu_k=\mathbb{P}^{\pi^k}(s_h=\cdot,a_h=\cdot),\end{cases}$$

we obtain

$$\sum_{t=1}^{k}\mathcal{E}(f^t,\pi^t,h)\le\mathcal{O}\left(\sqrt{k\cdot\dim_{\mathrm{BE}}(\mathcal{F},\mathcal{D}_{\mathcal{F}},\sqrt{1/K})\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(1/K)/\delta]}\right). \tag{D.3}$$

We can also invoke Lemma D.5.1 (b) and Lemma D.5.3 with

$$\begin{cases} \rho = \frac{1}{K},\ \omega = \sqrt{\frac{1}{K}},\ C = 1, \\ \mathcal{X} = \mathcal{S} \times \mathcal{A},\ \Phi = (I - \mathcal{T}_h)\mathcal{F},\ \text{and } \Pi = \mathcal{D}_{\Delta,h}, \\ \phi_k = f_h^k - \mathcal{T}_h f_{h+1}^k \text{ and } \mu_k = \mathbf{1}\{\cdot = (s_h^k, a_h^k)\}, \end{cases}$$

and obtain

$$\begin{aligned} \sum_{t=1}^{k} \mathcal{E}(f^t, \pi^t, h) \leq & \sum_{t=1}^{k} (f_h^t - \mathcal{T} f_{h+1}^t)(s_h^t, a_h^t) + \mathcal{O}\left(\sqrt{k \log(k)}\right) \\ \leq & \mathcal{O}\left(\sqrt{k \cdot \dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_\Delta, \sqrt{1/K}) \log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(1/K)/\delta]}\right), \end{aligned} \tag{D.4}$$

where the first inequality follows from standard martingale concentration.

Plugging either equation Eq. (D.3) or Eq. (D.4) back into equation Eq. (D.2) completes the proof.

D.5.2 Proof of Corollary 5.4.3

Step 1. Bounding the regret by Bellman error By Lemma D.5.2, we can upper bound the cumulative regret by the summation of Bellman error with probability at least $1 - \delta$:

$$\sum_{k=1}^{K} \left(V_1^\star(s_1) - V_1^{\pi^k}(s_1)\right) \leq \sum_{k=1}^{K} \left(\max_a f_1^k(s_1, a) - V_1^{\pi^k}(s_1)\right) \overset{(i)}{=} \sum_{k=1}^{K} \sum_{h=1}^{H} \mathcal{E}(f^k, \pi^k, h), \tag{D.5}$$

where (i) follows from standard policy loss decomposition (e.g. Lemma 1 in Jiang et al. (2017)).

Step 2. Bounding cumulative Bellman error using DE dimension Next, we focus on a fixed step h and bound the cumulative Bellman error $\sum_{k=1}^{K} \mathcal{E}(f^k, \pi^k, h)$ using Lemma D.5.1.

If we invoke Lemma D.5.1 (a) with

$$\rho = \frac{\epsilon^2}{H^2 \cdot \dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_\mathcal{F}, \epsilon/H)},$$

and Lemma D.5.3 with

$$\begin{cases} \omega = \dfrac{\epsilon}{H},\ C = 1, \\ \mathcal{X} = \mathcal{S} \times \mathcal{A},\ \Phi = (I - \mathcal{T}_h)\mathcal{F},\ \Pi = \mathcal{D}_{\mathcal{F},h}, \\ \phi_k = f_h^k - \mathcal{T}_h f_{h+1}^k \text{ and } \mu_k = \mathbb{P}^{\pi^k}(s_h = \cdot, a_h = \cdot), \end{cases}$$

we obtain with probability at least $1 - 10^{-3}$,

$$\begin{aligned} \frac{1}{K}\sum_{k=1}^{K} \mathcal{E}(f^k, \pi^k, h) \leq & \mathcal{O}\left(\sqrt{\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon/H)[\frac{\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)]}{K} + \rho]} + \frac{\epsilon}{H}\right) \\ \leq & \mathcal{O}\left(\frac{\epsilon}{H} + \sqrt{\frac{d\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)]}{K}}\right), \end{aligned} \tag{D.6}$$

where the second inequality follows from the choice of ρ and $d := \dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon/H)$. Now we need to choose K such that

$$\sqrt{\frac{d\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)]}{K}} \leq \frac{\epsilon}{H}. \tag{D.7}$$

By simple calculation, one can verify it suffices to choose

$$K = \frac{H^2 d\log(Hd\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)/\epsilon)}{\epsilon^2}. \tag{D.8}$$

Plugging equation Eq. (D.6) back into equation Eq. (D.5) completes the proof. We can similarly prove the bound in terms of the BE dimension with respect to $\mathcal{D}_{\Delta}$.

D.5.3 Proofs of concentration lemmas

To begin with, recall the Freedman's inequality that controls the sum of martingale difference by the sum of their predicted variance.

Lemma D.5.4 (Freedman's inequality (e.g., Agarwal et al., 2014)). *Let $(Z_t)_{t\leq T}$ be a real-valued martingale difference sequence adapted to filtration $\mathfrak{F}_t$, and let $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathfrak{F}_t]$. If $|Z_t| \leq R$ almost*

surely, then for any $\eta \in (0, \frac{1}{R})$ *it holds that with probability at least* $1-\delta$,

$$\sum_{t=1}^{T} Z_t \leq \mathcal{O}\left(\eta \sum_{t=1}^{T} \mathbb{E}_{t-1}[Z_t^2] + \frac{\log(\delta^{-1})}{\eta}\right).$$

Proof of Lemma D.5.1

Proof. We prove inequality (b) first.

Consider a fixed (k, h, f) tuple. Let

$$X_t(h,f) := (f_h(s_h^t, a_h^t) - r_h^t - f_{h+1}(s_{h+1}^t, \pi_f(s_{h+1}^t)))^2 - ((\mathcal{T} f_{h+1})(s_h^t, a_h^t) - r_h^t - f_{h+1}(s_{h+1}^t, \pi_f(s_{h+1}^t)))^2$$

and $\mathfrak{F}_{t,h}$ be the filtration induced by $\{s_1^i, a_1^i, r_1^i, \ldots, s_H^i\}_{i=1}^{t-1} \bigcup \{s_1^t, a_1^t, r_1^t, \ldots, s_h^t, a_h^t\}$. We have

$$\mathbb{E}[X_t(h,f) \mid \mathfrak{F}_{t,h}] = [(f_h - \mathcal{T} f_{h+1})(s_h^t, a_h^t)]^2$$

and

$$\mathrm{Var}[X_t(h,f) \mid \mathfrak{F}_{t,h}] \leq \mathbb{E}[(X_t(h,f))^2 \mid \mathfrak{F}_{t,h}] \leq 36[(f_h - \mathcal{T} f_{h+1})(s_h^t, a_h^t)]^2 = 36\mathbb{E}[X_t(h,f) \mid \mathfrak{F}_{t,h}].$$

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

$$\left|\sum_{t=1}^{k} X_t(h,f) - \sum_{t=1}^{k} \mathbb{E}[X_t(h,f) \mid \mathfrak{F}_{t,h}]\right| \leq \mathcal{O}\left(\sqrt{\log(1/\delta) \sum_{t=1}^{k} \mathbb{E}[X_t \mid \mathfrak{F}_{t,h}]} + \log(1/\delta)\right).$$

Let $\mathcal{Z}_\rho$ be a ρ-cover of $\mathcal{F}$. Now taking a union bound for all $(k, h, \phi) \in [K] \times [H] \times \mathcal{Z}_\rho$, we obtain that with probability at least $1-\delta$, for all $(k, h, \phi) \in [K] \times [H] \times \mathcal{Z}_\rho$

$$\left|\sum_{t=1}^{k} X_t(h,\phi) - \sum_{t=1}^{k} [(\phi_h - \mathcal{T}\phi_{h+1})(s_h^t, a_h^t)]^2\right| \leq \mathcal{O}\left(\sqrt{\iota \sum_{t=1}^{k} [(\phi_h - \mathcal{T}\phi_{h+1})(s_h^t, a_h^t)]^2} + \iota\right), \quad \text{(D.9)}$$

where $\iota = \log(HK|\mathcal{Z}_\rho|/\delta)$. From now on, we will do all the analysis conditioning on this event being

true.

Consider an arbitrary $(h,k) \in [H] \times [K]$ pair. By the definition of $\mathcal{B}^k$ and Assumption 5.4.1

$$\begin{aligned}
\sum_{t=1}^{k-1} X_t(h, f^k) =& \sum_{t=1}^{k-1} [f_h^k(s_h^t, a_h^t) - r_h^t - f_{h+1}^k(s_{h+1}^t, \pi_{f^k}(s_{h+1}^t))]^2 \\
& - \sum_{t=1}^{k-1} [(\mathcal{T} f_{h+1}^k)(s_h^t, a_h^t) - r_h^t - f_{h+1}^k(s_{h+1}^t, \pi_{f^k}(s_{h+1}^t))]^2 \\
\leq & \sum_{t=1}^{k-1} [f_h^k(s_h^t, a_h^t) - r_h^t - f_{h+1}^k(s_{h+1}^t, \pi_{f^k}(s_{h+1}^t))]^2 \\
& - \inf_{g \in \mathcal{G}} \sum_{t=1}^{k-1} [g_h(s_h^t, a_h^t) - r_h^t - f_{h+1}^k(s_{h+1}^t, \pi_{f^k}(s_{h+1}^t))]^2 \leq \beta.
\end{aligned}$$

Define $\phi^k = \operatorname{argmin}_{\phi \in \mathcal{Z}_\rho} \max_{h \in [H]} \|f_h^k - \phi_h^k\|_\infty$. By the definition of $\mathcal{Z}_\rho$, we have

$$\left| \sum_{t=1}^{k-1} X_t(h, f^k) - \sum_{t=1}^{k-1} X_t(h, \phi^k) \right| \leq \mathcal{O}(k\rho).$$

Therefore,

$$\sum_{t=1}^{k-1} X_t(h, \phi^k) \leq \mathcal{O}(k\rho) + \beta. \tag{D.10}$$

Recall inequality Eq. (D.9) implies

$$\left| \sum_{t=1}^{k-1} X_t(h, \phi^k) - \sum_{t=1}^{k-1} [(\phi_h^k - \mathcal{T}\phi_{h+1}^k)(s_h^t, a_h^t)]^2 \right| \leq \mathcal{O}\left(\sqrt{\iota \sum_{t=1}^{k-1} [(\phi_h^k - \mathcal{T}\phi_{h+1}^k)(s_h^t, a_h^t)]^2 + \iota} \right). \tag{D.11}$$

Putting Eq. (D.10) and Eq. (D.11) together, we obtain

$$\sum_{t=1}^{k-1} [(\phi_h^k - \mathcal{T}\phi_{h+1}^k)(s_h^t, a_h^t)]^2 \leq \mathcal{O}(\iota + k\rho + \beta).$$

Because ϕ^k is an ρ-approximation to f^k, we conclude

$$\sum_{t=1}^{k-1}[(f_h^k - \mathcal{T} f_{h+1}^k)(s_h^t, a_h^t)]^2 \leq \mathcal{O}(\iota + k\rho + \beta).$$

Therefore, we prove inequality (b) in Lemma D.5.1.

To prove inequality (a), we only need to redefine $\mathfrak{F}_{t,h}$ to be the filtration induced by $\{s_1^i, a_1^i, r_1^i, \ldots, s_H^i\}_{i=1}^{t-1}$ and then repeat the arguments above verbatim. □

Proof of Lemma D.5.2

Proof. Let $\mathcal{V}_\rho$ be a ρ-cover of $\mathcal{G}$.

Consider an arbitrary fixed tuple $(k, h, g) \in [K] \times [H] \times \mathcal{G}$. Let

$$W_t(h,g) := (g_h(s_h^t, a_h^t) - r_h^t - Q_{h+1}^\star(s_{h+1}^t, \pi_{Q^\star}(s_{h+1}^t)))^2 - (Q_h^\star(s_h^t, a_h^t) - r_h^t - Q_{h+1}^\star(s_{h+1}^t, \pi_{Q^\star}(s_{h+1}^t)))^2$$

and $\mathfrak{F}_{t,h}$ be the filtration induced by $\{s_1^i, a_1^i, r_1^i, \ldots, s_H^i\}_{i=1}^{t-1} \bigcup \{s_1^t, a_1^t, r_1^t, \ldots, s_h^t, a_h^t\}$. We have

$$\mathbb{E}[W_t(h,g) \mid \mathfrak{F}_{t,h}] = [(g_h - Q_h^\star)(s_h^t, a_h^t)]^2$$

and

$$\mathrm{Var}[W_t(h,g) \mid \mathfrak{F}_{t,h}] \leq \mathbb{E}[(W_t(h,g))^2 \mid \mathfrak{F}_{t,h}] \leq 36((g_h - Q_h^\star)(s_h^t, a_h^t))^2 = 36\mathbb{E}[W_t(h,g) \mid \mathfrak{F}_{t,h}].$$

By Freedman's inequality, with probability at least $1 - \delta$,

$$\left|\sum_{t=1}^{k} W_t(h,g) - \sum_{t=1}^{k}[(g_h - Q_h^\star)(s_h^t, a_h^t)]^2\right| \leq \mathcal{O}\left(\sqrt{\log(1/\delta)\sum_{t=1}^{k}[(g_h - Q_h^\star)(s_h^t, a_h^t)]^2} + \log(1/\delta)\right).$$

By taking a union bound over $[K] \times [H] \times \mathcal{V}_\rho$ and the non-negativity of $\sum_{t=1}^{k}[(g_h - Q_h^\star)(s_h^t, a_h^t)]^2$,

we obtain that with probability at least $1-\delta$, for all $(k,h,\psi) \in [K] \times [H] \times \mathcal{V}_\rho$

$$-\sum_{t=1}^{k} W_t(h,\psi) \le \mathcal{O}(\iota),$$

where $\iota = \log(HK|\mathcal{V}_\rho|/\delta)$. This directly implies for all $(k,h,g) \in [K] \times [H] \times \mathcal{G}$

$$\begin{aligned}
&\sum_{t=1}^{k-1} [Q_h^\star(s_h^t, a_h^t) - r_h^t - Q_{h+1}^\star(s_{h+1}^t, \pi_{Q^\star}(s_{h+1}^t))]^2 \\
\le &\sum_{t=1}^{k-1} [g_h(s_h^t, a_h^t) - r_h^t - Q_{h+1}^\star(s_{h+1}^t, \pi_{Q^\star}(s_{h+1}^t))]^2 + \mathcal{O}(\iota + k\rho).
\end{aligned}$$

Finally, by recalling the definition of $\mathcal{B}^k$, we conclude that with probability at least $1-\delta$, $Q^\star \in \mathcal{B}^k$ for all $k \in [K]$. □

D.5.4 Proof of Lemma D.5.3

The proof in this subsection basically follows the same arguments as in Appendix C of Russo and Van Roy (2013). We firstly prove the following proposition which bounds the number of times $|\mathbb{E}_{\mu_t}[\phi_t]|$ can exceed a certain threshold.

Proposition D.5.5. *Given a function class Φ defined on $\mathcal{X}$, and a family of probability measures Π over $\mathcal{X}$. Suppose sequence $\{\phi_k\}_{k=1}^K \subset \Phi$ and $\{\mu_k\}_{k=1}^K \subset \Pi$ satisfy that for all $k \in [K]$, $\sum_{t=1}^{k-1} (\mathbb{E}_{\mu_t}[\phi_k])^2 \le \beta$. Then for all $k \in [K]$,*

$$\sum_{t=1}^{k} \mathbf{1}\big\{|\mathbb{E}_{\mu_t}[\phi_t]| > \epsilon\big\} \le \big(\frac{\beta}{\epsilon^2} + 1\big) \dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon).$$

Proof of Proposition D.5.5. We first show that if for some k we have $|\mathbb{E}_{\mu_k}[\phi_k]| > \epsilon$, then μ_k is ϵ-dependent on at most β/ϵ^2 disjoint subsequences in $\{\mu_1, \ldots, \mu_{k-1}\}$. By definition of DE dimension, if $|\mathbb{E}_{\mu_k}[\phi_k]| > \epsilon$ and μ_k is ϵ-dependent on a subsequence $\{\nu_1, \ldots, \nu_\ell\}$ of $\{\mu_1, \ldots, \mu_{k-1}\}$, then we

should have $\sum_{t=1}^{\ell}(\mathbb{E}_{\nu_t}[\phi_k])^2 \geq \epsilon^2$. It implies that if μ_k is ϵ-dependent on L disjoint subsequences in $\{\mu_1, \ldots, \mu_{k-1}\}$, we have

$$\beta \geq \sum_{t=1}^{k-1}(\mathbb{E}_{\mu_t}[\phi_k])^2 \geq L\epsilon^2$$

resulting in $L \leq \beta/\epsilon^2$.

Now we want to show that for any sequence $\{\nu_1, \ldots, \nu_\kappa\} \subseteq \Pi$, there exists $j \in [\kappa]$ such that ν_j is ϵ-dependent on at least $L = \lceil (\kappa - 1)/\dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon) \rceil$ disjoint subsequences in $\{\nu_1, \ldots, \nu_{j-1}\}$. We argue by the following mental procedure: we start with singleton sequences $B_1 = \{\nu_1\}, \ldots, B_L = \{\nu_L\}$ and $j = L+1$. For each j, if ν_j is ϵ-dependent on $B_1, \ldots, B_L$ we already achieved our goal so we stop; otherwise, we pick an $i \in [L]$ such that ν_j is ϵ-independent of B_i and update $B_i = B_i \cup \{\nu_j\}$. Then we increment j by 1 and continue this process. By the definition of DE dimension, the size of each $B_1, \ldots, B_L$ cannot get bigger than $\dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon)$ at any point in this process. Therefore, the process stops before or on $j = L \dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon) + 1 \leq \kappa$.

Fix $k \in [K]$ and let $\{\nu_1, \ldots, \nu_\kappa\}$ be subsequence of $\{\mu_1, \ldots, \mu_k\}$, consisting of elements for which $|\mathbb{E}_{\mu_t}[\phi_t]| > \epsilon$. Using the first claim, we know that each ν_j is ϵ-dependent on at most β/ϵ^2 disjoint subsequences of $\{\nu_1, \ldots, \nu_{j-1}\}$. Using the second claim, we know there exists $j \in [\kappa]$ such that ν_j is ϵ-dependent on at least $(\kappa/\dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon)) - 1$ disjoint subsequences of $\{\nu_1, \ldots, \nu_{j-1}\}$. Therefore, we have $\kappa/\dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon) - 1 \leq \beta/\epsilon^2$ which results in

$$\kappa \leq (\frac{\beta}{\epsilon^2} + 1)\dim_{\mathrm{DE}}(\Phi, \Pi, \epsilon)$$

and completes the proof. □

Proof of Lemma D.5.3. Fix $k \in [K]$; let $d = \dim_{\mathrm{DE}}(\Phi, \Pi, \omega)$. Sort the sequence $\{|\mathbb{E}_{\phi_1}[\phi_1]|, \ldots, |\mathbb{E}_{\mu_k}[\phi_k]|\}$ in a decreasing order and denote it by $\{e_1, \ldots, e_k\}$ $(e_1 \geq e_2 \geq \cdots \geq e_k)$.

$$\sum_{t=1}^{k}|\mathbb{E}_{\mu_t}[\phi_t]| = \sum_{t=1}^{k} e_t = \sum_{t=1}^{k} e_t \mathbf{1}\{e_t \leq \omega\} + \sum_{t=1}^{k} e_t \mathbf{1}\{e_t > \omega\} \leq k\omega + \sum_{t=1}^{k} e_t \mathbf{1}\{e_t > \omega\}.$$

For $t \in [k]$, we want to prove that if $e_t > \omega$, then we have $e_t \le \min\{\sqrt{\frac{d\beta}{t-d}}, C\}$. Assume $t \in [k]$ satisfies $e_t > \omega$. Then there exists α such that $e_t > \alpha \ge \omega$. By Proposition D.5.5, we have

$$t \le \sum_{i=1}^{k} \mathbf{1}\{e_i > \alpha\} \le \big(\frac{\beta}{\alpha^2} + 1\big) \dim_{\mathrm{DE}}(\Phi, \Pi, \alpha) \le \big(\frac{\beta}{\alpha^2} + 1\big) \dim_{\mathrm{DE}}(\Phi, \Pi, \omega),$$

which implies $\alpha \le \sqrt{\frac{d\beta}{t-d}}$. Besides, recall $e_t \le C$, so we have $e_t \le \min\{\sqrt{\frac{d\beta}{t-d}}, C\}$.

Finally, we have

$$\begin{aligned}\sum_{t=1}^{k} e_t \mathbf{1}\{e_t > \omega\} &\le \min\{d,k\}C + \sum_{t=d+1}^{k} \sqrt{\frac{d\beta}{t-d}} \le \min\{d,k\}C + \sqrt{d\beta}\int_0^k \frac{1}{\sqrt{t}} dt \\ &\le \min\{d,k\}C + 2\sqrt{d\beta k},\end{aligned}$$

which completes the proof. □

D.6 Proofs for Olive

In this section, we provide the formal proof for the results stated in Appendix D.1.

D.6.1 Full proof of Theorem D.1.1

Proof of Theorem D.1.1. By standard concentration arguments (Hoeffding's inequality plus union bound argument), with probability at least $1-\delta$, the following events hold for the first $dH+1$ phases (please refer to Appendix D.6.2 for the proof)

1. If the elimination procedure is activated at the h^{th} step in the k^{th} phase, then $\mathcal{E}(f^k, \pi^k, h) > \zeta_{\text{act}}/2$ and all $f \in \mathcal{F}$ satisfying $|\mathcal{E}(f, \pi^k, h)| \ge 2\zeta_{\text{elim}}$ get eliminated.
2. If the elimination procedure is not activated in the k^{th} phase, then, $\sum_{h=1}^{H} \mathcal{E}(f^k, \pi^k, h) < 2H\zeta_{\text{act}} = 4\epsilon$.
3. $Q^\star$ is not eliminated.

Therefore, if we can show OLIVE terminates within $dH+1$ phases, then with high probability the output policy is 4ϵ-optimal by the optimism of f^k and simple policy loss decomposition (e.g. Lemma 1 in Jiang et al. (2017)):

$$\left(V_1^\star(s_1) - V_1^{\pi^k}(s_1)\right) \le \max_a f^k(s_1,a) - V^{\pi^k}(s_1) = \sum_{h=1}^H \mathcal{E}(f^k,\pi^k,h) \le 4\epsilon. \tag{D.12}$$

In order to prove that OLIVE terminates within $dH+1$ phases, it suffices to show that for each $h\in[H]$, we can activate the elimination procedure at the h^{th} step for at most d times.

For the sake of contradiction, assume that OLIVE does not terminate in $dH+1$ phases. Within these $dH+1$ phases, there exists some $h\in[H]$ for which the activation process has been activated for at least $d+1$ times. Denote by $k_1<\cdots<k_{d+1}\le dH+1$ the indices of the phases where the elimination is activated at the h^{th} step. By the high-probability events, for all $i<j\le d+1$, we have $|\mathcal{E}(f^{k_j},\pi^{k_i},h)|<2\zeta_{\text{elim}}$ and for all $l\le d+1$, we have $\mathcal{E}(f^{k_l},\pi^{k_l},h)>\zeta_{\text{act}}/2$. This means for all $l\le d+1$, we have both $\sqrt{\sum_{i=1}^{l-1}\left(\mathcal{E}(f^{k_l},\pi^{k_i},h)\right)^2} < \sqrt{d}\times 2\zeta_{\text{elim}} = \epsilon/H$ and $\mathcal{E}(f^{k_l},\pi^{k_l},h)>\zeta_{\text{act}}/2=\epsilon/H$. Therefore, the roll-in distribution of $\pi^{k_1},\dots,\pi^{k_{d+1}}$ at step h is an ϵ/H-independent sequence of length $d+1$, which contradicts with the definition of BE dimension. So OLIVE should terminate within $dH+1$ phases.

In sum, with probability at least $1-\delta$, Algorithm 18 will terminate and output a 4ϵ-optimal policy using at most

$$(dH+1)(n_{\text{act}}+n_{\text{elim}}) \le \frac{3cH^3d^2\log(\mathcal{N}(\mathcal{F},\zeta_{\text{elim}}/8))\cdot\iota}{\epsilon^2}$$

episodes. □

D.6.2 Concentration arguments for Theorem D.1.1

Recall in Algorithm 18 we choose

$$\zeta_{\text{act}} = \frac{2\epsilon}{H},\ \zeta_{\text{elim}} = \frac{\epsilon}{2H\sqrt{d}},\ n_{\text{act}} = \frac{cH^2\iota}{\epsilon^2},\ \text{and } n_{\text{elim}} = \frac{cH^2d\log(\mathcal{N}(\mathcal{F},\zeta_{\text{elim}}/8))\cdot\iota}{\epsilon^2},$$

where $d = \max_{h\in[H]} \dim_{\mathrm{BE}}\big(\mathcal{F}, \mathcal{D}_{\mathcal{F},h}, \epsilon/H\big)$, $\iota = \log[Hd/\delta\epsilon]$ and c is a large absolute constant.Our goal is to prove with probability at least $1-\delta$, the following events hold for the first $dH+1$ phases

1. If the elimination procedure is activated at the h^{th} step in the k^{th} phase, then $\mathcal{E}(f^k, \pi^k, h) > \zeta_{\mathrm{act}}/2$ and all $f \in \mathcal{F}$ satisfying $|\mathcal{E}(f, \pi^k, h)| \geq 2\zeta_{\mathrm{elim}}$ get eliminated.
2. If the elimination procedure is not activated in the k^{th} phase, then, $\sum_{h=1}^{H} \mathcal{E}(f^k, \pi^k, h) < 2H\zeta_{\mathrm{act}} = 4\epsilon$.
3. $Q^\star$ is not eliminated.

We begin with the activation procedure.

Concentration in the activation procedure Consider a fixed $(k,h) \in [dH+1]\times[H]$ pair. By Azuma-Hoefdding's inequality, with probability at least $1 - \frac{\delta}{8H(dH^2+1)}$, we have

$$|\hat{\mathcal{E}}(f^k, \pi^k, h) - \mathcal{E}(f^k, \pi^k, h)| \leq \mathcal{O}\left(\sqrt{\frac{\iota}{n_{\mathrm{act}}}}\right) \leq \frac{\epsilon}{2H} \leq \zeta_{\mathrm{act}}/4,$$

where the second inequality follows from $n_{\mathrm{act}} = C\frac{H^2\iota}{\epsilon^2}$ with C being chosen large enough.

Take a union bound for all $(k,h) \in [dH+1]\times[H]$, we have with probability at least $1-\delta/4$, the following holds for all $(k,h) \in [dH+1]\times[H]$

$$|\hat{\mathcal{E}}(f^k, \pi^k, h) - \mathcal{E}(f^k, \pi^k, h)| \leq \zeta_{\mathrm{act}}/4.$$

By Algorithm 18, if the elimination procedure is not activated in the k^{th} phase, we have $\sum_{h=1}^{H} \hat{\mathcal{E}}(f^k, \pi^k, h) \leq H\zeta_{\mathrm{act}}$. Combine it with the concentration argument we just proved,

$$\sum_{h=1}^{H} \mathcal{E}(f^k, \pi^k, h) \leq \sum_{h=1}^{H} \hat{\mathcal{E}}(f^k, \pi^k, h) + \frac{H\zeta_{\mathrm{act}}}{4} < \frac{5H\zeta_{\mathrm{act}}}{4}.$$

On the other hand, if the elimination procedure is activated at the h^{th} step in the k^{th} phase, then

$\hat{\mathcal{E}}(f^k, \pi^k, h) > \zeta_{\text{act}}$. Again combine it with the concentration argument we just proved,

$$\mathcal{E}(f^k, \pi^k, h) \geq \hat{\mathcal{E}}(f^k, \pi^k, h) - \frac{\zeta_{\text{act}}}{4} > \frac{3\zeta_{\text{act}}}{4}.$$

Concentration in the elimination procedure Now, let us turn to the elimination procedure. First, let $\mathcal{Z}$ be an $\zeta_{\text{elim}}/8$-cover of $\mathcal{F}$ with cardinality $\mathcal{N}(\mathcal{F}, \zeta_{\text{elim}}/8)$. With a little abuse of notation, for every $f \in \mathcal{F}$, define $\hat{f} = \text{argmin}_{g \in \mathcal{Z}} \max_{h \in [H]} \|f_h - g_h\|_\infty$. By applying Azuma-Hoeffding's inequality to all $(k, g) \in [dH + 1] \times \mathcal{Z}$ and taking a union bound, we have with probability at least $1 - \delta/4$, the following holds for all $(k, g) \in [dH + 1] \times \mathcal{Z}$

$$|\hat{\mathcal{E}}(g, \pi^k, h_k) - \mathcal{E}(g, \pi^k, h_k)| \leq \zeta_{\text{elim}}/4.$$

Recall that Algorithm 18 eliminates all f satisfying $|\hat{\mathcal{E}}(f, \pi^k, h_k)| > \zeta_{\text{elim}}$ when the elimination procedure is activated at the h_k^{th} step in the k^{th} phase. Therefore, if $|\mathcal{E}(f, \pi^k, h_k)| \geq 2\zeta_{\text{elim}}$, f will be eliminated because

$$\begin{aligned}
|\hat{\mathcal{E}}(f, \pi^k, h_k)| &\geq |\hat{\mathcal{E}}(\hat{f}, \pi^k, h_k)| - 2 \times \frac{\zeta_{\text{elim}}}{8} \\
&\geq |\mathcal{E}(\hat{f}, \pi^k, h_k)| - \frac{\zeta_{\text{elim}}}{2} \\
&\geq |\mathcal{E}(f, \pi^k, h_k)| - \frac{\zeta_{\text{elim}}}{2} - 2 \times \frac{\zeta_{\text{elim}}}{8} > \zeta_{\text{elim}}.
\end{aligned}$$

Finally, note that $\mathcal{E}(Q^\star, \pi, h) \equiv 0$ for any π and h. As a result, it will never be eliminated within the first $dH + 1$ phases because we can similarly prove

$$|\hat{\mathcal{E}}(Q^\star, \pi^k, h_k)| \leq |\mathcal{E}(Q^\star, \pi^k, h_k)| + \frac{3\zeta_{\text{elim}}}{4} < \zeta_{\text{elim}}.$$

Wrapping up: take a union bound for the activation and elimination procedure, and conclude that the three events, listed at the beginning of this section, hold for the the first $dH + 1$ phases with probability at least $1 - \delta/2$.

D.7 Proofs for V-type Variants

In this section, we provide formal proofs for the results stated in Section D.2.

D.7.1 Proof of Theorem D.2.5

The proof is similar to that in Appendix D.6.

Proof of Theorem D.2.5. By standard concentration arguments (Hoeffding's inequality, Bernstein's inequality, and union bound argument), with probability at least $1-\delta$, the following events hold for the first $dH+1$ phases (please refer to Appendix D.7.1 for the proof)

1. If the elimination procedure is activated at the h^{th} step in the k^{th} phase, then $\mathcal{E}_{\text{V}}(f^k, \pi^k, h) > \zeta_{\text{act}}/2$ and all $f \in \mathcal{F}$ satisfying $|\mathcal{E}_{\text{V}}(f, \pi^k, h)| \geq 2\zeta_{\text{elim}}$ get eliminated.
2. If the elimination procedure is not activated in the k^{th} phase, then, $\sum_{h=1}^{H} \mathcal{E}_{\text{V}}(f^k, \pi^k, h) < 2H\zeta_{\text{act}} = 4\epsilon$.
3. $Q^\star$ is not eliminated.

Therefore, if we can show OLIVE terminates within $dH+1$ phases, then with high probability the output policy is 4ϵ-optimal by the optimism of f^k and simple policy loss decomposition (e.g., Lemma 1 in Jiang et al. (2017)):

$$\left(V_1^\star(s_1) - V_1^{\pi^k}(s_1)\right) \leq \max_a f^k(s_1, a) - V^{\pi^k}(s_1) = \sum_{h=1}^{H} \mathcal{E}_{\text{V}}(f^k, \pi^k, h) \leq 4\epsilon. \tag{D.13}$$

In order to prove that OLIVE terminates within $dH+1$ phases, it suffices to show that for each $h \in [H]$, we can activate the elimination procedure at the h^{th} step for at most d times.

For the sake of contradiction, assume that OLIVE does not terminate in $dH+1$ phases. Within these $dH+1$ phases, there exists some $h \in [H]$ for which the activation process has been activated for at least $d+1$ times. Denote by $k_1 < \cdots < k_{d+1} \leq dH+1$ the indices of the phases where the elimination is activated at the h^{th} step. By the high-probability events, for all $i < j \leq d+1$, we

have $|\mathcal{E}_{\mathrm{V}}(f^{k_j}, \pi^{k_i}, h)| < 2\zeta_{\mathrm{elim}}$ and for all $l \le d+1$, we have $\mathcal{E}_{\mathrm{V}}(f^{k_l}, \pi^{k_l}, h) > \zeta_{\mathrm{act}}/2$. This means for all $l \le d+1$, we have both $\sqrt{\sum_{i=1}^{l-1} \left(\mathcal{E}_{\mathrm{V}}(f^{k_l}, \pi^{k_i}, h)\right)^2} < \sqrt{d} \times 2\zeta_{\mathrm{elim}} = \epsilon/H$ and $\mathcal{E}_{\mathrm{V}}(f^{k_l}, \pi^{k_l}, h) > \zeta_{\mathrm{act}}/2 = \epsilon/H$. Therefore, the roll-in distribution of $\pi^{k_1}, \ldots, \pi^{k_{d+1}}$ at step h is an ϵ/H-independent sequence of length $d+1$ with respect to $(I - \mathcal{T}_h)V_{\mathcal{F}}$, which contradicts with the definition of BE dimension. So OLIVE should terminate within $dH+1$ phases.

In sum, with probability at least $1-\delta$, Algorithm 18 will terminate and output a 4ϵ-optimal policy using at most

$$(dH+1)(n_{\mathrm{act}} + n_{\mathrm{elim}}) \le \frac{3cH^3d^2|\mathcal{A}|\log(|\mathcal{F}|)\cdot\iota}{\epsilon^2}$$

episodes. □

Concentration arguments for Theorem D.2.5

Recall in Algorithm 20 we choose

$$\zeta_{\mathrm{act}} = \frac{2\epsilon}{H},\ \zeta_{\mathrm{elim}} = \frac{\epsilon}{2H\sqrt{d}},\ n_{\mathrm{act}} = \frac{cH^2\iota}{\epsilon^2},\ \text{and } n_{\mathrm{elim}} = \frac{c|\mathcal{A}|H^2d\log(\mathcal{N}(\mathcal{F}, \zeta_{\mathrm{elim}}/8))\cdot\iota}{\epsilon^2},$$

where $d = \max_{h\in[H]} \dim_{\mathrm{VBE}}\big(\mathcal{F}, \mathcal{D}_{\mathcal{F},h}, \epsilon/H\big)$, $\iota = \log[Hd/\delta\epsilon]$ and c is a large absolute constant. Our goal is to prove with probability at least $1-\delta$, the following events hold for the first $dH+1$ phases

1. If the elimination procedure is activated at the h^{th} step in the k^{th} phase, then $\mathcal{E}_{\mathrm{V}}(f^k, \pi^k, h) > \zeta_{\mathrm{act}}/2$ and all $f \in \mathcal{F}$ satisfying $|\mathcal{E}_{\mathrm{V}}(f, \pi^k, h)| \ge 2\zeta_{\mathrm{elim}}$ get eliminated.
2. If the elimination procedure is not activated in the k^{th} phase, then, $\sum_{h=1}^{H} \mathcal{E}_{\mathrm{V}}(f^k, \pi^k, h) < 2H\zeta_{\mathrm{act}} = 4\epsilon$.
3. $Q^\star$ is not eliminated.

We begin with the activation procedure.

Concentration in the activation procedure Consider a fixed $(k,h) \in [dH+1] \times [H]$ pair. By Azuma-Hoefdding's inequality, with probability at least $1 - \frac{\delta}{8H(dH+1)}$, we have

$$|\tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) - \mathcal{E}_{\mathrm{V}}(f^k, \pi^k, h)| \le \mathcal{O}\left(\sqrt{\frac{\iota}{n_{\mathrm{act}}}}\right) \le \frac{\epsilon}{2H} \le \zeta_{\mathrm{act}}/4,$$

where the second inequality follows from $n_{\mathrm{act}} = C\frac{H^2\iota}{\epsilon^2}$ with C being chosen large enough.

Take a union bound for all $(k,h) \in [dH+1] \times [H]$, we have with probability at least $1-\delta/4$, the following holds for all $(k,h) \in [dH+1] \times [H]$

$$|\tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) - \mathcal{E}_{\mathrm{V}}(f^k, \pi^k, h)| \le \zeta_{\mathrm{act}}/4.$$

By Algorithm 20, if the elimination procedure is not activated in the k^{th} phase, we have $\sum_{h=1}^{H} \tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) \le H\zeta_{\mathrm{act}}$. Combine it with the concentration argument we just proved,

$$\sum_{h=1}^{H} \mathcal{E}_{\mathrm{V}}(f^k, \pi^k, h) \le \sum_{h=1}^{H} \tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) + \frac{H\zeta_{\mathrm{act}}}{4} \le \frac{5H\zeta_{\mathrm{act}}}{4}.$$

On the other hand, if the elimination procedure is activated at the h^{th} step in the k^{th} phase, then $\tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) > \zeta_{\mathrm{act}}$. Again combine it with the concentration argument we just proved,

$$\mathcal{E}_{\mathrm{V}}(f^k, \pi^k, h) \ge \tilde{\mathcal{E}}_{\mathrm{V}}(f^k, \pi^k, h) - \frac{\zeta_{\mathrm{act}}}{4} > \frac{3\zeta_{\mathrm{act}}}{4}.$$

Concentration in the elimination procedure Now, let us turn to the elimination procedure. We start by bounding the the second moment of

$$\frac{\mathbf{1}[\pi_f(s_h) = a_h]}{1/|\mathcal{A}|}\big(f_h(s_h,a_h) - r_h - \max_{a' \in \mathcal{A}} f_{h+1}(s_{h+1}, a')\big)$$

for all $f \in \mathcal{F}$. Let $y(s_h, a_h, r_h, s_{h+1}) = f_h(s_h, a_h) - r_h - \max_{a' \in \mathcal{A}} f_{h+1}(s_{h+1}, a') \in [-2, 1]$, then we have

$$\begin{aligned}
&\mathbb{E}[\big(|\mathcal{A}|\mathbf{1}[\pi_f(s_h) = a_h]y(s_h, a_h, r_h, s_{h+1})\big)^2 \mid s_h \sim \pi^k, a_h \sim \text{Uniform}(\mathcal{A})] \\
\leq & 4|\mathcal{A}|^2\mathbb{E}[\mathbf{1}[\pi_f(s_h) = a_h] \mid s_h \sim \pi^k, a_h \sim \text{Uniform}(\mathcal{A})] = 4|\mathcal{A}|.
\end{aligned}$$

For a fixed $(k, f) \in [dH + 1] \times \mathcal{F}$, by applying Azuma-Bernstein's inequality, with probability at least $1 - \frac{\delta}{8(dH+1)|\mathcal{F}|}$ we have

$$|\hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, h_k) - \mathcal{E}_{\mathrm{V}}(f, \pi^k, h_k)| \leq \mathcal{O}\left(\sqrt{\frac{|\mathcal{A}|\iota'}{n_{\text{elim}}}} + \frac{|\mathcal{A}|\iota'}{n_{\text{elim}}}\right) \leq \mathcal{O}\left(\sqrt{\frac{|\mathcal{A}|\iota'}{n_{\text{elim}}}}\right) \leq \zeta_{\text{elim}}/2,$$

where $\iota' = \log[8(dH + 1)|\mathcal{F}|/\delta]$, and the third inequality follows from $n_{\text{elim}} = C|\mathcal{A}|\iota/\zeta_{\text{elim}}^2$ with C being chosen large enough.

Taking a union bound over $[dH + 1] \times \mathcal{F}$, we have with probability at least $1 - \delta/4$, the following holds for all $(k, f) \in [dH + 1] \times \mathcal{F}$

$$|\hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, h_k) - \mathcal{E}_{\mathrm{V}}(f, \pi^k, h_k)| \leq \zeta_{\text{elim}}/2.$$

Recall that Algorithm 20 eliminates all f satisfying $|\hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, h_k)| > \zeta_{\text{elim}}$ when the elimination procedure is activated at the h_k^{th} step in the k^{th} phase. Therefore, if $|\mathcal{E}_{\mathrm{V}}(f, \pi^k, h_k)| \geq 2\zeta_{\text{elim}}$, f will be eliminated because

$$|\hat{\mathcal{E}}_{\mathrm{V}}(f, \pi^k, h_k)| \geq |\mathcal{E}_{\mathrm{V}}(f, \pi^k, h_k)| - \frac{\zeta_{\text{elim}}}{2} > \zeta_{\text{elim}}.$$

Finally, note that $\mathcal{E}_{\mathrm{V}}(Q^\star, \pi, h) \equiv 0$ for any π and h. As a result, it will never be eliminated within the first $dH + 1$ phases because we can similarly prove

$$|\hat{\mathcal{E}}_{\mathrm{V}}(Q^\star, \pi^k, h_k)| \leq |\mathcal{E}_{\mathrm{V}}(Q^\star, \pi^k, h_k)| + \frac{\zeta_{\text{elim}}}{2} < \zeta_{\text{elim}}.$$

Wrapping up: take a union bound for the activation and elimination procedure, and conclude that

the three events, listed at the beginning of this section, hold for the the first $dH+1$ phases with probability at least $1-\delta/2$.

D.7.2 Proof of Theorem D.2.4

The proof is basically the same as that of Theorem 5.4.2 in Appendix D.5.

To begin with, we have the following lemma (akin to Lemma D.5.1 and D.5.2) showing that with high probability: (i) any function in the confidence set has low Bellman-error over the collected Datasets $\mathcal{D}_1,\ldots,\mathcal{D}_H$ as well as the distributions from which $\mathcal{D}_1,\ldots,\mathcal{D}_H$ are sampled; (ii) the optimal value function is inside the confidence set. Its proof is almost identical to that of Lemma D.5.1 and D.5.2 which can be found in Appendix D.5.3.

Lemma D.7.1 (Akin to Lemma D.5.1 and D.5.2)**.** *Let $\rho>0$ be an arbitrary fixed number. If we choose $\beta=c\big(\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)/\delta]+K\rho\big)$ with some large absolute constant c in Algorithm 19, then with probability at least $1-\delta$, for all $(k,h)\in[K]\times[H]$, we have*

(a) $\sum_{i=1}^{k-1}\mathbb{E}[\big(f_h^k(s_h,a_h)-(\mathcal{T}f_{h+1}^k)(s_h,a_h)\big)^2\mid s_h\sim\pi^i,a_h\sim\mathrm{Uniform}(\mathcal{A})]\leq\mathcal{O}(\beta)$,

(b) $\frac{1}{|\mathcal{A}|}\sum_{i=1}^{k-1}\sum_{a\in\mathcal{A}}\big(f_h^k(s_h^i,a)-(\mathcal{T}f_{h+1}^k)(s_h^i,a)\big)^2\leq\mathcal{O}(\beta)$,

(c) $Q^\star\in\mathcal{B}^k$,

where s_h^i denotes the state at step h collected according to Line 5 in Algorithm 19 following π^i.

Proof of Lemma D.7.1. To prove inequality (a), we only need to redefine the filtration $\mathfrak{F}_{t,h}$ in Appendix D.5.3 to be the filtration induced by $\{s_1^i,a_1^i,r_1^i,\ldots,s_H^i\}_{i=1}^{t-1}$ and repeat the arguments there verbatim.

To prove inequality (b), we only need to redefine the filtration $\mathfrak{F}_{t,h}$ in Appendix D.5.3 to be the filtration induced by $\{s_1^i,a_1^i,r_1^i,\ldots,s_H^i\}_{i=1}^{t-1}\bigcup\{s_1^t,a_1^t,r_1^t,\ldots,s_h^t\}$ and repeat the arguments there verbatim.

The proof of (c) is the same as that of Lemma D.5.2 in Appendix D.5.3. □

Step 1. Bounding the regret by Bellman error By Lemma D.7.1 (c), we can upper bound the cumulative regret by the summation of Bellman error with probability at least $1-\delta$:

$$\sum_{k=1}^{K}\left(V_1^\star(s_1)-V_1^{\pi^k}(s_1)\right)\le\sum_{k=1}^{K}\left(\max_a f_1^k(s_1,a)-V_1^{\pi^k}(s_1)\right)\overset{(i)}{=}\sum_{k=1}^{K}\sum_{h=1}^{H}\mathcal{E}_{\mathrm{V}}(f^k,\pi^k,h), \tag{D.14}$$

where (i) follows from standard policy loss decomposition (e.g. Lemma 1 in Jiang et al. (2017)).

Step 2. Bounding cumulative Bellman error using DE dimension Next, we focus on a fixed step h and bound the cumulative Bellman error $\sum_{k=1}^{K}\mathcal{E}_{\mathrm{V}}(f^k,\pi^k,h)$ using Lemma D.7.1.

Invoking Lemma D.7.1 (a) with

$$\rho=\frac{\epsilon^2}{H^2\cdot\dim_{\mathrm{VBE}}(\mathcal{F},\mathcal{D}_{\mathcal{F}},\epsilon/H)\cdot|\mathcal{A}|}$$

implies that with probability at least $1-\delta$, for all $(k,h)\in[K]\times[H]$, we have

$$\sum_{i=1}^{k-1}\mathbb{E}\left[\left(f_h^k(s_h,\pi_{f_h^k}(s_h))-(\mathcal{T}f_{h+1}^k)(s_h,\pi_{f_h^k}(s_h))\right)^2\mid s_h\sim\pi^i\right]\le\mathcal{O}(|\mathcal{A}|\beta).$$

Further invoking Lemma D.5.3 with

$$\begin{cases}\omega=\dfrac{\epsilon}{H},\ C=1,\\ \mathcal{X}=\mathcal{S},\ \Phi=(I-\mathcal{T}_h)V_{\mathcal{F}},\ \Pi=\mathcal{D}_{\mathcal{F},h},\\ \phi_k(s):=(f_h^k-\mathcal{T}_hf_{h+1}^k)(s,\pi_{f_h^k}(s))\text{ and }\mu_k=\mathbb{P}^{\pi^k}(s_h=\cdot),\end{cases}$$

we obtain

$$\frac{1}{K}\sum_{t=1}^{K}\mathcal{E}_{\mathrm{V}}(f^t,\pi^t,h)\le\mathcal{O}\left(\sqrt{\frac{\dim_{\mathrm{VBE}}(\mathcal{F},\mathcal{D}_{\mathcal{F}},\epsilon/H)|\mathcal{A}|\log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)/\delta]}{K}}+\frac{\epsilon}{H}\right).$$

Plugging in the choice of K completes the proof.

Similarly, for $\mathcal{D}_\Delta$, we can invoke Lemma D.7.1 (b) witht

$$\rho = \frac{\epsilon^2}{H^2 \cdot \dim_{\mathrm{VBE}}(\mathcal{F}, \mathcal{D}_\Delta, \epsilon/H) \cdot |\mathcal{A}|},$$

and Lemma D.5.3 with

$$\begin{cases} \omega = \dfrac{\epsilon}{H},\ C = 1, \\ \mathcal{X} = \mathcal{S},\ \Phi = (I - \mathcal{T}_h)V_{\mathcal{F}},\ \Pi = \mathcal{D}_{\Delta,h}, \\ \phi_k(s) := (f_h^k - \mathcal{T}_h f_{h+1}^k)(s, \pi_{f_h^k}(s)) \text{ and } \mu_k = \mathbf{1}\{\cdot = s_h^k\}, \end{cases}$$

and obtain

$$\begin{aligned} \frac{1}{K}\sum_{t=1}^{K} \mathcal{E}_{\mathrm{V}}(f^t, \pi^t, h) \leq & \frac{1}{K}\sum_{t=1}^{K} (f_h^t - \mathcal{T} f_{h+1}^t)(s_h^t, \pi_{f_h^t}(s_h^t)) + \mathcal{O}\left(\sqrt{\frac{\log K}{K}}\right) \\ \leq & \mathcal{O}\left(\sqrt{\frac{\dim_{\mathrm{VBE}}(\mathcal{F}, \mathcal{D}_\Delta, \epsilon/H)|\mathcal{A}| \log[KH\mathcal{N}_{\mathcal{F}\cup\mathcal{G}}(\rho)/\delta]}{K}} + \frac{\epsilon}{H} + \sqrt{\frac{\log K}{K}}\right), \end{aligned}$$

where the first inequality follows from standard martingale concentration.

Plugging in the choice of K completes the proof.

D.8 Proofs for Examples

D.8.1 Proof of Proposition D.3.6

Proof. Suppose $\mathcal{F}$ has finite ϵ-effective dimension and denote the corresponding mapping by ϕ. Then we can rewrite $\mathcal{F}$ in the form of $\mathcal{F} = \{f_\theta(\cdot) = \langle \phi(\cdot), \theta \rangle_{\mathcal{H}} \mid \theta \in \Theta\}$, where $\Theta \subset B_{\mathcal{H}}(1)$.

Suppose there exists an ϵ'-independent sequence $x_1', \ldots, x_n' \in \mathcal{X}$ with respect to $\mathcal{F}$ where $\epsilon' \geq \epsilon$. By the definition of independent sequence, this is equivalent to the existence of $\theta_1, \ldots, \theta_n \in (\Theta - \Theta)$

and $x_1, \ldots, x_n \in \phi(\mathcal{X})$ such that

$$\begin{cases} \sum_{i=1}^{t-1} (x_i^\top \theta_t)^2 \le \epsilon'^2, & t \in [n] \\ |x_t^\top \theta_t| \ge \epsilon', & t \in [n]. \end{cases} \tag{D.15}$$

Define $\Sigma_t = \sum_{i=1}^{t-1} x_i x_i^\top + \frac{\epsilon'^2}{4} \cdot \mathrm{I}$. We have

$$\|\theta_t\|_{\Sigma_t} \le \sqrt{2}\epsilon' \quad \Longrightarrow \quad \epsilon' \le |x_t^\top \theta_t| \le \|\theta_t\|_{\Sigma_t} \cdot \|x_t\|_{\Sigma_t^{-1}} \le \sqrt{2}\epsilon' \|x_t\|_{\Sigma_t^{-1}}, \quad t \in [n]. \tag{D.16}$$

As a result, we should have $\|x_t\|^2_{\Sigma_t^{-1}} \ge 1/2$ for all $t \in [n]$. Now we can apply the standard log-determinant argument,

$$\sum_{t=1}^{n} \log(1 + \|x_t\|^2_{\Sigma_t^{-1}}) = \log\left(\frac{\det(\Sigma_{n+1})}{\det(\Sigma_1)}\right) = \log\det\left(\mathrm{I} + \frac{4}{\epsilon'^2}\sum_{i=1}^{n} x_i x_i^\top\right),$$

which implies

$$0.5 \le \min_{t \in [n]} \|x_t\|^2_{\Sigma_t^{-1}} \le \exp\left(\frac{1}{n}\log\det\left(\mathrm{I} + \frac{4}{\epsilon'^2}\sum_{i=1}^{n} x_i x_i^\top\right)\right) - 1. \tag{D.17}$$

Choose $n = d_{\text{eff}}(\mathcal{F}, \epsilon/2)$ that is the minimum positive integer satisfying

$$\sup_{x_1, \ldots, x_n \in \phi(\mathcal{X})} \frac{1}{n}\log\det\left(\mathrm{I} + \frac{4}{\epsilon^2}\sum_{i=1}^{n} x_i x_i^\top\right) \le e^{-1}. \tag{D.18}$$

This leads to a contradiction because $\epsilon' \ge \epsilon$ and $0.5 > e^{e^{-1}} - 1$. So we must have

$$\dim_{\mathrm{E}}(\mathcal{F}, \epsilon) \le d_{\text{eff}}(\mathcal{F}, \epsilon/2).$$

□

D.8.2 Proof of Proposition D.3.9

Proof. Consider fixed $\epsilon \in \mathbb{R}^+$ and $h \in [H]$, and denote $n = \dim_{\mathrm{E}}(\mathcal{F}, \epsilon)$. Then by the definition of Eluder dimension, there must exist $x_1, \ldots, x_n \in \mathcal{X}_h$ where $\mathcal{X}_h = \{\phi_h(s,a) : (s,a) \in \mathcal{S} \times \mathcal{A}\}$ so that for any $\theta, \theta' \in B_{\mathcal{H}}(H+1-h)$, if $\sum_{i=1}^n (\langle x_i, \theta - \theta' \rangle_{\mathcal{H}})^2 \le \epsilon^2$, then $|\langle z, \theta - \theta' \rangle_{\mathcal{H}}| \le \epsilon$ for any $z \in \mathcal{X}_h$. In other words, $x_1, \ldots, x_n$ is one of the longest independent subsequences. Therefore, in order to cover $\mathcal{F}_h$, we only need cover the projection of $B_{\mathcal{H}}(H+1-h)$ onto the linear subspace spanned by $x_1, \ldots, x_n$, which is at most n dimensional.

By standard ϵ-net argument, there exists $\mathcal{C} \subset B_{\mathcal{H}}(H+1-h)$ such that: (a) $\log|\mathcal{C}| \le \mathcal{O}(n \cdot \log(1 + nH/\epsilon))$, (b) for any $\theta \in B_{\mathcal{H}}(H+1-h)$, there exists $\hat{\theta} \in \mathcal{C}$ satisfying $\sum_{i=1}^n (\langle x_i, \theta - \hat{\theta} \rangle_{\mathcal{H}})^2 \le \epsilon^2$. By the property of $x_1, \ldots, x_n$, $\{\phi_h(\cdot,\cdot)^\top \hat{\theta} \mid \hat{\theta} \in \mathcal{C}\}$ is an ϵ-cover of $\mathcal{F}_h$. Since $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_H$, we obtain $\log \mathcal{N}_{\mathcal{F}}(\epsilon) \le \mathcal{O}\big(Hn \cdot \log(1 + nH/\epsilon)\big)$. Finally, by Proposition D.3.8, $n \le d(\epsilon)$, which concludes the proof.

□

D.8.3 Proof of Proposition D.3.11

Proof. Assume there exists $h \in [H]$ such that $\dim_{\mathrm{DE}}((I - \mathcal{T}_h)\mathcal{F}, \mathcal{D}_{\mathcal{F},h}, \epsilon) \ge m$. Let $\mu_1, \ldots, \mu_n \in \mathcal{D}_{\mathcal{F},h}$ be a an ϵ-independent sequence with respect to $(I - \mathcal{T}_h)\mathcal{F}$. By Definition 5.3.1, there exist $f^1, \ldots, f^n$ such that for all $t \in [n]$, $\sqrt{\sum_{i=1}^{t-1} (\mathbb{E}_{\mu_i}[f_h^t - \mathcal{T}_h f_{h+1}^t])^2} \le \epsilon$ and $|\mathbb{E}_{\mu_t}[f_h^t - \mathcal{T}_h f_{h+1}^t]| > \epsilon$. Since $\mu_1, \ldots, \mu_n \in \mathcal{D}_{\mathcal{F},h}$, there exist $g^1, \ldots, g^n \in \mathcal{F}$ so that μ_i is generated by executing π_{g^i}, for all $i \in [n]$. By the definition of effective Bellman rank, this is equivalent to: $\sqrt{\sum_{i=1}^{t-1} (\langle \phi_h(f^t), \psi_h(g^i) \rangle)^2} \le \epsilon$ and $|\langle \phi_h(f^t), \psi_h(g^t) \rangle| > \epsilon$ for all $t \in [n]$. For notational simplicity, define $x_i = \psi_h(g^i)$ and $\theta_i = \phi_h(f^i)$. Then

$$\begin{cases} \displaystyle\sum_{i=1}^{t-1} (x_i^\top \theta_t)^2 \le \epsilon^2, \quad t \in [n] \\ |x_t^\top \theta_t| \ge \epsilon, \quad t \in [n]. \end{cases} \tag{D.19}$$

The remaining arguments follow the same as in the proof of Proposition D.3.6 except that we replace ϵ by ϵ/ζ. □

D.8.4 Proof of Proposition D.3.15

Proof. Note that the case $h = 1$ is trivial because each episode always starts from a fixed initial state independent of the policy. For any policy π, function $f \in \mathcal{F}$, and step $h \geq 2$

$$\begin{aligned}\mathcal{E}_{\mathrm{V}}(f, \pi, h) =& \mathbb{E}[f_h(o_h, a_h) - r_h(o_h, a_h) - f_{h+1}(o_{h+1}, a_{h+1}) \mid s_h \sim \pi, a_{h:h+1} \sim \pi_f] \\ =& \mathbb{E}[f_h(o_h, a_h) - r_h(o_h, a_h) - f_{h+1}(o_{h+1}, a_{h+1}) \mid (s_{h-1}, a_{h-1}) \sim \pi, a_{h:h+1} \sim \pi_f] \\ =& \sum_{s,a \in \mathcal{S}} \sum_{s' \in \mathcal{S}} \mathbb{P}^{\pi}(s_{h-1} = s, a_{h-1} = a) \cdot \langle \phi_{h-1}(s, a), \psi_{h-1}(s') \rangle_{\mathcal{H}} \cdot \mathcal{V}(s'),\end{aligned}$$

where

$$\mathcal{V}(s') = \mathbb{E}[f_h(o_h, a_h) - r_h(o_h, a_h) - f_{h+1}(o_{h+1}, a_{h+1}) \mid s_h = s', a_{h:h+1} \sim \pi_f].$$

As a result, we obtain

$$\begin{aligned}& \mathbb{E}[f_h(o_h, a_h) - r_h(o_h, a_h) - f_{h+1}(o_{h+1}, a_{h+1}) \mid s_h \sim \pi, a_{h:h+1} \sim \pi_f] \\ =& \left\langle \mathbb{E}_{\pi}[\phi_{h-1}(s_{h-1}, a_{h-1})], \sum_{s' \in \mathcal{S}} \psi_{h-1}(s') \mathcal{V}(s') \right\rangle_{\mathcal{H}}.\end{aligned}$$

Notice that the left hand side of the inner product only depends on π while the right hand side only depends on f. Moreover, by the definition of kernel reactive POMDPs, the RHS has norm at most 2. Therefore, we conclude the proof by revoking Proposition D.3.13 with $\zeta = 2$. □

D.9 Discussions on $\mathcal{D}_{\mathcal{F}}$ versus $\mathcal{D}_{\Delta}$ in BE Dimension

In this work, we have mainly focused on the BE dimension induced by two special distribution families: (*a*) $\mathcal{D}_{\mathcal{F}}$ — the roll-in distributions produced by executing the greedy policies induced by the functions in $\mathcal{F}$, (*b*) $\mathcal{D}_{\Delta}$ — the collection of all Dirac distributions. And we prove that both low $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon)$ and low $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon)$ can imply sample-efficient learning. As a result, it is natural to ask what is the relation between $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon)$ and $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon)$? Is it possible that one of them is always no larger than the other so that we only need to use the smaller one? We

answer this question with the following proposition, showing that either of them can be arbitrarily larger than the other.

Proposition D.9.1. *There exists absolute constant c such that for any $m \in \mathbb{N}^+$,*

(a) there exist an MDP and a function class $\mathcal{F}$ satisfying for all $\epsilon \in (0, 1/2]$, $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \le c$ while $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon) \ge m$.

(b) there exist an MDP and a function class $\mathcal{F}$ satisfying for all $\epsilon \in (0, 1/2]$, $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon) \le c$ while $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \ge m$.

Proof. We prove (a) first. Consider the following contextual bandits problem ($H = 1$).

- There are m states $s_1, \ldots, s_m$ but the agent always starts at s_1. This means the agent can never visit other states because each episode contains only one step ($H = 1$).
- There are two actions a_1 and a_2. The reward function is zero for any state-action pair.
- The function class $\mathcal{F}_1 = \{f_i(s, a) = \mathbf{1}(s = s_i) + \mathbf{1}(a = a_1) : \ i \in [m]\}$.

First of all, note in this setting $\mathcal{D}_{\Delta}$ is the collection of all Dirac distributions over $\mathcal{S} \times \mathcal{A}$, $\mathcal{D}_{\mathcal{F},1}$ is a singleton containing only $\delta_{(s_1,a_1)}$, and $(I - \mathcal{T}_1)\mathcal{F}$ is simply $\mathcal{F}_1$ because $H = 1$ and $r \equiv 0$. Since $\mathcal{D}_{\mathcal{F},1}$ has cardinality one, it follows directly from definition that $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon)$ is at most 1. Moreover, it is easy to verify that $(s_1, a_2), (s_2, a_2), \ldots, (s_m, a_m)$ is a 1-independent sequence with respect to $\mathcal{F}$ because we have $f_i(s_j, a_2) = \mathbf{1}(i = j)$ for all $i, j \in [m]$. As a result, we have $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon) \ge m$ for all $\epsilon \in (0, 1]$.

Now we come to the proof of (b). Consider the following contextual bandits problem ($H = 1$).

- There are 2 states s_1 and s_2. In each episode, the agent starts at s_1 or s_2 uniformly at random.
- There are m actions $a_1, \ldots, a_m$. The reward function is zero for any state-action pair.
- The function class $\mathcal{F}_1 = \{f_i(s, a) = (2 \cdot \mathbf{1}(s = s_1) - 1) + 0.5 \cdot \mathbf{1}(a = a_i) : \ i \in [m]\}$.

First of all, note in this setting $(I - \mathcal{T}_1)\mathcal{F}$ is simply $\mathcal{F}_1$ and the roll-in distribution induced by the greedy policy of f_i is the uniform distribution over (s_1, a_i) and (s_2, a_i), which we denote as μ_i. It is easy to verify that $\mu_1, \ldots, \mu_m$ is a 0.5-independent sequence with respect to $\mathcal{F}$ because $\mathbb{E}_{(s,a)\sim\mu_i}[f_j(s,a)] = 0.5 \cdot \mathbf{1}(i = j)$. Therefore, for all $\epsilon \in (0, 0.5]$, $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\mathcal{F}}, \epsilon) \geq m$.

Next, we upper bound $\dim_{\mathrm{BE}}(\mathcal{F}, \mathcal{D}_{\Delta}, \epsilon)$ which is equivalent to $\dim_{\mathrm{DE}}(\mathcal{F}_1, \mathcal{D}_{\Delta}, \epsilon)$ in this problem. Assume $\dim_{\mathrm{DE}}(\mathcal{F}_1, \mathcal{D}_{\Delta}, \epsilon) = k$. Then there exist $g_1, \ldots, g_k \in \mathcal{F}_1$ and $w_1, \ldots, w_k \in \mathcal{S} \times \mathcal{A}$ such that for all $i \in [k]$, $\sqrt{\sum_{t=1}^{i-1}(g_i(w_i))^2} \leq \epsilon$ and $|g_i(w_i)| > \epsilon$. Note that we have $|f(s,a)| \in [0.5, 1.5]$ for all $(s, a, f) \in \mathcal{S} \times \mathcal{A} \times \mathcal{F}_1$. Therefore, if $\epsilon > 1.5$, then $k = 0$; if $\epsilon \leq 1.5$, then $k \leq 10$ because $0.5 \times \sqrt{k-1} \leq \sqrt{\sum_{t=1}^{k-1}(g_k(w_t))^2} \leq \epsilon \leq 1.5$. □

Appendix E

Remaining Proofs of Chapter 6

E.1 Proofs for Section 6.3

In this section we provide formal proofs for the results stated in Section 6.3.

Proof of Proposition 6.3.1. We need to verify that $\mathcal{M}^m$ is an MDP. To do so, we check that the state space induces a Markovian dynamics and that the expected reward is also a function of the state. These two properties follow from the m-step decodability assumption.

- *Reward depends on the states.* This holds since the reward is assumed to depend only on the current observation o_h and the current observation is included in the megastate. Formally, for any $s^{m,h} = \left(o_h, o_{h-1}, a_{h-1}, \ldots, o_{\min(h-m,1)}, a_{\min(h-m,1)}\right) \in \mathcal{S}^{m,h}$, any history $\mathcal{H}$, and policy π, it holds that

$$\mathbb{E}_\pi \left[r | s^{m,h}, \mathcal{H}\right] = \mathbb{E}_\pi \left[r | s^{m,h}\right] = r(o_h)$$

  where $o_h \in s^{m,h}$ due to the assumption on the reward generation process of m-step decodable POMDP.

- *Transition model is Markov.* For any $s^{m,h} = \left(o_h^h, o_{h-1}^h, a_{h-1}^h, \ldots, o_{\min(h-m,1)}^h, a_{\min(h-m,1)}^h\right) \in \mathcal{S}^{m,h}$ and $s_{h+1}^n = \left(o_{h+1}^{h+1}, o_h^{h+1}, a_h^{h+1}, \ldots, o_{\min(h+1-m,1)}^{h+1}, a_{\min(h+1-m,1)}^{h+1+1}\right) \in \mathcal{S}_{h+1}^m$, any action $a_h^h \in \mathcal{A}$, any history $\mathcal{H}$ and any policy π it holds that

$$\mathbb{P}_\pi\left(s^{m,h+1} \mid s^{m,h}, a_h, \mathcal{H}\right) = \mathbb{P}_\pi\left(o_{h+1} \mid s^{m,h}, a_h, \mathcal{H}\right) \cdot \prod_{j=\min(h+1-m,1)}^{h} \delta\left(o_j^{h+1} = o_j^h, a_j^{h+1} = a_j^h\right)$$

Finally, observe that by the m-step decodability assumption it holds that

$$\mathbb{P}_\pi\left(o_{h+1} \mid \phi^\star(s^{m,h}) = s, a_h, \mathcal{H}\right) = \mathbb{O}_{h+1}\left(o_{h+1} \mid s_{h+1}\right) \mathbb{P}\left(s_{h+1} \mid \phi^\star(s^{m,h}) = s, a_h\right),$$

where the last relation holds by the Markov assumption of the latent model. This shows that

$$\mathbb{P}_\pi\left(s^{m,h+1} \mid s^{m,h}, a_h, \mathcal{H}\right) = \mathbb{P}\left(s^{m,h+1} \mid s^{m,h}, a_h\right),$$

and hence the dynamics are Markovian.

Lastly, we elaborate on the optimality of any optimal policy of $\mathcal{M}^m$; that is, any optimal policy of $\mathcal{M}^m$ is an optimal policy of the m-step decodable POMDP. First, observe that the optimal policy of the latent MDP that underlies the m-step decodable POMDP is also the optimal policy of the m-step decodable POMDP.

Further, since the latent state is decodable from a suffix of length m of the history, any state in $\mathcal{S}^{m,h}$ (that represents a reachable suffix) can decode the latent state. Hence, the optimal policy on the latent MDP can be executed based on the states in $\mathcal{S}^{m,h}$. Thus, an optimal policy of $\mathcal{M}^m$ is also an optimal policy of the m-step decodable POMDP; otherwise, an optimal policy of the latent MDP is not optimal for the m-step decodable POMDP.

□

Proof of Corollary 6.3.2. The sample complexity follows immediately from a standard online-to-batch conversion of the minimax optimal regret bound in Azar et al. (2017), combined with

Proposition 6.3.1. In particular, the online-to-batch conversion gives $\tilde{O}(HSA\log^2(1/\delta)/\epsilon^2)$ sample complexity in an MDP with S states and A actions. By Proposition 6.3.1 we have an MDP with $O^m A^{m-1}$ states, so the result follows. □

Proof sketch of Proposition 6.3.3. We construct a simple m-step decodable POMDP with horizon m, two states per layer and two actions. The construction and argument are identical to the one in Krishnamurthy et al. (2016), so we only sketch the construction here. It is a standard "combination lock" construction, with A actions and no observations, but where the state is decodable from the past actions.

In particular, the agent starts in the "good state" g_1 and at each time step h can be either in the good state g_h or the "bad state" b_h. From the good state, a special action $a_h^\star$ transits to the next good state, while all other actions (from both good or bad state) transit to the next bad state b_{h+1}. At the last time step the agent gets reward for being in state g_m. There are no observations (or there is a trivial observation), but note that the latent state is decodable using the history of actions. Thus provided the horizon $H \leq m$ the process is m-step decodable.

Intuitively, the construction requires the agent to try all A^m action sequences before finding the reward. More formally this construction embeds an $\Omega(A^m)$ armed bandit problem resulting in a sample complexity lower bound of $\Omega(A^m/\epsilon^2)$. We refer the reader to Krishnamurthy et al. (2016) for more details.

□

E.2 Proof for Section 6.4 and 6.5

In this section we provide formal proofs for the results stated in Section 6.4 and 6.5.

E.2.1 Properties of Moment Matching Policy

We start with formal definition of *moment matching policy*. For a policy π, we construct $\nu^{\pi,h}_{h'}$ for $h' \geq h - m$ such that it matches the distribution of the action $a_{h'}$ conditioning on latent states and observations from time step $h - m + 1$ to time step h under the sampling process of π. For this reason we refer to ν^{π} as the *moment matching policy* for π (see Fig. 6.2 for illustration). Formally, we define it as follows:

Definition E.2.1 (Moment-Matching Policy for π)**.** Denote $m(h) = h - m + 1$; Fix $h \in [H]$ and for $h' \in [m(h), h]$ we define

$$x_{h'} = \left(s_{m(h):h'}, o_{m(h):h'}, a_{m(h):h'-1}\right) \in \mathcal{X}_l,$$

where $\mathcal{X}_l = \mathcal{S}^l \times \mathcal{O}^l \times \mathcal{A}^{l-1}$ and $l = h' - m(h)$. For a m-step policy π and $h \in [H]$, we define the moment matching policy $\mu^{\pi,h} = \{\mu^{\pi,h}_{h'} : \mathcal{X}_l \to \Delta(\mathcal{A})\}^{h}_{h'=m(h)}$ as following:

$$\mu^{\pi,h}_{h'}(a_{h'} \mid x_{h'}) \coloneqq \mathbb{E}_{\pi}[\pi_{h'}(a_{h'} \mid z_{h'}) \mid x_{h'}].$$

By Assumption 6.2.2, states and therefore x'_h is decodable by the history of actions and observations, therefore we let

$$\nu^{\pi,h}_{h'}(a_{h'} \mid o_{1:h'}, a_{1:h'-1}) \coloneqq \mu^{\pi,h}_{h'}(a_{h'} \mid x'_h).$$

As we discussed in Section 6.5, we prove the following lemma that establishes two important properties of the moment matching policy.

Lemma E.2.2. *For a fixed $h \in [H]$ and fixed m-step policies $\pi, \bar{\pi}$, define policy $\tilde{\pi}^h$ which takes first $m(h) - 1$ actions from π and remaining actions from $\nu^{\bar{\pi},h}$, i.e. $\tilde{\pi}^h = \pi \circ_{m(h)} \nu^{\pi,h}$. Then we have,*

1. *If $\pi = \bar{\pi}$, for any $z_h \in \mathcal{Z}_h$, $P_{\pi}(z_h) = P_{\tilde{\pi}^h}(z_h)$*

2. *For any function* $g : \mathcal{Z}_h \to [0,1]$,

$$\mathbb{E}_{\tilde{\pi}^h}[g(z_h)] = \langle \zeta_h(\pi), \xi_h(g, \bar{\pi}) \rangle,$$

where $\zeta_h(\pi), \xi_h(g,\bar{\pi}) \in \mathbb{R}^S$ *satisfying* $\|\zeta_h(\pi)\| \le 1$ *and* $\|\xi_h(g,\bar{\pi})\| \le \sqrt{S}$.

Recall that we use the notation $m(h) = \min\{h-m+1, 1\}$ and that we define $x_{h'} = (s_{m(h):h'}, o_{m(h):h'}, a_{m(h):h'-1})$ for $h' \in [m(h), h]$. By definition of $\mu^{\pi,h}$ (as in Definition E.2.1), for $h' \in [m(h), h]$ we have

$$\mu^{\pi,h}_{h'}(a_{h'} \mid x_{h'}) P_\pi\left[x_{h'}\right] = \sum_{(o,a)_{m(h'):m(h)-1}} \pi(a_{h'} \mid z_{h'}) P_\pi\left[(o,a)_{m(h'):m(h)-1}, x_{h'}\right] \tag{E.1}$$

We will this identity below.

Proof of Lemma E.2.2. Recall that we define $\tilde{\pi}^h$ to take actions $a_{1:m(h)-1}$ according to π and take actions $a_{m(h):h-1}$ according to the moment matching policy $\nu^{\pi,h}$.

Item 1. We prove the first item by induction on $h' \in \{m(h), \dots, h\}$, where the induction hypothesis is

$$\forall x_{h'} : \quad P_\pi\left[x_{h'}\right] = P_{\tilde{\pi}^h}\left[x_{h'}\right]$$

- **Base case:** The base case is when $h' = m(h)$. In this case, $P_\pi\left[(s,o)_{m(h)}\right] = P_{\tilde{\pi}^h}\left[(s,o)_{m(h)}\right]$ since all actions up to $a_{m(h)-1}$ are taken by the same policy.
- **Induction step**: Let $h' \in \{m(h), \dots, h\}$ and assume $P_\pi\left[x_{h'-1}\right] = P_{\tilde{\pi}^h}\left[x_{h'-1}\right]$. We have

$$\begin{aligned}
P_\pi(x_{h'+1}) &= P_\pi\left[(s,o,a)_{m(h):h'}, (s,o)_{h'+1}\right] \\
&= \sum_{(o,a)_{m(h'):m(h)-1}} P_\pi\left[(o,a)_{m(h'):m(h)-1}, x_{h'}, a_{h'}, (s,o)_{h'+1}\right] \\
&= \sum_{(o,a)_{m(h'):m(h)-1}} \mathbb{O}(o_{h'+1} \mid s_{h'+1}) \mathbb{P}(s_{h'+1} \mid s_{h'}, a_{h'}) \pi(a_{h'} \mid z_{h'}) P_\pi\left[(o,a)_{m(h'):m(h)-1}, x_{h'}\right]
\end{aligned}$$

Similarly we have,

$$\begin{aligned}
P_{\tilde{\pi}^h}(x_{h'+1}) &= P_{\tilde{\pi}^h}\left[(s,o,a)_{m(h):h'}, (s,o)_{h'+1}\right] \\
&= P_{\tilde{\pi}^h}\left[x_{h'}, a_{h'}, (s,o)_{h'+1}\right] \\
&= \mathbb{O}(o_{h'+1} \mid s_{h'+1})\mathbb{P}(s_{h'+1} \mid s_{h'}, a_{h'})\mu_{h'}^{\pi,h}(a_{h'} \mid x_{h'})P_{\tilde{\pi}^h}\left[x_{h'}\right] \\
&\overset{(i)}{=} \mathbb{O}(o_{h'+1} \mid s_{h'+1})\mathbb{P}(s_{h'+1} \mid s_{h'}, a_{h'})\mu_{h'}^{\pi,h}(a_{h'} \mid x_{h'})P_{\pi}\left[x_{h'}\right],
\end{aligned}$$

where (i) uses the induction hypothesis. Eq. (E.1) implies that right-hand side of the two above expressions are equal, which completes the proof of induction step.

Now item 1 is immediate since the variables in z_h are contained within x_h, in particular

$$P_\pi(z_h) = \sum_{s_{m(h):h}} P_\pi(x_h) = \sum_{s_{m(h):h}} P_{\tilde{\pi}^h}(x_h) = P_{\tilde{\pi}^h}(z_h).$$

Item 2. Recall that here $\tilde{\pi}^h$ is defined to take actions $a_{1:m(h)-1} \sim \pi$ and $a_{m(h):h-1} \sim \nu^{\bar{\pi},h}$ where π and $\bar{\pi}$ may not be equal. Since $\mu^{\bar{\pi},h}$ is defined to be independent of the past give $s_{m(h)}$ we have the factorization

$$\mathbb{E}_{\tilde{\pi}^h}[g(z_h)] = \sum_{s_{m(h)} \in \mathcal{S}} P_\pi(s_{m(h)}) \cdot \mathbb{E}_{a_{m(h):h-1} \sim \mu^{\bar{\pi},h}}[g(z_h) \mid s_{m(h)}].$$

We note that $\mu^{\bar{\pi},h}$ only depends on $(s,o)_{m(h):h-1}$ and $a_{m(h):h-2}$, thus the second term is independent of π and only depends g and $\bar{\pi}$. Defining

$$\zeta_h(\pi) \coloneqq \left(P_\pi(s_{m(h)})\right)_{s_{m(h)} \in \mathcal{S}} \in \mathbb{R}^S \quad \text{and} \quad \xi_h(g, \bar{\pi}) = \left(\mathbb{E}_{a_{m(h):h-1} \sim \mu^{\bar{\pi},h}}[g(z_h) \mid s_{m(h)}]\right)_{s_{m(h)} \in \mathcal{S}} \in \mathbb{R}^S,$$

completes the proof. □

E.2.2 Concentration lemmas

We start with the following lemma, which is quite similar to Lemmas 39 and 40 in Jin et al. 2021. The lemma shows that: (1) with high probability any function in the confidence set at the k^{th} iteration has low Bellman error over the data distributions from visited in the previous iterations at all layers $h \in [H]$ and (2) the optimal value function is inside the confidence set with high probability.

Lemma E.2.3. *For any $\rho > 0$ and $\delta \in (0,1)$, if we run Algorithm 6 with $\beta = c\Big(\log\big[KH\mathcal{N}_{\mathcal{G}}(\rho)/\delta\big] + K\rho\Big)$ where $c > 0$ is an absolute constant, then with probability at least $1-\delta$, we have*

1. $\sum_{i=1}^{k-1} \mathbb{E}\Big[\big(f_h^k(z_h,a_h) - (\mathcal{T}_h f_h^k)(z_h,a_h)\big)^2 \mid a_{1:h-m} \sim \pi^i, a_{h-m+1:h} \sim \text{unif}(\mathcal{A})\Big] \le \mathcal{O}(\beta)$ *for all* $(k,h) \in [K] \times [H]$,
2. $Q^\star \in \mathcal{B}^k$ *for all* $k \in [K]$.

Proof of Lemma E.2.3. The proof relies on a standard martingale concentration inequality (e.g., Freedman's inequality), the construction of our confidence set, and our generalized completeness assumption (Assumption 6.2.4). The argument is almost identical to the proofs of Lemma 39 and 40 in Jin et al. 2021 and therefore omitted for brevity. □

Lemma E.2.4. *For any $\delta \in (0,1)$, if we choose $K_{\text{est}} = c \cdot \big(\log[\mathcal{N}_{\mathcal{F}}(\rho_{\text{est}})/\delta]/\rho_{\text{est}}^2\big)$ where $c > 0$ is some absolute constant; then, with probability at least $1-\delta$ for any $f \in \mathcal{F}$, we have*

$$|\hat{f}_1 - \mathbb{E}_{s_1}\big[f_1(o_1, \pi_f(o_1))\big]| \le \mathcal{O}(\rho_{\text{est}}).$$

Proof. The proof follows from applying uniform concentration argument over a ρ_{est}-cover of $\mathcal{F}$; then, a covering argument finishes the proof. □

E.2.3 Eluder Dimension

In this section we describe complexity measure *Eluder dimension* proposed by Russo and Van Roy (2013) since it has been used in the analysis of the original GOLF algorithm Jin et al. (2021).

Definition E.2.5 (ϵ-Independence)**.** Let $\mathcal{W}$ be a function class defined over domain $\mathcal{Y}$ and $y^1, \ldots, y^n, \bar{y}$ be elements in $\mathcal{Y}$. We say $\bar{y}$ is ϵ-independent with respect to $\mathcal{W}$, if there exists $w \in \mathcal{W}$ such that $\sqrt{\sum_{i=1}^n [w(y^i)]^2} \leq \epsilon$, but $|w(\bar{y})| > \epsilon$.

Definition E.2.6 (Eluder Dimension)**.** The Eluder dimension $\dim_{\mathrm{E}}(\mathcal{W}, \epsilon)$, is the length of the longest sequence of $\{y^1, \ldots, y^n\}$ in $\mathcal{Y}$, such that there exists $\epsilon' \geq \epsilon$ where y^i is ϵ'-independent of $\{y^i, \ldots, y^{i-1}\}$ with respect to $\mathcal{W}$ for all $i \in [n]$.

The following proposition shows that if $\mathcal{W}$ has a low rank structure with rank d, then the Eluder dimension can be upper bounded by $\tilde{\mathcal{O}}(d)$.

Proposition E.2.7 (Proposition 6 in Russo and Van Roy 2013)**.** *Suppose for any $w \in \mathcal{W}$ and any $y \in \mathcal{Y}$, we have $w(y) = \langle \zeta(y), \xi(w) \rangle$, where $\zeta(y), \xi(w) \in \mathbb{R}^d$ satisfying $\|\zeta(y)\| \cdot \|\xi(w)\| \leq \gamma$. Then we have,*

$$\dim_{\mathrm{E}}(\mathcal{W}, \epsilon) \leq \mathcal{O}\big(1 + d \log[1 + \gamma/\epsilon^2]\big).$$

The following lemma could be seen as an analogue to the standard elliptical potential argument for Eluder dimension that was proposed by Russo and Van Roy (2013) and been used in analysis of GOLF. The following lemma could be obtained from Lemma 41 in Jin et al. (2021) by setting the family of probability measures used in that lemma to be $\{\delta_y \mid y \in \mathcal{Y}\}$, where δ_y is the dirac measure centered at y.

Lemma E.2.8 (Simplification of Lemma 41 in Jin et al. 2021)**.** *Given a function class $\mathcal{W}$ defined over $\mathcal{Y}$ with $w(y) \leq C$ for all $(w, y) \in \mathcal{W} \times \mathcal{Y}$; Suppose $\{y^i\}_{i=1}^K \subseteq \mathcal{Y}$ and $\{w^i\}_{i=1}^K \subseteq \mathcal{W}$ satisfy that for all $k \in K$, $\sum_{i=1}^{k-1} [w^k(y^i)]^2 \leq \alpha$. Then for all $k \in [K]$ and $\omega > 0$, we have*

$$\sum_{i=1}^{k} |w^i(y^i)| \leq \mathcal{O}\Big(\sqrt{\dim_{\mathrm{E}}(\mathcal{W}, \omega)\alpha k} + \min\{k, \dim_{\mathrm{E}}(\mathcal{W}, \omega)\} \cdot C + k\omega\Big).$$

E.2.4 Proof of Theorem 6.4.1

We use $\mathcal{E}_h(\pi, f)$ to denote the Bellman error of function $f \in \mathcal{F}$ at step h using roll-in policy π, which is defined as

$$\mathcal{E}_h(\pi, f) = \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h)) \mid a_{1:h-1} \sim \pi].$$

In addition, we use $\mathcal{E}^\star_h(\pi, f)$ to denote the Bellman error of function f at step h using roll-in policy π for the first $h - m$ steps and $\nu^{\pi_f, h}$ (the moment matching policy for π_f) for $a_{m(h):h-1}$; namely,

$$\mathcal{E}^\star_h(\pi, f) = \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h)) \mid a_{1:h-m} \sim \pi, a_{m(h):h-1} \sim \nu^{\pi_f, h}].$$

The next lemma shows that $\mathcal{E}^\star_h$ satisfies two important properties that are critical to the rest of the proof. The first property is that when $\pi = \pi_f$, $\mathcal{E}_h$ and $\mathcal{E}^\star$ coincide. The second property shows that $\mathcal{E}^\star_h$ has low rank or bilinear structure.

Lemma E.2.9. *For any policy π, any function $f \in \mathcal{F}$, and any $h \in [H]$, we have*

1. $\mathcal{E}_h(\pi_f, f) = \mathcal{E}^\star_h(\pi_f, f)$
2. $\mathcal{E}^\star_h(\pi, f) = \langle \zeta_h(\pi), \xi_h(f) \rangle$ *where* $\zeta_h(\pi), \xi_h(f) \in \mathbb{R}^S$ *satisfy* $\|\zeta_h(\pi)\| \le 1$ *and* $\|\xi_h(f)\| \le 2\sqrt{S}$.

Proof of Lemma E.2.9. For item (1) define $\tilde{\pi}^h_f$ to be the policy that takes actions $a_{1:h-m} \sim \pi_f$ and $a_{m(h):h-1} \sim \nu^{\pi_f, h}$, and let $g : \mathcal{Z}_h \to [0, 2]$ be defined as $g(z_h) = (f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h))$. Then by item (1) of Lemma E.2.2 we have

$$\begin{aligned}
\mathcal{E}^\star_h(\pi, f) &= \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h))) \mid a_{1:h-m} \sim \pi, a_{m(h):h} \sim \nu^{\pi_f, h}] \\
&= \sum_{z_h \in \mathcal{Z}_h} P_{\tilde{\pi}^h_f}(z_h) \cdot g(z_h) = \sum_{z_h \in \mathcal{Z}_h} P_{\pi_f}(z_h) \cdot g(z_h) \\
&= \mathbb{E}[(f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h))) \mid a_{1:h-1} \sim \pi] = \mathcal{E}_h(\pi_f, f),
\end{aligned}$$

Item (2) immediately follows from item (2) of Lemma E.2.2 by selecting g as $g(z_h) = (f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h))$ and $\bar{\pi} = \pi_g$. □

The following corollary shows that Eluder dimension with respect to $\mathcal{E}^\star$ is upper bounded by $\tilde{\mathcal{O}}(S)$. The proof immediately follows from Lemma E.2.9 and Proposition E.2.7.

Corollary E.2.10. *Let Π to be set of all m-step policies, and define $\mathcal{W}^\star_{\mathcal{F}} = \{\mathcal{E}^\star(\cdot, f) : \Pi \to [0,2] \mid f \in \mathcal{F}\}$, then*

$$\dim_{\mathrm{E}}(\mathcal{W}^\star_{\mathcal{F}}, e) \leq \mathcal{O}\big(S \log[S/\epsilon]\big).$$

Now we are ready to prove Theorem 6.4.3.

Proof of Theorem 6.4.1. With probability at least $1-2\delta$ the events in Lemma E.2.3 and Lemma E.2.4 holds. Under this good event, we proceed in several steps.

Step 1. Bounding the optimality gap by the Bellman error. Lemma E.2.3 guarantees that $\forall k \in [K]: \; Q^\star \in \mathcal{B}^k$, this together with optimistic choice of f^k (Line 4 in Algorithm 6), for all $k \in [K]$, we have:

$$V^\star \leq \hat{Q}^\star_1 + \mathcal{O}(\rho_{\mathrm{est}}) \leq \hat{f}^k_1 + \mathcal{O}(\rho_{\mathrm{est}}) \leq \mathbb{E}_{s_1}\big[f^k_1(o_1, \pi_{f^k}(o_1))\big] + 2 \cdot \mathcal{O}(\rho_{\mathrm{est}}).$$

It implies that $\sum_{k=1}^K \big(V^\star - V^{\pi^k}\big) \leq \sum_{k=1}^K \mathbb{E}_{s_1}\big[f^k_1(o_1, \pi_{f^k}(o_1))\big] - V^{\pi^k} + \mathcal{O}(K\rho_{\mathrm{est}})$. We also have

$$\mathbb{E}_{s_1}\big[f^k_1(o_1, \pi_{f^k}(o_1))\big] - V^{\pi^k} \overset{(i)}{=} \sum_{k=1}^K \sum_{h=1}^H \mathcal{E}_h(\pi^k, f^k) \overset{(ii)}{=} \sum_{h=1}^H \sum_{k=1}^K \mathcal{E}^\star_h(\pi^k, f^k),$$

where (i) is by standard policy loss decomposition (e.g., Lemma 1 in Jiang et al. 2017) and (ii) is due to part (1) of Lemma E.2.9 since we have $\pi^k = \pi_{f^k}$. Therefore, we showed

$$\sum_{k=1}^K \big(V^\star - V^{\pi^k}\big) \leq \sum_{h=1}^H \sum_{k=1}^K \mathcal{E}^\star_h(\pi^k, f^k) + \mathcal{O}(K\rho_{\mathrm{est}})$$

Step 2: Utilizing the confidence set. By Lemma E.2.3, we have

$$\sum_{i=1}^{k-1}\mathbb{E}\Big[\big((f_h^k-\mathcal{T}_h f_{h+1}^k)(z_h,a_h)\big)^2 \mid a_{1:h-m}\sim\pi^i, a_{h-m+1:h}\sim\mathrm{unif}(\mathcal{A})\Big]\le\mathcal{O}(\beta)\quad\forall(k,h)\in[K]\times[H].$$

It implies that

$$\begin{aligned}\sum_{i=1}^{k-1}[\mathcal{E}_h^\star(\pi^i,f^k)]^2 &\le \sum_{i=1}^{k-1}\mathbb{E}\Big[\big((f_h^k-\mathcal{T}_h f_{h+1}^k)(z_h,\pi_f(z_h))\big)^2 \mid a_{1:h-m}\sim\pi^i, a_{h-m+1:h}\sim\nu^{\pi_{f^k},h}\Big]\\ &\le A^m\sum_{i=1}^{k-1}\mathbb{E}\Big[\big((f_h^k-\mathcal{T}_h f_{h+1}^k)(z_h,a_h)\big)^2 \mid a_{1:h-m}\sim\pi^i, a_{h-m+1:h}\sim\mathrm{unif}(\mathcal{A})\Big]\\ &\le \mathcal{O}(A^m\beta).\end{aligned}$$

Here the A^m factor arises to change measure from $\nu^{\pi_{f^k},h}$ to the uniform distribution over actions $a_{h-m+1:h}$.

Step 3: Utilizing Low-rank Structure. From previous step, we know that $\sum_{i=1}^{k-1}[\mathcal{E}_h^\star(\pi^i,f^k)]^2\le A^m\beta$, Therefore if we invoke Lemma E.2.8 and Corollary E.2.10 with

$$\begin{cases}\mathcal{Y}=\Pi, \qquad \mathcal{W}=\mathcal{W}_{\mathcal{F}}^\star=\{\mathcal{E}^\star(\cdot,f):\Pi\to[0,2]\mid f\in\mathcal{F}\},\\ \omega=\epsilon/H, \quad \alpha=\mathcal{O}(A^m\beta), \quad C=2,\end{cases}$$

we obtain

$$\frac{1}{K}\sum_{k=1}^{K}\mathcal{E}_h^\star(\pi^k,f^k)\le\mathcal{O}\Big(\sqrt{\frac{A^m S\log[S/\epsilon]\beta}{K}}+\epsilon/H\Big)$$

Step 4: Putting everything together Choosing $\rho_{\mathrm{est}}=\mathcal{O}(\epsilon)$ and combining the conclusion of step 1 and step 3, we have

$$\frac{1}{K}\sum_{k=1}^{K}\big(V^\star-V^{\pi^k}\big)\le\frac{1}{K}\sum_{h=1}^{H}\sum_{k=1}^{K}\mathcal{E}_h^\star(\pi^k,f^k)\le\mathcal{O}\Big(\sqrt{\frac{H^2A^mS\log[S/\epsilon]\beta}{K}}+\epsilon\Big)+\mathcal{O}(\epsilon).$$

By definition of π^{out}, we have

$$V^\star - V^{\pi^{\text{out}}} = \frac{1}{K}\sum_{k=1}^{K}\big(V^\star - V^{\pi^k}\big) \le \mathcal{O}\Big(\sqrt{\frac{H^2A^mS\log[S/\epsilon]\beta}{K}}\Big) + \mathcal{O}(\epsilon)$$
$$\overset{(i)}{\le} \mathcal{O}\Big(\sqrt{\frac{H^2A^mS\log[S/\epsilon]\log[KH\mathcal{N}_{\mathcal{G}}(\rho)/\delta]}{K} + H^2A^mS\log[S/\epsilon]\rho}\Big) + \mathcal{O}(\epsilon)$$
$$\overset{(ii)}{\le} \mathcal{O}\Big(\sqrt{\frac{H^2A^mS\log[S/\epsilon]\log[KH\mathcal{N}_{\mathcal{G}}(\rho)/\delta]}{K}}\Big) + \mathcal{O}(\epsilon),$$

where (i) is follows from $\beta = c\Big(\log\big[KH\mathcal{N}_{\mathcal{G}}(\rho)/\delta\big] + K\rho\Big)$ as in Lemma E.2.3 and (ii) is by picking

$$\rho = \frac{\epsilon^2}{(H^2A^mS\log[S/\epsilon])}.$$

We need to pick K such that

$$\sqrt{\frac{H^2A^mS\log[S/\epsilon]\log[KH\mathcal{N}_{\mathcal{G}}(\rho)/\delta]}{K}} \le \mathcal{O}(\epsilon).$$

By simple calculations, one can verify that it suffices to pick

$$K \ge \Omega\Big(\frac{H^2SA^m}{\epsilon^2}\cdot\log[HSA^m\mathcal{N}_{\mathcal{G}}(\rho)/(\delta\epsilon)]\cdot\log[S/\epsilon]\Big),$$

which completes the proof. □

E.2.5 Proof for Theorem 6.4.3

The following lemma (akin to part (2) of Lemma E.2.9) shows that $\mathcal{E}^*$ has low rank structure with rank d_{lin}. The proof of Theorem 6.4.3 is almost identical to proof of Theorem 6.4.1 where the only difference is to use Lemma E.2.11 instead of part (2) of Lemma E.2.9 resulting in S being replaced by d_{lin} wherever it has been used.

Lemma E.2.11 (akin to part (2) of Lemma E.2.9)**.** *Under Definition 6.4.2; for any policy π and any function $f \in \mathcal{F}$, and any $h \in [H]$, we have $\mathcal{E}_h^\star(\pi, f) = \langle\zeta_h(\pi), \xi_h(f)\rangle$ where $\zeta_h(\pi), \xi_h(f) \in \mathbb{R}^{d_{\text{lin}}}$*

Algorithm 21 IS-RL: Importance sampling for Reinforcement Learning

1: **Initialize:** N number of samples, policy class Π,
2: **Collect:** N trajectories $\{o_h^{(t)}, a_h^{(t)}, r_h^{(t)}\}_{h=1}^H$ for $t \in [N]$ by executing the uniform policy $a_h^{(t)} \sim \text{Uniform}(\mathcal{A})$.
3: For any $\pi \in \Pi$ calculate its empirical value

$$\widehat{V}^\pi = \frac{1}{N} \sum_{t=1}^{N} \prod_{h=1}^{H} \left(\frac{\pi(a_h^{(t)} \mid z_h^{(t)})}{1/A} \right) \cdot \left(\sum_{h=1}^{H} r_h^{(t)} \right)$$

4: **Output** $\widehat{\pi} \in \arg\max_{\pi \in \Pi} \widehat{V}^\pi$.

satisfy $\|\zeta_h(\pi)\| \le 1$ *and* $\|\xi_h(f)\| \le 2\sqrt{d_{\text{lin}}}$.

Proof of Lemma E.2.11. Let g be a function $g : \mathcal{Z}_h \to [0,1]$ and $\tilde{\pi}^h = \pi \circ_{m(h)} \bar{\pi}$. Recall that here $\tilde{\pi}^h$ is defined to take actions $a_{1:m(h)-1} \sim \pi$ and $a_{m(h):h-1} \sim \nu^{\bar{\pi},h}$ where π and $\bar{\pi}$ may not be equal. Since $\mu^{\bar{\pi},h}$ is defined to be independent of the past given $s_{m(h)}$ we have the factorization

$$\begin{aligned}
\mathbb{E}_{\tilde{\pi}^h}[g(z_h)] &= \mathbb{E}_\pi \Big[\int_{s_{m(h)} \in \mathcal{S}} \langle \psi_\pi(s_{m(h)-1}, a_{m(h)-1}), \boldsymbol{\mu}(s_{m(h)}) \cdot \mathbb{E}_{a_{m(h):h-1} \sim \mu^{\bar{\pi},h}}[g(z_h) \mid s_{m(h)}] \Big] \\
&= \big\langle \mathbb{E}_\pi \psi_\pi(s_{m(h)-1}, a_{m(h)-1}), \int_{s_{m(h)} \in \mathcal{S}} \boldsymbol{\mu}(s_{m(h)}) \cdot \mathbb{E}_{a_{m(h):h-1} \sim \mu^{\bar{\pi},h}}[g(z_h) \mid s_{m(h)}] \big\rangle
\end{aligned}$$

We note that $\mu^{\bar{\pi},h}$ only depends on $(s,o)_{m(h):h-1}$ and $a_{m(h):h-2}$, thus the second term is independent of π and only depends g and $\bar{\pi}$. Define

$$\begin{aligned}
\zeta_h(\pi) &:= \mathbb{E}_\pi \psi_\pi(s_{m(h)-1}, a_{m(h)-1}) \in \mathbb{R}^d_{\text{lin}} \\
\xi_h(g, \bar{\pi}) &= \int_{s_{m(h)} \in \mathcal{S}} \boldsymbol{\mu}(s_{m(h)}) \cdot \mathbb{E}_{a_{m(h):h-1} \sim \mu^{\bar{\pi},h}}[g(z_h) \mid s_{m(h)}] \in \mathbb{R}^d_{\text{lin}}.
\end{aligned}$$

Picking g as $g(z_h) = (f_h - \mathcal{T}_h f_{h+1})(z_h, \pi_f(z_h))$ and $\bar{\pi} = \pi_g$ completes the proof. □

E.3 On H-Step Decodable POMDPs

In this section, we show that there exists an algorithm that returns an ϵ optimal policy for any H-step decodable POMDP with sample complexity which is only polynomial in $|\mathcal{O}|$, the cardinality

of the observation space. To do so, we construct a policy class Π that contains the optimal policy and has cardinality bounded by $|\Pi| \leq O\left(H(SA)^{2HSOA}\right)$ and we use this policy class in a standard importance-sampling procedure. The procedure is formally specified Algorithm 21, and Proposition 6.6.1 follows immediately from Corollary E.3.2 and Lemma E.3.3.

Constructing the policy class Π via recurrent function class. Let $\mathcal{B}_h$ denote the set of all mappings of the form $b_h : \mathcal{S}_{h-1} \times \mathcal{A}_{h-1} \times \mathcal{O}_h \to \mathcal{S}_h$. This class represents all mappings from the latent state at the previous time step, action at the previous time step, and current observation to the latent state at the current time step. We call them *belief operators*.

We show that the latent state at time step h is decodable from the tuple (o_h, s_{h-1}, a_{h-1}). In other words, we can write $\phi^\star(z_h) = b_h^\star(\phi^\star(z_{h-1}), a_{h-1}, o_h)$ for some belief operator $b_h^\star \in \mathcal{B}_h$. This relation is established in the following lemma.

Lemma E.3.1. *For each $h \in [H]$ there exists $b_h^\star \in \mathcal{B}_h$ such that for all reachable histories z_h we have $\phi^\star(z_h) = b_h^\star(\phi^\star(z_{h-1}), a_{h-1}, o_h)$.*

Using the belief operator class we can design a policy class that contains the optimal policy for any H-step decodable POMDP. Given a decoder $\vec{b} := (b_1, \ldots, b_H) \in \mathcal{B}_1 \times \ldots \times \mathcal{B}_H$ and a trajectory z_H (or a partial trajectory z_h), the predicted state is updated recursively as $\hat{s}_1 = b_1(o_1)$, $\hat{s}_h = b_h(\hat{s}_{h-1}, a_{h-1}, o_h)$. Then we can define $\Pi_{\vec{b}} := \{\pi : \pi(a_h \mid z_h) = \pi_h(a_h \mid \hat{s}_h)\}$, where here implicitly we are updated $\hat{s}_h$ using $\vec{b}$. Then we can take $\Pi = \bigcup_{\vec{b} \in \vec{\mathcal{B}}} \Pi_b$. For this class we have the following corollary.

Corollary E.3.2. *We have $|\Pi| \leq (SA)^{2SHOA}$ and for any H-step decodable POMDP $\pi^\star \in \Pi$.*

Importance Sampling Procedure for H-step POMDPs. Algorithm 21 describes a standard importance sampling approach for policy learning in POMDPs, which is essentially the same as the trajectory tree method of Kearns et al. (1999). A standard analysis of importance weighting using

Bernstein's inequality and a uniform convergence argument yield the following lemma. As the result is quite standard, we omit the proof here.

Lemma E.3.3. *Fix any $\epsilon, \delta > 0$ and let $N = \Omega\left(HA^H \log\left(|\Pi|/\delta\right)/\epsilon^2\right)$. Then with probability at least $1-\delta$, Algorithm 21 returns a policy $\widehat{\pi} \in \Pi$ such that*

$$\max_{\pi \in \Pi} V^{\pi} \leq V^{\widehat{\pi}} + \epsilon.$$

E.3.1 Proofs

We now turn to the proofs of Lemma E.3.1 and Corollary E.3.2.

Proof of Lemma E.3.1. By the decodability assumption, for any $z_h = (o_{1:h}, a_{1:h-1})$ such that $\sup_{\pi} \mathbb{P}^{\pi}[z_h] > 0$, it holds that

$$\mathbb{P}(s_h \mid z_h) = \delta\left(\phi^{\star}(z_h)\right).$$

On the other hand, it holds that

$$\mathbb{P}(s_h \mid z_h) = \frac{\sum_{s_{h-1}} \mathbb{P}(s_h, o_h, s_{h-1} \mid o_{h-1:1}, a_{h-1:1})}{\sum_{s_{h-1}} \mathbb{P}(o_h, s_{h-1} \mid o_{h-1:1}, a_{h-1:1})}. \tag{E.2}$$

By the POMDP model assumption and decodability the numerator is also given by,

$$\mathbb{P}(s_h, o_h, s_{h-1} \mid o_{h-1:1}, a_{h-1:1}) = \mathbb{P}(s_h, o_h \mid s_{h-1}, a_{h-1})\delta(s_{h-1} = \phi^{\star}(z_{h-1})).$$

Similarly, the denominator is given by

$$\mathbb{P}(o_h, s_{h-1} \mid o_{h-1:1}, a_{h-1:1})\mathbb{P}(s_h \mid s_{h-1}, a_{h-1}) = \sum_{\bar{s}_h} \mathbb{P}(\bar{s}_h, o_h \mid s_{h-1}, a_{h-1})\delta(s_{h-1} = \phi^{\star}(z_{h-1})).$$

Plugging this back into equation Eq. (E.2) we obtain

$$\begin{aligned}
\mathbb{P}(s_h \mid z_h) &= \frac{\sum_{s_{h-1}} \mathbb{P}(s_h, o_h \mid s_{h-1}, a_{h-1}) \delta(s_{h-1} = \phi^\star(z_{h-1}))}{\sum_{s_{h-1}} \sum_{\bar{s}_h} \mathbb{P}(\bar{s}_h, o_h \mid s_{h-1}, a_{h-1}) \delta(s_{h-1} = \phi^\star(z_{h-1}))} \\
&= \frac{\mathbb{P}(s_h, o_h \mid \phi^\star(z_{h-1}), a_{h-1})}{\sum_{\bar{s}_h} \mathbb{P}(\bar{s}_h, o_h \mid \phi^\star(z_{h-1}), a_{h-1})} \\
&= \frac{\mathbb{P}(s_h \mid o_h, \phi^\star(z_{h-1}), a_{h-1}) \mathbb{P}(o_h \mid \phi^\star(z_{h-1}), a_{h-1})}{\sum_{\bar{s}_h} \mathbb{P}(\bar{s}_h \mid \phi^\star(z_{h-1}), a_{h-1}) \mathbb{P}(o_h \mid \phi^\star(z_{h-1}), a_{h-1})} \\
&= \frac{\mathbb{P}(s_h \mid o_h, \phi^\star(z_{h-1}), a_{h-1})}{\sum_{\bar{s}_h} \mathbb{P}(\bar{s}_h \mid \phi^\star(z_{h-1}), a_{h-1})} \\
&= \mathbb{P}(s_h \mid o_h, \phi^\star(z_{h-1}), a_{h-1}).
\end{aligned}$$

Recall that $\mathbb{P}(s_h \mid z_h) = \delta\left(s_h = \phi^\star(z_h)\right)$ by the decodability assumption. Hence, it holds that

$$\mathbb{P}(s_h \mid o_h, \phi^\star(z_{h-1}), a_{h-1}) = \delta(s_h = \phi^\star(z_h)).$$

Therefore for any reachable z_h, with $s_{h-1} = \phi^\star(z_{h-1})$ we take $b_h^\star(s_{h-1}, a_{h-1}, o_h)$ to be the unique s_h for which $\mathbb{P}(s_h \mid o_h, s_{h-1}, a_{h-1}) \neq 0$ and if this does not completely specify $b_h^\star$, we complete can complete it arbitrarily. □

Proof of Corollary E.3.2. The fact that $\pi^\star \in \Pi$ follows directly from Lemma E.3.1, since $\vec{b}^\star \in \mathcal{B}$ and for any H-step POMDP the optimal action depends only on the state. As for the size of Π observe that for each h we have $|\mathcal{B}_h| \leq S^{SOA}$ and so $|\vec{\mathcal{B}}| \leq S^{HSOA}$. Finally, for each $\vec{b} \in \vec{\mathcal{B}}$ we have $|\Pi_{\vec{b}}| = A^{SH}$. Taken together we have $|\Pi| \leq (SA)^{HSOA}$ as desired. □

E.4 Proof for Proposition 6.5.1

Here we construct an instance of a 2-step decodable POMDP in which the bellmank rank scales with the number of observations O. We further show that the OLIVE algorithm has sample complexity that scales polynomially with O, thus motivating our new algorithmic techniques. We believe a similer construction will also show that this model does not fall into either the bilinear class or Bellman-Eluder frameworks (Du et al., 2021; Jin et al., 2021).

The key idea is to use a construction inspired by the Hadamard matrix. Let $O = 2^s$ for some natural number s and $\mathcal{O} = \{1, \ldots, O\}$. Then, there exist sets $S_1, \ldots, S_{O-1} \subset \mathcal{O}$ such that:

$$\forall i : |S_i| = O/2, \quad \text{and} \quad \forall i \neq j : |S_i \cap S_j| = |S_i \cap \bar{S}_j| = O/4 \tag{E.3}$$

The existence of these can be verified by the existence and orthogonality of Hadamard matrices in dimension $O = 2^s$. Indeed, if we define $\{v_i\}_{i=0}^{O} \subset \{\pm 1\}^O$ such that $v_0 = \mathbf{1}$ and v_i is the ± 1 indicator vector for set S_i. Then the first property above is equivalent to $v_i^\top v_0 = 0$ for all $i \neq 0$ while the second property is equivalent to

$$\forall i \neq j \in \{1, \ldots, O\} \sum_k \mathbf{1}\{v_i[k] = +1\} v_j[k] = 0$$

We claim that these two properties are satisfied if the vectors v are the columns of a Hadamard matrix. The first follows directly from orthogonality. For the second, since $v_i^\top v_j = 0$ and $v_j^\top v_0 = 0$ both by orthogonality, we have

$$v_i^\top v_j = 0 \Rightarrow \underbrace{\sum_k \mathbf{1}\{v_i[k] = +1\} v_j[k]}_{=:A_{ij}} - \underbrace{\sum_k \mathbf{1}\{v_i[k] = -1\} v_j[k]}_{=:B_{ij}} = 0$$

$$v_j^\top v_0 = 0 \Rightarrow \sum_k \mathbf{1}\{v_i[k] = +1\} v_j[k] + \sum_k \mathbf{1}\{v_i[k] = -1\} v_j[k] = 0.$$

Thus we have $A_{ij} + B_{ij} = A_{ij} - B_{ij} = 0$ which implies that $A_{ij} = 0$. So we have established the existence of $O - 1$ sets satisfying Eq. (E.3).

Let us now put this construction to use in a 2-step decodable POMDP. We consider a $H = 2$, three state POMDP with initial state s_0 and two states s_1, s_2 reachable at time $h = 2$. We have: $\mathbb{O}(\cdot \mid s_0) = \text{Unif}(\{1, \ldots, O\})$ while $\mathbb{O}(\cdot \mid s_1) = \mathbb{O}(\cdot \mid s_2) = \delta(\{\perp\})$. In words, from the initial state we see an observation uniformly at random, while from s_1 or s_2 we see no observation. The dynamics are such that taking a_1 from s_0 reaches s_1 and taking a_2 from s_0 reaches s_2. Only a single action a_1 is available from s_1 or s_2 and it enjoys reward $R(s_1, a_1) = 1/2$, $R(s_2, a_1) = 3/4$. Clearly this

POMDP is 2-step decodable since the first state is always decodable and the previous action uniquely determines the second state.

We have a function class $\mathcal{F}$ of 2-step candidate Q functions. The functions are $\mathcal{F} := \{Q^\star\} \cup \{f_i\}_{i=1}^{O-1}$ where each f_i is associated with a set S_i from the above Hadamard construction. These functions are defined as

$$f_i(oa_1) = \mathbf{1}\{o \in S_i\}, \quad f_i(oa_2) = 3/4, \quad f_i(oa_1 \perp a_1) = \mathbf{1}\{o \in S_i\}, \quad f_i(oa_2 \perp a_1) = 3/4$$

It is easy to very that these functions have zero bellman error at the first time step, that is

$$\forall (o, a) : f_i(oa) = f_i(oa \perp a_1)$$

On the other hand, f_i has very high bellman error at the second time step, since it never correctly predicts the reward for state s_1. In particular we have $\mathbb{E}_{d_2^{\pi_{f_i}}}[f_i(oa \perp a_1) - r] = 1/4$, since π_{f_i} visits states s_1 on half of the observations and every time it does it overpredicts the reward by $1/2$. However, observe that

$$\mathbb{E}_{d_2^{\pi_{f_i}}}[f_j(oa \perp a_1) - r] = \frac{1}{O} \sum_{o \in S_i} \mathbf{1}\{o \in S_j\}(1 - 1/2) + \mathbf{1}\{o \notin S_j\}(0 - 1/2) = 0,$$

where the last identity uses Eq. (E.3). Thus we see that we have embedded an $(O-1) \times (O-1)$-sized identity matrix inside of the Bellman error matrix at time 2, which shows that the Bellman rank is $\Omega(O)$.

Note that the OLIVE algorithm itself will also incur $\mathrm{poly}(O)$ sample complexity in this instance. This is because the value predicted by f_i at the starting state, namely $\mathbb{E}[\max_a f(oa)]$, is $1/2 + 3/8$ which is greater than $V^\star = 3/4$. Thus OLIVE will enumerate over the f_i functions, eliminating one at a time and incurring a $\mathrm{poly}(O)$ sample complexity.