Large Scale Off-Policy and Offline Learning

Germano Gabbianelli

DOCTORAL THESIS UPF / 2022

THESIS SUPERVISORS
Dr. Gergely Neu

Dept. of Information and Communication Technologies

Abstract

Reinforcement Learning (RL), a subfield of machine learning and artifical intelligence, is a learning paradigm where an artificial agent learns to reach a predefined goal by trying to maximize a reward signal while interacting with the environment. In recent years RL has witnesses unprecedented breakthroughs, driven mainly by the integration of deep learning techniques. However, the deployment of RL algorithms in real-world scenarios poses challenges, particularly in environments where exploration is impractical or hazardous, such as autonomous driving or healthcare applications. Moreover, the current poor theoretical understanding of RL algorithms poses an additional limit to their usefulness in safety-critical scenarios.

This thesis focuses on the design of *provably efficient* algorithms for the settings of off-policy and offline learning. These paradigm constrain the agent to learn without directly receiving any feedback for its actions, and instead observing the rewards obtained by an other policy. In particular, the task of offline learning consists in learning a near-optimal policy only having access to a dataset of past interactions.

In summary, the theoretical exploration of off-policy and offline RL not only contributes to the broader understanding of RL algorithms but also offers a principled approach to training in scenarios where safety and reliability are paramount. The findings presented in this thesis aim to be a small step towards a broader adoption of RL in high-stakes environments, underpinned by robust theoretical frameworks and regret bounds.

Acknowledgments

Contents

1 Introduction **1**
- 1.1 Thesis Structure . . . 3
- 1.2 Notation . . . 4

2 Online Learning **5**
- 2.1 Online Learning . . . 5
- 2.2 Multi-armed Bandits . . . 9
- 2.3 Adversarial Bandits . . . 11
- 2.4 Contextual Bandits . . . 11

3 Reinforcement Learning **13**
- 3.1 Markov Decision Processes . . . 14

4 Large Scale Off-Policy and Offline Learning **19**
- 4.1 Online Learning with Off-Policy Feedback . . . 19
- 4.2 Offline Learning . . . 22
- 4.3 Large Scale Learning . . . 24
- 4.4 The Naïve Approach . . . 27
- 4.5 The Pessimism Principle . . . 30
- 4.6 Exploration and Coverage . . . 31
- 4.7 Coverage Definitions with Linear Rewards . . . 34
- 4.8 Main Contributions . . . 36

5 Online Learning with Off-Policy Feedback **39**
- 5.1 Preliminaries . . . 42

5.2 Known Behavior Policy . . . 43
5.3 Unknown Behavior Policy . . . 46
5.4 Linear Contextual Bandits . . . 47
5.5 Analysis . . . 51
5.6 Empirical Results . . . 54

6 Importance-Weighted Offline Learning 57
6.1 Preliminaries . . . 60
6.2 Pessimistic importance-weighted offline learning in contextual bandits . . . 60
6.3 Pessimism and Variance Reduction via Implicit Exploration 62
6.4 A PAC-Bayesian extension . . . 66
6.5 Adaptivity to the coverage . . . 69
6.6 Experiments . . . 70
6.7 Further details on the experiments . . . 72

7 Offline Learning in Linear Markov Decision Processes 77
7.1 Preliminaries . . . 81
7.2 Algorithm and Main Results . . . 83
7.3 Analysis . . . 88
7.4 Extension to Average-Reward MDPs . . . 90
7.5 Detailed Computations for Comparing Coverage Ratios . . 92

8 Conclusions 97
8.1 Online Learning with Off-Policy Feedback . . . 97
8.2 Importance-Weighted Offline Learning . . . 99
8.3 Offline Learning in Markov Decision Processes . . . 102

Bibliography 107

A Proofs for Chapter 5 119
A.1 The proof of Lemma 5.5.1 . . . 119
A.2 The proof of Theorem 5.3.1 . . . 120
A.3 The proof of Theorem 5.4.1 . . . 122
A.4 The proof of the regret decomposition of Equation (A.10) . 124

B Proofs for Chapter 6 127

B.1 The proof of Lemma 6.3.2 . . . 127
B.2 The proof of Lemma 6.3.3 . . . 128
B.3 The proofs of Lemmas 6.4.2 and 6.4.3 . . . 129

C Proofs for the discounted setting of Chapter 7 **131**
C.1 Proof of Lemma 7.3.1 . . . 131
C.2 Proof of Lemma 7.3.2 . . . 132
C.3 Regret bounds for stochastic gradient descent / ascent . . . 133

D Auxiliary Lemmas **137**

E Details for the Average-Reward MDP Setting **141**
E.1 Algorithm for average-reward MDPs . . . 145
E.2 Analysis . . . 147
E.3 Missing proofs for Lemma E.2.2 . . . 151

Chapter 1

Introduction

Artificial Intelligence (AI, Russell and Norvig 2020) is now ubiquitous. We use it in our smartphones when we unlock them using face recognition, or every time we take a picture. We use it when translating text from one language to an other, or when interacting with a voice assistant such as Siri, or Alexa. AI algorithms are used even inside modern CPUs architectures. Furthermore, AI algorithms found applications in many fields and industries, such as biology, healthcare, recommender systems, finance, games, and many more.

The advances in the field of Machine Learning (ML) are at the core of this rapid expansion. In particular, Deep Learning and Reinforcement Learning (RL) play an important role in many recent succesess. Machine learning is a field of AI focused on the development of algorithms that are able to perform tasks without being explicitly programmed, by learning to form generalizations from data. Deep learning (LeCun, Bengio, and Hinton 2015) denotes all the algorithmic methods employing multi-layered (i.e. *deep*) neural networks, and is the key ingredient to the incredible generalization capabilities of modern AI algorithms. Reinforcement learning is one of the three paradigms of machine learning, and focuses on the development of *learning agents*, which try to achieve a predefined goal by maximizing a reward signal, while interacting with their environment. Unlike most machine learning frameworks, RL considers *sequential* decision

making tasks in reactive environments, thus addressing concerns related to the implementation of learning systems in real-world applications. Specifically, the framework of RL allows modeling the interactions between the learning system and its environment, and algorithms allow accounting for long-term effects of decisions made by the learning system.

The strenghts of deep learning and reinforcement learning have been combined to achieve incredible breakthroughs in AI. In particular, deep learning brought the necessary generalization capabilities to reinforcement learning to enable learning in problems with very large state spaces. Two of the most notable examples being DQN (Mnih, Kavukcuoglu, Silver, Graves, et al. 2013; Mnih, Kavukcuoglu, Silver, Rusu, et al. 2015), a single algorithm that reached human-level performance on a set of 49 different arcade games by learning directly from the screen pixels, and AlphaGo (Silver et al. 2016), the first computer program to defeat the world champion of the acient Chinese game of Go.

However, these algorithms, as powerful and revolutionary as they might be, have some fundamental limitations. Deep learning techniques are unfortunately poorly understood from a theoretical point of view, and usually require a lot of data, which translates to RL agents requiring a lot of experience and thus learning very slowly. Furhtermore, Reinforcement learning agents learn by gathering experience while interacting with an environment, which inevitably requires making mistakes and performing suboptimal actions.

These limitations are side-stepped respectively by employing enormous amounts of compute resources to parallelize the training of algorithms, and by the use of simulators to avoid training in the real world. In particular, simulators provide two enormous advantages: they enable to greatly speed-up learning by increasing the simulation speed, and they completely eliminate the costs of learning (and thus doing mistakes) in a real-world environment.

Unfortunately, simulators are not always available, and the fact that in many real-world scenarios there is a huge cost associated in deploying a suboptimal policy, only aggravates the situation. A prime example of this are applications in healthcare, which must rely on historical data

of patients instead of performing real-time trials due to evident ethical concerns. Similar concerns may happen in settings making use of robots, where deploying a bad policy could result in expensive damage to the equipment.

In this thesis, we try to address some of these limitations by proposing algorithms with *strong theoretical guarantees*, which are able to learn in problems with *large state spaces*, and where there is *no control over exploration*. Specifically, we model in a precise mathematical way the problem of learning an optimal policy in tasks with very large state spaces. These modeling assumptions, while not completely closing the gap between theory and practical real-world scenarios, enable proving various kinds of rigorous properties of our algorithms. These are the computational time and memory complexity, and the sample complexity, which is the amount of experience required to learn an optimal policy. We are specifically interested in problems where the learning agent has no control over exploration, because they allow to model scenarios where there is no simulator, and there is a high cost of failure.

1.1 Thesis Structure

In Chapters 2 and 3 we briefly introduce the fundamental concepts of Online Learning and Reinforcement Learning. These are the two main research areas of our contribution.

In Chapter 4 we give a more detailed overviews of the specific settings studied in this thesis. That is learning in *large scale* problems, learning with *off-policy* feedback, and learning from *offline* data.

In the next three chapters, we present our main contributions. Concretely, in Chapter 5 we study the framework of online learning with off-policy feedback, where the learning agent cannot directly observe the outcome of its actions, but must instead observe the feedback obtained by another policy. This allows to model situations where the observations of the learning agent are unreliable. In Chapters 6 and 7 instead we propose algorithms for the offline learning setting, in which the learning agent cannot interact with the environment at all, and must instead learn an optimal policy using the observations collected by another policy.

Finally, in Chapter 8 we draw some conclusions for each of the presented contributions.

Following this, there are some chapters of appendix, dedicated to listing all the proofs omitted from the main text, and some other details.

1.2 Notation

We denote the set of probability distributions over a measurable set $\mathcal{S}$ as $\Delta(\mathcal{S})$, and the probability simplex in $\mathbb{R}^d$ as Δ_d.

We denote the scalar product of $x, y \in \mathbb{R}^d$ as $\langle x, y \rangle$ and use $\|\cdot\|_2$ to denote the Euclidean norm.

For a positive semi-definite matrix $A \in \mathbb{R}^{d \times d}$, we write $\lambda_{\min}(A)$ and $\operatorname{Tr}(A)$ to denote respectively its smallest eigenvalue and its trace.

We denote vectors with bold letters, such as $\mathbf{x} \doteq [x_1, \dots, x_d]^\top \in \mathbb{R}^d$, and use $\mathbf{e}_i$ to denote the i-th standard basis vector.

We interchangeably denote functions $f : \mathcal{X} \to \mathbb{R}$ over a finite set $\mathcal{X}$, as vectors $\mathbf{f} \in \mathbb{R}^{|\mathcal{X}|}$ with components $f(x)$, and use $\succeq$ to denote element-wise comparison.

Where possible, we use upper-case letters for random variables, such as S, and denote the uniform distribution over a finite set of n elements as $\mathcal{U}(n)$.

In the context of iterative algorithms, we use $\mathcal{F}_{t-1}$ to denote the sigma-algebra generated by all events up to the end of iteration $t-1$, and use the shorthand notation $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \,|\, \mathcal{F}_{t-1}]$ to denote expectation conditional on the history, and $\mathbb{P}_t[\cdot] = \mathbb{P}(\cdot \,|\, \mathcal{F}_{t-1})$.

For nested-loop algorithms, we write $\mathcal{F}_{t,i-1}$ for the sigma-algebra generated by all events up to the end of iteration $i-1$ of round t, and $\mathbb{E}_{t,i}[\cdot] = \mathbb{E}[\cdot \,|\, \mathcal{F}_{t,i-1}]$ for the corresponding conditional expectation.

Finally, we use the convention that $\prod_{k=i}^{j} = 1$ and $\sum_{k=i}^{j} = 0$ when $j < i$.

Chapter 2

Online Learning

In this chapter we are going to give a brief introduction to the field of Online Learning, with particular attention to the Multi-Armed Bandit problem and its variations.

For a deeper introduction to the topic, see the books of Slivkins (2019) and Lattimore and Szepesvári (2020).

In the whole manuscript, we are going to assume the existence of a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, whose outcomes $\omega \in \Omega$ encode all required randomness.

2.1 Online Learning

The term *online learning* refers to a large class of sequential decision making problems (often also referred to as "games"), where generally an *agent* (or learner) needs to repeatedly decide which action to perform, between a set of alternative choices $\mathcal{A}$, to maximize its total gain. The framework of online learning has been studied since the fifties and allows to model learning in a variety of real-world scenarios, such as advertisement, recommender systems and drug prescription.

In a somewhat general form, the problem can be formulated as follows.

Definition 2.1.1 (online learning game)**.** A generic online learning problem is a n-rounds sequential game between an agent and an adversary, where at each round, or time-step, $t \in [n]$:

- A context (or state) $X_t \in \mathcal{X}$ is drawn from a possibly unknown distribution ν and revelead to the agent
- The adversary chooses a reward function $g_t : \mathcal{X} \times \mathcal{A} \to \mathcal{R}$, mapping contexts and actions to a numerical reward (i.e. $\mathcal{R} \subseteq \mathbb{R}$)
- The agent picks an action $A_t \in \mathcal{A}$
- The agent obtains a reward $R_t = g_t(X_t, A_t)$.

The context (or state) represents some side information available to the agent at time-step t. For example, in a recommender system the context could represent the session data (language, location, preferences, etc) of the user currently using the system.

Definition 2.1.1 is very general, and if we are to design efficient algorithms, we need to specify some important details. For example, we need to know if the set of actions is assumed to be finite or infinite; if the agent and the adversary are allowed to randomize their choices of actions and rewards respectively; if the agent gets to observe only R_t or the full function g_t; and most importantly what is the precise objective of the agent. Different answers to these question lead to different online learning settings and different trade-offs in algorithm design. The subset of settings relevant for this work are detailed in Sections 2.2 to 2.4. However, we anticipate here that in this thesis we consider actions sets $\mathcal{A}$ of finite size, and never let the agent observe the full reward function g_t

We now present some notions that are relevant to all settings considered in this thesis.

Definition 2.1.2 (observations)**.** Let O_t represent the sequence consisting of all the quantities observed by the agent up to the moment before selecting action A_t in time-step t; and let $\mathcal{O}_t$ denote the set of all possible sequence of observations.

Typically, we have

$$O_t = (X_1, A_1, R_1, \dots, X_{t-1}, A_{t-1}, R_{t-1}, X_t). \tag{2.1}$$

This definition aligns well with the typical scenario in online learning, where the agent can observe its own rewards. Nevertheless, this is not the only possibility. Specifically, this thesis focuses on settings where the agent lacks direct access to its own rewards. Instead, the agent must rely on observations gathered by an other policy, either run in parallel or beforehand. Thus, it is important to keep in mind that o_t represents the information available to the agent at time-step t, and its concrete definition may vary from the one given above depending on the setting we consider.

We now give a very general definition of policies and decision rules, borrowed from Puterman (1994). These definitions are more general than what is strictly useful in the context of this chapter, but they allow us to give a single definition for these concepts.

Definition 2.1.3 (policies and decision rules)**.** The behavior of an agent is defined by a sequence of *decision rules*. This sequence is called the agent's *policy* and is denoted with the letter π. For a given time-step t, a decision rule, is a function $\pi_t : \mathcal{O}_t \to \Delta_{\mathcal{A}}$ mapping the observations of the agent up to time-step t to a distribution over actions. Concretely, we write $\pi_t(a|o_t)$ to denote the probability of selecting action a, after observing history o_t.

When a policy is constant with respect to time $\pi = (\pi_1, \pi_1, \dots, \pi_1)$, we say that it is *stationary*. However, the agent's policy is often *non-stationary*, meaning that the decision rules are not fixed for all time-steps t but they change over time as the agent learns. Moreover, Puterman (1994) refers to policies depending only on the last context (or state) as *markovian*, while the ones on the whole history (or parts thereof) are named *history dependent*.

The fact that a policy can be non-stationary, or a decision rule can depend on the whole history, is the most general scenario, and does not imply that this must be always the case. In fact, it is important to point out that in bandit problems, where the next context is not influenced by the actions of the learner, it is pointless to consider policies depending on more information than the current context x_t.

In summary, an agent acts according to a policy, which is possibly improved at every round t, and it can make use of its previous observations o_t to derive its current decision rule. Thus, at time-step t the agent draws its action according to $\pi_t(\cdot|o_t)$. However, when dealing with random variables, we often slightly abuse notation and write instead

$$\pi_t \doteq \pi_t(O_t).$$

This is just a convenient way to denote the mapping $\pi_t : \Omega \to \Delta_{\mathcal{A}}$ defined as $\omega \mapsto \pi_t(O_t(\omega))$, and the tilde is use to emphasize that the resulting distribution over actions is a random variable. Furthermore, we write π to refer to the whole sequence of random variables $(\pi_1, \dots, \pi_n)$. We almost always omit ω from our derivations, to make the notation lighter.

Intuitively, the objective of the agent is to gather as much reward as possible over many repeated games. Formally, the performance of the agent is defined with respect to a stationary comparator policy $\pi^* \in \Delta_{\mathcal{A}}$, and is measured in terms of *regret*, that is, as the difference in *expected return* (or value) obtained by the comparator and the agent.

Definition 2.1.4 (return)**.** Let G_t denote the *return* of the agent, defined as the cumulative sum of rewards obtained until time-step t:

$$G_t = R_1 + \cdots + R_t.$$

Moreover, let ρ denote the mean return obtained by the agent over an infinite number of games

$$\rho_t = \mathbb{E}\Big[G_t\Big] = \int_{\Omega} G_t(\omega)\mathbb{P}(d\omega)$$

When we need to distinguish between the actions taken, or the rewards received by more than one policy, we usually add a superscript π to the relevant quantities (i.e. a_t^π, r_t^π) to denote they correspond to policy π. However, we adopt the functional syntax $G_n(\pi)$ and $\rho_n(\pi)$ for the random return and its expectation, since these quantity appears very often.

Finally, we observe that according to Definition 2.1.1, the adversary is *adaptive*. That is, it can observe all the actions played by the agent up

to the previous round $t-1$ and use them to choose the reward function g_t for the current round. This makes the adversary very powerful and having almost no assumptions on how the rewards are generated makes it possible to model real-world scenarios where we have no information about the structure of the rewards or their distribution, and thus enables the derivation of very broadly applicable algorithms. One may wonder if giving so much freedom to the adversasry, would make all learning hopeless, since the adversary could, in principle, always choose "bad" rewards for all actions. However, this problem is prevented by our definition of the learning objective. In fact the goal of the agent is not defined as obtaining "high" reward in absolute terms, but to perform good with respect to a stationary comparator policy π^*. This implies that such a strategy would result in all actions being optimal, and thus the ardversary is required to implent a smarter strategy, trying to make optimal and suboptimal actions hard to distinguish, while keeping their reward gap significant.

Definition 2.1.5 (regret)**.** The objective of the agent is to minimize its *regret* with respect to a given stationary (non-history-dependent) policy $\pi^* : \mathcal{X} \to \Delta_{\mathcal{A}}$. The regret is defined as the difference in value between the the comparator policy π^*, and the policy of the agent

$$\mathfrak{R}(\pi; \pi^*) = \rho_n(\pi^*) - \rho_n(\pi).$$

This definition of regret using the value $\rho(\cdot)$ is very common, but has the downside of taking away all randomness, which makes it unsuitable to prove bounds in high probability.

For this reason, in parts of this thesis we are going to provide bounds on a version of regret, named *pseudo-regret*, which takes an expectation only with respect to some of the randomness involved in the process.

2.2 Multi-armed Bandits

The term *Multi-Armed Bandits* (MAB) comes from the following scenario, where a slot machine is referred to as a one-armed bandit:

> Consider a gambler who is presented with the opportunity to play any of n one-armed bandit machines. He whishes to allo-

> cate his successive plays amongst these machines to maximize his expected total-discounted reward. He does this one play at a time, on the basis of prior information and observations to date.
>
> — Weber (1992)

This term (sometimes abbreviated to just *bandits*) is used to refer to all the variants of the online learning problem where the agent can not observe the whole reward function, but usually only sees the reward for the action that was played. This form of feedback is thus called *bandit feedback*. However, the term "multi-armed bandits" is also used to refer to the special case of online learning with bandit feedback and *stochastic rewards*, which we are going to describe in this section. To avoid ambiguity we refer to this latter setting as Stochastic bandits.

Stochastic bandit are a special case of the online learning setting detailed in the previous section, where there are no contexts and where the reward functions g_t are not arbitrary, but realizations of a sequence of independent and identically distributed random functions $g_t : \Omega \to \mathcal{R}^{\mathcal{A}}$.

Definition 2.2.1 (Stochastic Bandit)**.** A stochastic bandit is a sequential game between an *agent* and the *environment*. It is denoted by the tuple $(\mathcal{R}, \mathcal{A}, \mathbb{P}_g, n)$ consisting of a set $\mathcal{R} \subseteq \mathbb{R}$ of possible rewards; a set $\mathcal{A}$ of actions among which the agent can choose; a probability distribution $\mathbb{P}_g : \Delta(\mathcal{R}^{\mathcal{A}})$ for the reward functions; and a positive integer n, called the *horizon* and denoting the total number of time-steps (or stages) the game is played for. Moreover, we denote the mean of $\mathbb{P}_g$ as $\bar{g}$.

The game is played for n time-steps. At each time-step t

- The environment samples a reward function $g_t : \mathcal{A} \to \mathcal{R}$ from $\mathbb{P}_g$
- The agent picks an action $A_t \in \mathcal{A}$
- The agent obtains reward $R_t = g_t(a_t)$

Notice that mean reward for each action $\bar{g}(a)$ does not change during the game. For this reason, in this setting it is common to choose as the comparator policy, the deterministic policy always playing the action a^*

with highest mean reward

$$a^* = \operatorname*{argmax}_{a \in \mathcal{A}} \bar{g}(a). \tag{2.2}$$

2.3 Adversarial Bandits

Building on the stochastic bandit setting detailed in the previous section, the setting of adversarial bandits drops the assumption that the reward functions are independent and identically distributed.

The agent is not competing anymore against a passive environment, but is instead facing an adversary that arbitrarily selects the rewards $g_t(a)$ for each action a and time-step t.

In the most general case, the adversary is assumed to be adaptive. That is, it is allowed to select the reward function g_t using all previous information, including the actions played by the agents in previous rounds. Because of this dependence on random quantities, in the adaptive case, it may be necessary to consider the reward functions as realizations of random variables g_t sampled from the unknown distributions $\mathbb{P}_{g_t}$.

2.4 Contextual Bandits

Contextual bandits generalize the multi-armed bandit settings seen so far by introducing a context X_t, observed by the agent at every time-step t. The role of the context is to model additional side information that is available to the agent. For example, when trying to design a recommender system, the context X_t could represent the information available about the current user. This in turn enables to design algorithms which select a context-dependent optimal action, as opposed to the settings of the previous sections which admit a single global optimal action.

Similarly to the stochastic bandit settings, here the agent plays against a passive *environment*, which selects rewards in an i.i.d. fashion according to a fixed distribution.

Definition 2.4.1 (contextual bandit)**.** A *contextual bandit* is a sequential-decision game between an agent and the environment. It is denoted by

the tuple $(\mathcal{R}, \mathcal{X}, \mathcal{A}, \nu, \mathbb{P}_g, n)$ composed of a set $\mathcal{R}$ of rewards; a finite set $\mathcal{A}$ of actions; a finite but potentially very large set of contexts $\mathcal{X}$; a probability distribution $\nu : \Delta(\mathcal{X})$ over contexts; a probability distribution $\mathbb{P}_g : \Delta(\mathcal{R}^{\mathcal{X}\times\mathcal{A}})$ for the rewards functions; and a positive integer n, called the horizon and denoting the total number of time-steps (or stages) the game is played for. As before, we denote the mean rewards with $\bar{g}$.

The interaction protocol, is as follows. At each time-step $t \in [n]$

1. A context X_t is drawn by the environment according to ν and revealed to the agent.
2. A reward function $g_t : \mathcal{X} \times \mathcal{A} \to \mathcal{R}$ is sampled according to $\mathbb{P}_g$
3. The agent picks an action $A_t \in \mathcal{A}$
4. The agent obtains reward $R_t = g_t(X_t, A_t)$

In this setting we are interested in finding the optimal action given each possible context $x \in \mathcal{X}$. For this reason we model policies as a function from the context-space to a distribution over actions, and we write $\pi(a|x)$ to denote the probability of picking action a given that the current context is x.

Chapter 3

Reinforcement Learning

Reinforcement Learning (RL) is the activity performed by an **agent** who tries to achieve a **goal**, by maximizing a **reward signal**, while interacting with an **environment**.

As an example, consider the game of Tetris: the agent is whoever is playing the game, either human or an algorithm; the goal is to clear all the levels; the reward signal to be maximized is the game's score, and the environment is the playing field.

In a Reinforcement Learning problem the agent interacts with the environment by performing some actions, observing what effect they have both on the environment and reward signal, and using this knowledge to improve its behavior.

For example, in the game of Tetris, the possible actions are moving the falling piece sideways, rotating it by $\pm 90°$, or waiting for it to fall. Therefore, an agent who does not have any previous knowledge of the game, should hopefully notice, after playing more or less randomly for a while, that when the fallen pieces are aligned to form a horizontal line without gaps, the score increases. A smart agent will consequentily try to improve its strategy by repeating the actions that led to an increase of the reward. Effectively using the reward signal as a positive *reinforcement* of the desired behaviour.

For a general introduction to the concepts of Reinforcement Learning see Sutton and Barto (2018).

3.1 Markov Decision Processes

Markov Decision Processes (MDP, Puterman 1994) are the main mathematical abstraction used to model full reinforcement learning problems. Differently from what happens in contextual bandits the current context (now called state) influences the probability distribution over the contexts for the following round. This makes the problem much harder, while at same time giving us the chance to model problems of greater complexity.

Definition 3.1.1 (Markov Decision Process)**.** A Markov Decision Process is a discrete-time stochastic process defined by the tuple $(\mathcal{X}, \mathcal{A}, \mathcal{R}, p, r, \nu_0)$, composed of a finite, but potentially very large, set of states $\mathcal{X}$; a finite set of actions $\mathcal{A}$; a bounded set of $\mathcal{R} = [0, 1]$; a probability distribution $p : \mathcal{X} \times \mathcal{A} \to \Delta_{\mathcal{X}}$ over next states, given the current state and action; a deterministic reward function $r : \mathcal{X} \times \mathcal{A} \to \mathcal{R}$ assigning a bounded reward to each state-action pair; and an initial state distribution $\nu_0 \in \Delta_{\mathcal{X}}$, used to sample the initial state of the process.

The process starts at time-step $t = 0$, in a state X_0 sampled from ν_0, and then at each time-step t

1. the agent observes the current state X_t and draws an action A_t from its current policy $\pi_t(\cdot|X_t)$,
2. the agent receives a reward $r(X_t, A_t)$ and is moved to the next state according to $X_{t+1} \sim p(\cdot|X_t, A_t)$

The transition function p is also denoted as the matrix $\mathbf{P} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times|\mathcal{X}|}$ and the reward as the vector $\mathbf{r} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|}$

Differently than all the settings considered in the previous sections, here we are interested in modeling scenarios where the horizon n is potentially infinite. For this reason, we have to exercise a bit more care in the choice of our learning objective. Infact defininining the return ρ_t simply as the

sums of the rewards, as done in Definition 2.1.4 could lead to the sum diverging.

For this reason we consider the discounted reward setting. Morevoer, it is possible to show (Puterman 1994, Section 5.5) that for this setting, it is sufficient to restrict our attention to markovian policies.

3.1.1 Discounted Reward

In the discounted reward setting the rewards are discounted by a *discount factor*, denoted as γ, and assumed to be part of the MDP definition.

Definition 3.1.2 (discounted return)**.** We define the return $G_{i:j}$ of the agent as the discounted sum of rewards obtained between time-step i and j:

$$G_{i:j} = R_i + \gamma R_{i+1} + \cdots + \gamma^{j-i} R_j = \sum_{k=i}^{j} \gamma^{k-i} R_k.$$

When working with MDPs, it is useful to define value functions representing the expected return (or value) obtained by a policy when starting from any given state x.

Definition 3.1.3 (value function)**.** Given any markovian policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$. Let $v^\pi : \mathcal{X} \to \mathbb{R}$ denote the expected value obtained by policy π after starting from $X_1 = x$

$$v^\pi(x) = \mathbb{E}\Big[G_{0:\infty} \,\Big|\, X_0 = x\Big] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t \,\middle|\, X_0 = x\right].$$

Similarly, it is possible to define the state-action value function, or Q-function for any policy π.

Definition 3.1.4. Given any markovian policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$. Let $q^\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ denote the expected value obtained by policy π after starting from state x and taking action a

$$q^\pi(x, a) = \mathbb{E}\Big[G_{0:\infty} \,\Big|\, X_0 = x, A_0 = a\Big].$$

As usual when the policy π is clear from the context, we are going to omit the superscript from these quantities.

The value function and Q-function are related as follows

$$\begin{aligned} v^\pi(x) &= \mathbb{E}\Big[G_\infty \,\Big|\, X_0 = x\Big] = \mathbb{E}\Big[\mathbb{E}\Big[G_\infty \,\Big|\, A_0\Big] \,\Big|\, X_0 = s\Big] \\ &= \mathbb{E}\Big[q^\pi(x, A_0) \,\Big|\, X_0 = x\Big] = \sum_{a\in\mathcal{A}} \pi(a|x) q^\pi(x,a). \end{aligned} \tag{3.1}$$

Theorem 3.1.5. *Let $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ denote any stationary markovian policy. Then*

$$\begin{aligned} q^\pi(x,a) &= r(x,a) + \gamma \sum_{x'\in\mathcal{X}} p(x'|x,a) \sum_{a'\in\mathcal{A}} \pi(a'|x') q^\pi(x',a') \\ v^\pi(x) &= \sum_{a\in\mathcal{A}} \pi(a|x)\Big[r(x,a) + \gamma \sum_{x'\in\mathcal{X}} p(x'|x,a) v^\pi(x')\Big] \end{aligned}$$

Proof.

$$\begin{aligned} q^\pi(x,a) &= \mathbb{E}\Big[R_0 + \gamma G_{1:\infty} \,\Big|\, X_0 = x, A_0 = a\Big] \\ &= r(x,a) + \gamma \sum_{x'\in\mathcal{X}} \mathbb{E}[G_{1:\infty} \mid X_1] \mathbb{P}(X_1 = x' \mid X_0 = x, A_0 = a) \\ &= r(x,a) + \gamma \sum_{x'\in\mathcal{X}} \mathbb{E}[G_{1:\infty} \mid X_1] p(x'|x,a) \\ &= r(x,a) + \gamma \sum_{x'\in\mathcal{X}} \mathbb{E}[G_{0:\infty} \mid X_0] p(x'|x,a) \\ &= r(x,a) + \gamma \sum_{x'\in\mathcal{X}} p(x'|x,a) v^\pi(x'), \end{aligned}$$

In the second equality we used the law of total probability, and in the second to last, we used the markov property. We use Equation (3.1) to complete the proof. □

Morevoer, it is useful to represent the value function and the state-action value function as vectors. We use respectively $\mathbf{v} \in \mathbb{R}^{|\mathcal{X}|}$ and $\mathbf{q} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|}$ for this purpose.

Definition 3.1.6 (optimality)**.** We say that a policy π^* is optimal when it attains the maximum value among all policies, starting from any state. That is,

$$v^{\pi^*}(x) = \sup_{\pi} v^{\pi}(x) \qquad \forall x \in \mathcal{X},$$

where the supremum is over the set of all markovian policies.

It is possible to show that an optimal policy statisfies the *Bellman optimality equations*

$$v^*(x) = \max_{a \in \mathcal{A}} \left[r(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) v^*(x') \right] \tag{3.2}$$

$$q^*(x,a) = r(x,a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x,a) \max_{a' \in \mathcal{A}} q^*(x',a'). \tag{3.3}$$

An other fundamental quantity is the discounted state occupancy measure of each policy π, defined as

$$\nu^{\pi}(x) = (1-\gamma)\mathbb{E}\left[\sum_{t=1}^{n} \gamma^t \mathbb{1}\{X_t = s\} \right], \tag{3.4}$$

and the state-action occupancy measure, dervied as

$$p^{\pi}(x,a) = \nu^{\pi}(x)\pi(a|x). \tag{3.5}$$

These quantities also have a vector representation, respectively as $\boldsymbol{\nu}^{\pi}$ and $\mathbf{p}^{\pi}$.

Finally, in this setting it is convenient to use the *normalized* expected return, defined as

$$\rho^{\pi} = (1-\gamma)\mathbb{E}[G_{0:\infty}], \tag{3.6}$$

which, given the above definitions, can be rewritten equivalently as

$$\rho^{\pi} = (1-\gamma)\langle \boldsymbol{\nu}_0, \mathbf{v}^{\pi} \rangle = \langle \mathbf{r}, \mathbf{p}^{\pi} \rangle. \tag{3.7}$$

Chapter 4

Large Scale Off-Policy and Offline Learning

In this chapter we introduce the concepts of off-policy and offline learning. We talk about the reasons these settings are interesting and list some applications. We explain the way in which they are mathematically formalized; their characteristics (coverage) and challenges (distributional shift); and some of the most-common approaches (iw estimation, pessimism).

The concepts described in this chapter are broadly applicable and relate to the contextual bandit as well as the reinforcment learning settings. Therefore, we interchangeably refer to the information X_t observed by the agent at each round t, as either context or state.

4.1 Online Learning with Off-Policy Feedback

One of the prevalent approaches to Reinforcement Learning is *on-policy* learning. The defining characteristic of this learning paradigm is that the policy guiding the interaction with the environment, usually called *behavior policy*, coincides with the policy we are trying to evaluate, usually called the *target policy*. Methods like SARSA that adhere to the Generalized Policy Iteration scheme (GPI) exemplify this approach. Once

these methods gather enough observations to estimate the policy's value or action-value function, they perform a policy improvement step.

This step, where an improved policy is derived from the current estimate of the value function, is crucial and delicate, due to the *exploration-exploitation* dilemma. It requires striking a balance between *exploring* new actions to discern their potential benefits, and *exploiting* known actions to accumulate immediate reward. Being overly greedy based on the current understanding of the Q-function might completely eradicate exploration. For this reason, methods often employ an ε-greedy policy, trying to find a good balance between exploration and exploitation. While this ensures continued exploration of all actions, it comes at a significant compromise: the policy's optimality is bounded by ε. In other words, by ensuring exploration, on-policy methods often effectively cap their achievable policy quality to being the best among the ε-greedy policies, never truly reaching the overall optimal deterministic policy.

A more flexible alternative, which includes the on-policy case as a special case, is known as *off-policy* learning. This paradigm consists in keeping the behavior policy and the target policy separate, and includes famous methods such as Q-learning. The main characteristic of *off-policy* learning is that the concerns of exploration and exploitation are decoupled and addressed separately by the behavior and target policy. This enables algorithms such as Q-learning to use a more exploratory ε-greedy behavior policy to ensure enough exploration is performed, while still learning about and converging to the optimal deterministic policy.

Off-policy learning is one of the most fundamental concepts in reinforcement learning, concerned with the problem of learning an optimal behavior policy given sample observations generated by a (most likely suboptimal) behavior policy. This setting comes with a unique set of challenges arising from the fact that the learning agent has no influence over the observed data, and thus classical methods for reducing uncertainty via exploration do not directly apply. The inability to explore may suggest that off-policy learning is better approached as a simple "pure exploitation" problem and can be potentially solved by a greedy approach—however, more thought reveals that an effective learning method should also attempt to account for the uncertainty of the random observations. Indeed, the problem set-

ting comes with multiple layers of uncertainty. One layer is represented by the potentially random choices made by the behavior policy, and another by the randomness in the observed rewards. The setting of *Online Learning with Off-Policy Feedback* presented in this section and studied in Chapter 5 lets us decouple these two uncertainties and address them individually. Concretely, we study the problem of online learning against an adversarial sequence of rewards, with off-policy feedback revealed by a stationary random policy.

This setting lies in the intersection of two distinct paradigms of sequential decision making: adversarial online learning and off-policy reinforcement learning. Formally, we study a sequential decision making problem where in each round, the learner has to pick one of K actions in order to maximize its total rewards. The sequence of reward assignments to actions are decided by an adversary, with each reward function determined the moment before the learner selects its action. The unique feature of the setting is that the learner does not get to observe its reward. However, the learner does observe the reward of another action that has been randomly sampled according to a behavior policy that remains fixed during the learning process. The goal of the learner is then to gain nearly as much reward as the best fixed comparator policy.

Definition 4.1.1 (Off-Policy Adversarial Bandits)**.** Concretely, we study a generalization of the Adversarial Bandit setting presented in Section 2.3, where the feedback provided to the agent consists of the actions and rewards obtained by the behavior policy μ.

At each round $t \in [n]$

- The adversary chooses a reward function $g_t : \mathcal{A} \to \mathcal{R}$, mapping each action to a numerical reward (i.e. $\mathcal{R} \subseteq \mathbb{R}$)
- The agent picks an action $A_t \in \mathcal{A}$
- The behavior policy samples an action $A_t^\mu \in \mathcal{A}$ according to μ
- The agent obtains a reward $R_t = g_t(A_t)$
- The behavior policy obtains reward $R_t^\mu = g_t(A_t^\mu)$
- The agent observes A_t^μ and R_t^μ

In this case the observations available to the agent at time-step t are defined as $O_t = (A_1, A_1^\mu, R_1^\mu, \ldots, A_{t-1}, A_{t-1}^\mu, R_{t-1}^\mu)$.

A concrete motivating example is the following. Consider running a large online advertisement company with a well-established system that is deployed on most of the traffic. The infrastructure of the company allows real-time measurements of the clickthrough rates generated by this system. Now, imagine that the research division is given access to some small amount of traffic where a new recommendation method can be deployed, but real-time logging is not reliable due to the lower volume of traffic assigned for experimentation. Thus, the decisions of the experimental recommendation system have to be driven by the real-time logs obtained from the original system on the main traffic, which may have poor coverage of some good actions that the new system can implement. In this example, the original system corresponds to the behavior policy and the experimental system corresponds to the policy of the learner.

4.2 Offline Learning

Offline Learning, often also called *batch learning*, refers to the task of learning an optimal policy, without being able to directly interact with the environment during the learning process. Rather, the learner must rely exclusively on a pre-existing dataset D, which consists of observations previously gathered by a behavior policy μ during its interactions with the environment. It is common to assume that the learning policy is known and stationary. However, in real-world scenarios the behavior policy is often unknown, or it is itself a learning agent thus behaving non-stationarily.

This approach can be undetrstood as a special case of off-policy learning, yet with a distinctive constraint: the absolute lack of online interaction. Every part of the exploration process has been previously carried out by the behavior policy, which dictates the scope and limitations of the learning process. The intrinsic challenge in offline learning is making the best use of this fixed dataset, which may or may not cover the necessary state-action pair to learn an optimal policy.

Despite its challenges, the setting of offline learning has received a lot

of recent attention. One reason is that it enables efficient learning in situations where interactions with the environment can be risky, expensive, or impratical. A prime example of this are applications in healthcare, which must rely on historical data of patients instead of performing real-time trials due to evident ethical concerns. Similar concerns may happen in settings making use of robots, where deploying a bad policy could result in expensive damage to the equipment. Additionally, considering the abundance of data available today, the ability to tap into vast datasets to derive near-optimal policies and the possibility to leverage the advances in supervised-learning techniques made this setting even more attractive.

Concretely, in Chapter 6 we design algorithms able to produce an ε-optimal policy given only access to a dataset of the form

$$D = (X_t^\mu, A_t^\mu, R_t^\mu)_{t=1}^n.$$

This dataset is collected by a fixed policy $\mu : \mathcal{X} \to \Delta(\mathcal{A})$ according to the contextual bandit protocol outlined in Section 2.4.

As specified in Section 2.4 we are interested in context-dependent policies. For this reason, an algorithm can be modeled as a function $\mathscr{f} : (\mathcal{X} \times \mathcal{A} \times \mathcal{R})^n \to \Delta(\mathcal{A})^{\mathcal{X}}$ taking as input a dataset D and returning a single stationary policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$.

Keeping in mind that in this setting no learning is allowed to happen online, we can equivalently model an algorithm as a single decision rule $\pi : \mathcal{O}_n \to \Delta(\mathcal{A})$, where $\mathcal{O}_n = (\mathcal{X} \times \mathcal{A} \times \mathcal{R})^n \times \mathcal{X}$. This emphasizes the fact that at each time-step t the agent can only use as observations the dataset D and the current context X_t.

Considering that the output policy is not history-dependent, there is no reason to test its performance for a number of time-steps $n > 1$. Thus, using the definitions above and trying to compute $\rho_1(\pi)$ according to Definition 2.1.4 we get

$$\rho_1(\pi) = \mathbb{E}\Big[R_1\Big] = \mathbb{E}\Big[g_1(X_1, A_1)\Big] = \mathbb{E}\left[\sum_{a \in \mathcal{A}} \pi(a|D, X_1)\bar{g}(X_1, a)\right]. \quad (4.1)$$

In the rest of the thesis, to make the notation lighter, we are often going to drop the superscript $^{\mu}$ from the random variables contained in the dataset D. Similarly we are going to simply write $\rho(\pi)$, dropping the $_1$ subscript.

This setting is extended to Reinforcement Learning, specifically to Linear MDPs in Chapter 7.

4.3 Large Scale Learning

In this section we direct our attention to the implications that the size of the context/state space has on the design of algorithms.

When designing an algorithm for a given learning problem, we are generally also interested in proving some kind of performance guarantees. Specifically, this is accomplished by demonstrating that our algorithms are able to find an approximatively optimal (i.e. ε-optimal) solution using a polinomial amount of memory (*memory complexity*), processing time (*time complexity*), and interactions with the environment (*sample complexity*).

It is very important to specify with respect to which quantities these bounds have to be polynomial. The size of the context space, in bandits, or the state space, in reinforcement learning, is one such important quantity. In some cases the size of the state space may be deemed small enough to be fitted into memory and to be explicitly iterated upon. Thus, we may be willing to accept a polynomial dependency on its size in our bounds. This set of assumptions is referred to as the *tabular* setting.

However, in many real-world scenarios the context space is too large to be efficiently processed or stored in memory. Imagine, for example, a movie recommender system using as context the last 10 movies seen by the user. Assuming there are 1024 movies in the catalog, we obtain 2^{100} possible contexts, which would require 2^{47} petabytes to be stored.

In these cases, we cannot have any quantity depending linearly (or worse) on the size of the state space $|\mathcal{X}|$. Specifically, neither the sample complexity, the space complexity, nor the computational complexity should

have a worse-than-linear dependence on $|\mathcal{X}|$. This means that a logarithmic dependence is usually accepted (as 100 is a much smaller constant than 2^{100}).

In Sections 4.3.1 and 4.3.2, we discuss two popular assumptions, which are also used as the basis in our contributions, to enable learning in very large state spaces.

4.3.1 Learning with Linear Rewards

One way to make it possible to design efficient algorithms while respecting this constraint, is to make assumptions on the structure of the rewards. Concretely, it is very common to assume that the mean rewards are linear and of the form

$$\bar{g}(x,a) = \langle \theta^*, \varphi(x,a) \rangle \qquad \forall x, a \tag{4.2}$$

where $\theta^* \in \mathbb{R}^d$ is an unknown d-dimensional vector, and $\varphi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is a known feature map, mapping every state-action pair to a lower d-dimensional space. With this assumption, we can design algorithms depending on the feature dimension d, which is assumed to be much smaller than the dimension of the state space $|\mathcal{X}|$.

4.3.2 Learning with a Policy Class and a Computational Oracle

In this section we present an alternative learning framework, which does not require making structual assumptions about the environment. Indeed, making assumptions on the structure of the rewards, or the transition function may be convenient to prove bounds. However, the applicability of such assumptions is sometimes not easy to verify in practice.

For this reason, a common alternative (Dudík et al. 2011; Agarwal et al. 2014; L. Wang, Krishnamurthy, and Slivkins 2023) is to assume access to a *policy class* $\Pi \subseteq \{\pi : \mathcal{X} \to \Delta(\mathcal{A})\}$ and a *computational oracle*. The policy class may be very large but it is usually assumed to be finite. Precisely, we assume that the cardinality $|\Pi|$ of the policy class is too large to have a linear dependence on it in our bounds. However, as before, a logarithmic dependence is deemed acceptable.

There are ways to deal with infinite policy classes, by having bounds that depend on the covering number (Swaminathan and Joachims 2015) or the Natarajan dimension (Y. Jin, Ren, et al. 2022) of the policy class. In Chapter 6, we extend our work to infinite policy classes making use of PAC-Bayesian generalization bounds (McAllester 1998; Audibert 2004; Catoni 2007).

Having a finite policy class significantly simplifies proving sample complexity bounds, because union bounds covering the whole policy class become possible. However, since the policy class is too big to be iterated on, the computational aspect, both in terms of memory and time, is not immediately simplified by this assumption.

Langford and T. Zhang (2007); Dudík et al. (2011) and the line of works that followed, managed to tackle the issue by reducing the problem to a series of well-studied supervised learning problems, which can be solved efficiently. Precisely, Dudík et al. (2011) show how to reduce the offline contextual bandit problem to cost-sensitive classification, by making use of the following oracle.

Definition 4.3.1 ($\mathcal{AMO}$ Oracle)**.** For a set of policies Π, an argmax oracle ($\mathcal{AMO}$ for short), is an algorithnm, which for any sequence $\{(x_t, y_t)\}_{t=1}^n$, where $x_t \in \mathcal{X}$ and $y_t \in \mathbb{R}^{|\mathcal{A}|}$, computes

$$\operatorname*{argmax}_{\pi \in \Pi} \sum_{t=1}^{n} y_t(\pi(x_t)).$$

Their definition works for deterministic policies only. Moreover, the policy returned by the oracle can be seen as the optimal cost-sensitive classifier on the given data, when interpreting the rewards of each action as negative costs associated with misclassification errors.

In this thesis we make use of a slightly generalized version of the previous oracle, which allows stochastic policy classes.

Definition 4.3.2 (CSC oracle)**.** We assume access to a computational oracle that can return optimal policies given an appropriately defined input dataset. Precisely, the oracle takes as input a dataset $\{x_t, y_t\}_{t=1}^n$ with

contexts $x_t \in \mathcal{X}$ and gains $y_t \in \mathbb{R}^{\mathcal{A}}$, and returns

$$\operatorname*{argmax}_{\pi \in \Pi} \sum_{t=1}^{n} \sum_{a} \pi(a|x_t) y_t(a).$$

We used the notation y_t for the rewards passed as input to the oracle, to avoid confusion with the rewards associated to the current bandit instance.

4.4 The Naïve Approach

A natural approach for off-policy and offline learning, is to define an estimator $\hat{v}(\pi)$ for the value $\rho(\pi)$ of each policy π, and to find and return the policy $\hat{\pi}$ that maximizes it.

Let us assume that for every policy $\pi \in \Pi$ we can bound in high probability the error of our estimator as

$$\rho(\pi) - \hat{v}(\pi) \leq A(\pi) \qquad \text{(underestimation error, 4.3)}$$

$$\hat{v}(\pi) - \rho(\pi) \leq B(\pi) \qquad \text{(overestimation error, 4.4)}$$

Then, our algorithm can be defined as

$$\hat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \hat{v}(\pi). \qquad \text{(naïve algorithm, 4.5)}$$

We can then prove the following result.

Lemma 4.4.1 (Regret of the naive algorithm)**.** *Given any comparator policy $\pi^* \in \Pi$, the expected regret of the naive algorithm can be bounded as*

$$\rho(\pi^*) - \rho(\hat{\pi}) \leq A(\pi^*) + \max_{\pi \in \Pi} B(\pi).$$

Proof.

$$\begin{aligned}
\rho(\pi^*) &\leq \hat{v}(\pi^*) + A(\pi^*) \\
&\leq \hat{v}(\hat{\pi}) + A(\pi^*) \\
&\leq \rho(\hat{\pi}) + A(\pi^*) + B(\hat{\pi}_n) \\
&\leq \rho(\hat{\pi}) + A(\pi^*) + \max_{\pi \in \Pi} B(\pi)
\end{aligned}$$

□

The first inequality is follows from the underestimation error bound of Equation (4.3); the second follows from the definition of the algorithm in Equation (4.5), and the third from the overestimation error bound of Equation (4.4).

Notice that we cannot significantly improve the last step, where we bounded $B(\hat{\pi})$ with $\max_{\pi \in \Pi} B(\pi)$, because it is always possible to construct a problem instance where the two terms coincide.

To explain why this bound is not very satisfying, in the following section we will study a concrete instance of the estimator $\hat{v}(\pi)$ for the setting introduced in Section 4.3.2.

4.4.1 Importance-Weighted Estimation

The simplest possible estimator one can think of is the *importance-weighted* (IW) value estimator (Horvitz and Thompson 1952) defined for each policy π as

$$\hat{v}^{\mathrm{IW}}(\pi) = \frac{1}{n} \sum_{t=1}^{n} \frac{\pi(A_t|X_t)}{\mu(A_t|X_t)} \cdot R_t. \qquad \text{(IW, 4.6)}$$

This estimator is also known as the inverse-probability weighting estimator (L. Wang, Krishnamurthy, and Slivkins 2023) or inverse-propensity weighted estimator (Y. Jin, Ren, et al. 2022), both abbreviated as IPW.

It can be easily shown that this estimator is unbiased

$$\begin{aligned}
\mathbb{E}[\hat{v}^{\mathrm{IW}}(\pi)] &= \mathbb{E}\left[\frac{1}{n} \sum_{t=1}^{n} \frac{\pi(A_t|X_t)}{\mu(A_t|X_t)} \cdot g_t(X_t, A_t)\right] \\
&= \mathbb{E}\left[\frac{1}{n} \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \mu(a|X_t) \frac{\pi(a|X_t)}{\mu(a|X_t)} \cdot \bar{g}(X_t, a)\right] = \mathbb{E}\left[\sum_{a \in \mathcal{A}} \pi(a|X_1) \bar{g}(X_1, a)\right]
\end{aligned}$$

We can use this estimator for the naïve algorithm of Equation (4.5), and compute the output policy using an oracle, such as the one of Definition 4.3.2. It is sufficient to make a single call the oracle with the dataset

$\{X_t, \tilde{y}_t\}_{t=1}^n$, where the reward vectors are defined as

$$y_t(a) = \frac{\mathbb{1}\{A_t = a\}}{\mu(A_t|X_t)} R_t.$$

The fact that the two objectives are the same can be verified with a simple calculation:

$$\begin{aligned}\sum_{t=1}^n \sum_{a\in\mathcal{A}} \pi(a|X_t) y_t(a) &= \sum_{t=1}^n \sum_{a\in\mathcal{A}} \mathbb{1}\{A_t = a\} \frac{\pi(a|X_t)}{\mu(A_t|X_t)} R_t \\ &= \sum_{t=1}^n \frac{\pi(A_t|X_t)}{\mu(A_t|X_t)} R_t = \hat{v}^{\mathrm{IW}}(\pi).\end{aligned}$$

To use Lemma 4.4.1 to obtain a regret bound, we first need to bound in high probability the error of our estimator $\hat{v}^{\mathrm{IW}}$. We can accomplish this by using some concentration inequalities, such as Bernstein's, which is reported here for convenience.

Lemma 4.4.2 (Bernstein's corollary)**.** *Let $Z_1, \dots Z_n$ be a collection of n idependent and identically distributed random variables. Assume that all the random variables are bounded by a constant $\alpha > 0$*

$$\sup_i |Z_i| \leq \alpha \qquad a.s.$$

Then, for any $\delta \in (0, 1)$, the difference between the sample mean and its expected value can be bounded with probability at least $1 - \delta$ as

$$\left| \mathbb{E}[Z_1] - \frac{1}{n} \sum_{t=1}^n Z_t \right| \leq \sqrt{\frac{2\mathbb{V}[Z_1] \log(1/\delta)}{n}} + \frac{2\alpha \log(1/\delta)}{3n}.$$

For convenience, let us define $\hat{r}_t(\pi) \doteq R_t \cdot \pi(A_t|X_t)/\mu(A_t|X_t)$. Then, using Lemma 4.4.2 with $Z_t = \hat{r}_t(\pi)$ gives us a symmetric error bound for our estimator $\hat{v}_n(\pi)$

$$A(\pi) = B(\pi) = \sqrt{\frac{2\mathbb{V}[\hat{r}_1(\pi)] \log(1/\delta)}{n}} + \frac{2\alpha \log(1/\delta)}{3n}.$$

Finally, using an union bound over the class of policies Π and applying Lemma 4.4.1 yields the following guarantee.

Theorem 4.4.3 (naive)**.** *Let $\delta \in (0,1)$ and $\pi^* \in \Pi$ be any comparator policy. Then, the regret of the naïve algorithm, using the IW estimator can be bounded with probability at least $1-\delta$ as*

$$\rho(\pi^*) - \rho(\widehat{\pi}_n) \lesssim \sqrt{\frac{2\beta \log(|\Pi|/\delta)}{n}} + \frac{2\alpha \log(|\Pi|/\delta)}{3n},$$

where we denoted with α and β respectively

$$\alpha = \sup_{x,a} \frac{1}{\mu(a|x)} \qquad \beta = \sup_{\pi \in \Pi} \mathbb{V}[\widehat{r}_1(\pi)].$$

The problem with this result lies exactly in the quantities α and β appearing in the bound. For large policy classes and state spaces, those quantities are to be considered unbounded, for all practical purposes. It is sufficient that the behavior policy μ does not visit a single state for the whole bound to lose meaning.

4.5 The Pessimism Principle

In recent years, a range of ideas have been proposed to improve the naïve approach presented above. The most widely adopted approach, first proposed by Swaminathan and Joachims 2015, and later elaborated on in a variety of contexts by works like (London and Sandler 2019; Y. Jin, Z. Yang, and Z. Wang 2021; Rashidinejad, B. Zhu, et al. 2021; Y. Jin, Ren, et al. 2022; G. Li, Ma, and Srebro 2022), involves selecting a *pessimistic* policy with the goal of reducing the random fluctuations.

As before, this approach requires designing an estimator $\widehat{v}(\pi)$ with bounded error (Equations (4.3) and (4.4)). Usually this property is required to hold in high probability, and is achieved through the use of standard concentration inequalities such as Bernstein's.

However, they key difference from the naïve approach is that now the algorithm computes and returns the best pessimistic policy

$$\widehat{\pi} = \operatorname*{argmax}_{\pi \in \Pi} \Big(\widehat{v}(\pi) - B(\pi) \Big). \qquad \text{(pessimistic algorithm)} \qquad (4.7)$$

We call this pessimistic, because the overstimation bound (Equation (4.4)) guarantees that $\hat{v}(\pi) - B(\pi)$ is smaller than the true value $\rho(\pi)$ for any policy π.

We are now ready to bound the regret of the pessimistic algorithm (Equation (4.7)) with respect to any comparator policy π^*.

Lemma 4.5.1 (bound for the pessimistic algorithm)**.** *Given any comparator policy $\pi^* \in \Pi$. The regret of the pessimistic algorithm (Equation (4.7)) can be bounded as*

$$\mathfrak{R}(\hat{\pi}; \pi^*) \leq A(\pi^*) + B(\pi^*).$$

Proof.

$$\rho(\hat{\pi}) \geq \hat{v}(\hat{\pi}) - B(\hat{\pi}) \geq \hat{v}(\pi^*) - B(\pi^*) \geq \rho(\pi^*) - A(\pi^*) - B(\pi^*)$$

□

Where in the first and last step we used Equation (4.4), and for the second step we used the definition of the algorithm (Equation (4.7)).

This result is a big improvement with respect to Lemma 4.4.1, because we got rid of the maximum over the whole policy class on the right-hand side, and the bound now only depends (in principle) with quantities depending on the comparator policy π^*.

4.6 Exploration and Coverage

We have seen in the previous sections that results often depend on the comparator π^* and the behavior policy μ, and on the subset of the state-action space that they explore.

Infact, the quality of the observed data is of fundamental importance to guarantee the optimality of the learned policy. However, different algorithms pose different requirements on the structure of the data.

These requirements are commonly posed in terms of the "overlap" (or *coverage*) between the comparator policy and the behavior policy. There are many different defintions of coverage in the literature. Here we give

a general definition, and then show how to instantiate it to recover the quantities of interest.

Definition 4.6.1 (coverage ratio)**.** Given a weight vector $\mathbf{w} \in \mathbb{R}^{|\mathcal{X}||\mathcal{A}|}$, and a positive constant $p \in \mathbb{R} \cup \{\infty\}$. We define the generalized *policy coverage ratio* between any policy π and the behavior policy μ as a weighted norm between their state-action occupancy measures

$$\mathcal{C}_{\mathbf{w},p}(\pi) = \left(\sum_{x,a} \mathbf{w}(x,a) \left(\frac{p^\pi(x,a)}{p^\mu(x,a)} \right)^p \right)^{1/p} = \left\| \frac{p^\pi}{p^\mu} \right\|_{\mathbf{w},p}$$

The coverage ratio can be seen as a notion of similarity between π and μ: it is minimized when the two policies are equal, and otherwise grows to infinity as the two policies drift apart.

Some common specializations of Definition 4.6.1 are given below. The first is dervied by setting $\mathbf{w}$ to be equal to the state-action occupancy measure p^π of policy π[1]

$$C(\pi) \doteq \mathcal{C}_{p^\pi,1}(\pi) = \mathbb{E}_{X,A \sim p^\pi} \left[\frac{p^\pi(X,A)}{p^\mu(X,A)} \right]. \tag{4.8}$$

Other two common variations are derived by setting $\mathbf{w}$ respectively proportional to the *state* occupancy measure of π, and the one vector $\mathbf{1}$

$$C^\diamond(\pi) \doteq \mathcal{C}_{\nu^\pi,1}(\pi) = \mathbb{E}_{X \sim \nu^\pi} \left[\sum_{a \in \mathcal{A}} \frac{p^\pi(X,a)}{p^\mu(X,a)} \right] \tag{4.9}$$

$$C^\dagger(\pi) \doteq \mathcal{C}_{\mathbf{1},1}(\pi) = \sum_{x,a} \frac{p^\pi(x,a)}{p^\mu(x,a)} \tag{4.10}$$

We assigned a name to these quantities to refer to them easily later. One important difference between them, is that the first reduces to 1 when the two policies coincide, while the other two reduce to $|\mathcal{A}|$ and $|\mathcal{X}||\mathcal{A}|$ respectively.

[1] In the contextual bandit setting p^π reduces to $\nu(x)\pi(a|x)$.

Let us now have a look at what those quantities reduce to in the setting of contextual bandits:

$$C(\pi) = \mathbb{E}_{A\sim\pi(\cdot|X)}\left[\frac{\pi(A|X)}{\mu(A|X)}\right], C^{\diamond}(\pi) = \mathbb{E}\left[\sum_{a\in\mathcal{A}}\frac{\pi(a|X)}{\mu(a|X)}\right], C^{\dagger}(\pi) = \sum_{x,a}\frac{\pi(a|x)}{\mu(a|x)}.$$

Here we assumed that the context X is sampled according to the context distribution ν.

In the stochastic bandit case (i.e. in the absence of contexts) we are going to consider

$$C^{\diamond}(\pi) = C^{\dagger}(\pi) = \sum_{a\in\mathcal{A}}\frac{\pi(a)}{\mu(a)}.$$

An other very important version of the coverage ratio, is the L-∞ *coverage ratio* between policy π and μ, and is defined by setting $\mathbf{w}$ to the vector of all ones $\mathbf{1}$ and p to ∞

$$C_{\infty}(\pi) = \mathcal{C}_{\mathbf{1},\infty}(\pi) = \sup_{x,a}\frac{p^{\pi}(a|x)}{p^{\mu}(a|x)}, \tag{4.11}$$

where the range of the sup is to be intended restricted to the support of π. In the contextual bandit case this quantity reduces to the ratio between the two policies, and it is also referred to as the *worst-case density ratio* (L. Wang, Krishnamurthy, and Slivkins 2023) between π and μ.

One common assumption is to require the behavior policy to sufficiently explore all the possible states and actions (B. Zhang et al. 2012; Y. Zhao et al. 2012; Swaminathan and Joachims 2015; Y.-Q. Zhao et al. 2015; X. Zhou et al. 2017; Kallus 2018; Kitagawa and Tetenov 2018; Athey and Wager 2021; R. Zhan et al. 2021; Z. Zhou, Athey, and Wager 2023). This assumption, called *uniform* (or *full*) *coverage*, guarantees that the learner can correctly estimate the value of any action for any state. It can be stated as follows.

Assumption 4.6.2 (uniform coverage)**.** We assume that the worst-case density ratio between *any* policy $\pi \in \Pi$ and the behavior policy μ is upper-bounded by a constant $\alpha \in \mathbb{R}$. That is

$$\exists \alpha \in \mathbb{R} \quad \text{s.t.} \quad \forall \pi \in \Pi \qquad \sup_{x,a}\frac{\pi(a|x)}{\mu(a|x)} \leq \alpha.$$

However, this assumption is considered to be very strong, and is hardly satisfied in practice, because policies, especially those close to optimality, tend to concentrate on a subset of "good" actions and states.

Another, more desirable, assumption is called *partial coverage* and requires the behavior policy to only explore the subset of states and actions visited by the policies we want to compete with (typically, the optimal policy).

Assumption 4.6.3 (partial coverage)**.** Given a comparator policy $\pi^* \in \Pi$ of interest, the coverage ratio between π and μ is bounded. That is,

$$C(\pi^*) < \infty.$$

They key difference with the uniform coverage assumption is that only the coverage with respect to the comparator needs to be bounded, as opposed to requiring a bound on the coverage of all policies. The specific coverage used in the assumption is slightly less important. Here we used $C(\pi)$, but one can have a partial coverage assumption using any of the other definitions (e.g. $C^\diamond, C^\dagger, C_\infty$).

The quantities appearing in these assumptions, such as α, $C(\pi^*)$ and $C_\infty(\pi^*)$ are often found in the regret bounds of algorithms. Having a bound that scales with $C(\pi)$ or a similar coverage measure, is ideal because the bound automatically adapts to the quality of the data, giving better guarantees against well-covered policies. On the other hand, having a bound scaling with α is not great, and should be avoided, especially if we assume the context space to be very large.

It should also be noted that having a bound depending on $C_\infty(\pi^*)$ in a setting where the context/state space is assumed to be very large or infinite should really be avoided.

4.7 Coverage Definitions with Linear Rewards

All these definitions and assumptions can be translated to settings making assumptions on the structure of the environment, such as the linear bandit setting discussed in Section 4.3.1, or the linear MDP setting.

Equation (4.8) or Equation (4.11) could be used in this case as well. However, these quantities become unbounded when there exists even a tiny set of contexts where the two policies have no overlap. Intuitively, it should be possible to learn a good policy even when there are states where the policies pick different actions, as long as they are aligned in the feature space in an appropriate sense. To make this intuition formal, we will showcase and compare different notions of "feature coverager ratios" used in the state of the art.

We start by introducing the matrix $\mathbf{\Lambda}_\pi \in \mathbb{R}^{d\times d}$ for each policy π, which is a very common quantity in this line of works:

$$\mathbf{\Lambda}_\pi = \mathbb{E}_{X,A\sim p^\pi(\cdot|X)}[\varphi(X,A)\varphi(X,A)^\intercal]. \tag{4.12}$$

We can then show our first definition of coverage for the linear setting

$$C^\dagger(\pi) = \mathbb{E}_{X,A\sim p^\pi}\left[\varphi(X,A)^\intercal \mathbf{\Lambda}_\mu^{-1}\varphi(X,A)\right] = \mathrm{Tr}\big(\mathbf{\Lambda}_\mu^{-1}\mathbf{\Lambda}_\pi\big), \tag{4.13}$$

where last equality follows from the property of trace $\langle a, b\rangle = \mathrm{Tr}(ba^\intercal)$. This is one of the definitions we use in Chapter 5, and an other way to write it is

$$C^\dagger(\pi) = \mathbb{E}_{X,A\sim p^\pi}\left[\|\mathbf{\Lambda}_\mu^{-1}\varphi(X,A)\|_2^2\right].$$

We denoted this quantity with $C^\dagger$, because it is equivalent to Equation (4.10) when reducing the linear reward setting to the tabular setting, by taking $d = |\mathcal{X}||\mathcal{A}|$ and $\varphi(x,a)_i = \mathbb{1}\{x, a = i\}$.

Similarly, we can define the notion of coverage we use in Chapter 7.

Definition 4.7.1 (generalized feature coverage ratio)**.** Let $c \in \{1/2, 1\}$. We define the generalized coverage ratio as[2]

$$C_{\varphi,c}(\pi) = \bar{\varphi}_\pi^\top \mathbf{\Lambda}_\mu^{-2c}\bar{\varphi}_\pi = \mathrm{Tr}\big(\mathbf{\Lambda}_\mu^{-2c}\bar{\varphi}_\pi\bar{\varphi}_\pi^\intercal\big).$$

where $\bar{\varphi}_\pi \doteq \mathbb{E}_{X,A\sim p^\pi}[\varphi(X,A)]$.

Notice how this definition is equivalent to $C(\pi)$ when using $c = 1/2$.

We compare in detail these definitions of coverage, and others used in the state of the art, in Section 8.3 and Section 7.5.

[2]When $\mathbf{\Lambda}_\mu$ is not invertible but $\bar{\varphi}_{\pi^*}$ is in the column space of $\mathbf{\Lambda}_\mu$, we can define the coverage ratio using the Moore-Penrose pseudoinverse, and set it to $+\infty$ otherwise.

4.8 Main Contributions

In Chapter 5 we present our first contribution. We study the problem of online learning in adversarial bandit problems under off-policy feedback. In this sequential decision making problem, the learner cannot directly observe its rewards, but instead sees the ones obtained by another unknown policy run in parallel (behavior policy). Instead of a standard exploration-exploitation dilemma, the learner has to face another challenge in this setting: due to limited observations outside of their control, the learner may not be able to estimate the value of each policy equally well. To address this issue, we propose a set of algorithms that guarantee regret bounds that scale with a natural notion of mismatch between any comparator policy and the behavior policy, achieving improved performance against comparators that are well-covered by the observations. We also provide an extension to the setting of adversarial linear contextual bandits, and verify the theoretical guarantees via a set of experiments. Our key algorithmic idea is adapting the notion of pessimistic reward estimators that has been recently popular in the context of off-policy reinforcement learning.

In Chapter 6 we study the problem of offline policy optimization in stochastic contextual bandits. The goal is to learn a near-optimal policy based on a dataset of decision data collected by a suboptimal behavior policy. Rather than making any structural assumptions on the reward function, we assume access to a given policy class and aim to compete with the best comparator policy within this class. In this setting, a standard approach is to compute importance-weighted estimators of the value of each policy, and select a policy that minimizes the estimated value up to a "pessimistic" adjustment subtracted from the estimates to reduce their random fluctuations. In this thesis, we show that a simple alternative approach based on the "implicit exploration" estimator of Neu (2015) yields performance guarantees that are superior in nearly all possible terms to all previous results. Most notably, we remove an extremely restrictive "uniform coverage" assumption made in all previous works. These improvements are made possible by the observation that the upper and lower tails importance-weighted estimators behave very differently from each other, and their careful control can massively improve on previous results that were all based on symmetric two-sided concentration inequalities. We also

extend our results to infinite policy classes in a PAC-Bayesian fashion, and showcase the robustness of our algorithm to the choice of hyper-parameters by means of numerical simulations.

Finally, in Chapter 7 we present our third contribution. We study the problem of offline learning in the context of Reinforcement Learning. This problem has attracted a lot of attention recently, but most existing methods with strong theoretical guarantees are restricted to finite-horizon or tabular settings. In contrast, few algorithms for infinite-horizon settings with function approximation and minimal assumptions on the dataset are both sample and computationally efficient. Another gap in the current literature is the lack of theoretical analysis for the average-reward setting, which is more challenging than the discounted setting. In this thesis, we address both of these issues by proposing a primal-dual optimization method based on the linear programming formulation of RL. Our key contribution is a new reparametrization that allows us to derive low-variance gradient estimators that can be used in a stochastic optimization scheme using only samples from the behavior policy. Our method finds an ε-optimal policy with $O(\varepsilon^{-4})$ samples, while being computationally efficient for infinite-horizon discounted and average-reward MDPs with realizable linear function approximation and partial coverage. Moreover, to the best of our knowledge, this is the first theoretical result for average-reward offline RL.

Chapter 5

Online Learning with Off-Policy Feedback

In this chapter, we study the setting of online learning with off-policy feedback as outlined in Chapter 4, and in particular in Section 4.1.

Our main contribution is an online learning algorithm that guarantees a total expected regret against any comparator policy π^* that is of order

$$\sqrt{n} \cdot C^\dagger(\pi^*) = \sqrt{n} \cdot \sum_{a \in \mathcal{A}} \frac{\pi^*(a)}{\mu(a)},$$

where μ is the behavior policy and $\pi(a)$ denotes the probability that policy π plays action a. Our method makes use of a slight *pessimistic* adjustment to the classic importance-weighted reward estimators commonly used in the adversarial bandit literature. We refer to the problem-dependent factor appearing in the bound as the *coverage ratio* and denote it by $C^\dagger(\pi^*)$. The coverage ratio quantifies the overlap between the comparator and behavior policies: it is of order $|\mathcal{A}|$ when the two policies closely match each other, but it blows up quickly as the two policies start to differ. Notably, our bounds can be orders of magnitude better than what one would obtain by adapting a standard adversarial bandit method without adjustments. For instance, a naïve analysis of the classic Exp3 method only gives a regret bound of order $\sqrt{n / \min_a \mu(a)}$ against all comparator

policies—even against ones that are actually well covered by the behavior policy. Besides providing theoretical results, we also confirm empirically that the performance of these two methods can be quite different, and in particular that Exp3 can indeed fail to take advantage of the comparator policy being well covered by the behavior policy.

Moreover, our contributions naturally fit in the broader context of online learning under partial monitoring, which generally considers situations where the observations made by the learner are decoupled from its rewards (Rustichini 1999; Bartók et al. 2014; Lattimore and Szepesvári 2019). In a general partial monitoring scenario, the learner receives an observation that depends on its action but may be insufficient to reconstruct the obtained reward. A well-studied special case of partial monitoring problems is online learning with feedback graphs (Mannor and Shamir 2011; Kocák, Neu, Valko, and Munos 2014; Alon, Cesa-Bianchi, Dekel, et al. 2015; Kocák, Neu, and Valko 2016; Alon, Cesa-Bianchi, Gentile, et al. 2017). In this setting, the set of observations associated with each action are given by a directed graph whose nodes are the actions: if actions a and a' are connected with an arc pointing from a to a', the learner observes the reward of action a when it plays action a'. The graph may not have self-loops for every action, which allows the possibility that the learner will not observe its own reward. Clearly, our setting can be embedded in this class of problems by considering a sequence of randomly generated star graphs where the action taken by the behavior policy is connected with all other actions. However, the graph does not contain self-loops which renders all existing methods for this problem unsuitable for our problem. In this sense, our contribution sheds some new light on the hardness of learning with feedback graphs without self-loops, and can potentially inspire future work in this domain.

Another line of work closely related to ours is the literature on offline reinforcement learning, where the learner cannot interact with the environment and has instead only access to a fixed dataset gathered by a behavior policy (Levine et al. 2020). In this context, the idea of of employing some form of pessimism has been extremely popular in the last few years, and pessimism has been purported to come with many desirable properties (Buckman, Gelada, and Bellemare 2021; Y. Jin, Z. Yang,

and Z. Wang 2021; Rashidinejad, B. Zhu, et al. 2021; Uehara and Sun 2021; Xie, C. Cheng, et al. 2021). One of these is that pessimistic offline RL methods can overcome the typical limitation of requiring the behavior policy to sufficiently explore the *whole* state-action space, which many previous results suffer from (Antos, Szepesvári, and Munos 2008; Munos and Szepesvári 2008; J. Chen and Jiang 2019; Xie and Jiang 2021). This assumption is very strong and often not verified in practice. However, a series of recent works show that, via an appropriate use of pessimism, it is possible to obtain bounds which scale with the coverage with respect to a comparator policy, instead of the whole state-action space. Many of these results are surveyed in the work of Xiao et al. (2021), who show that pessimistic policies are minimax optimal with respect to a special objective that weighs problem instances with a notion of inherent difficulty of estimating the value of the optimal policy. On the other hand, they show that without such weighting, pessimism is in fact only one of many possible heuristics that are all minimax optimal when considering the natural version of the optimization objective. This highlights that pessimism may not necessarily play a special role in offline optimization, and that the quest to understand the complexity of offline reinforcement learning is far from being over.

While our results definitely do not settle the debate of whether or not pessimism is the best way to deal with off-policy observations, they do provide some new insights. Most importantly, our findings highlight that pessimism remains an effective method for obtaining comparator-dependent guarantees. Such guarantees have attracted quite some interest in the literature on online learning with fully observable outcomes Chaudhuri, Freund, and Hsu 2009; Koolen 2013; Koolen and Erven 2015; Luo and Schapire 2015; Orabona and Pál 2016; Cutkosky and Orabona 2018. One common building block of parameter-free methods in this context is the Prod algorithm of Cesa-Bianchi, Mansour, and Stoltz (2007), used, for instance, in the algorithm designs of Gaillard, Stoltz, and Erven (2014); Sani, Neu, and Lazaric (2014); Koolen and Erven (2015). Interestingly, our analysis also leans heavily on the tools developed by Cesa-Bianchi, Mansour, and Stoltz (2007). When it comes to the bandit setting, comparator-dependent results are apparently much more sparse and in fact we are only aware of the work of Lattimore 2015 that studies

the possibility of guaranteeing better performance against certain comparators. As for our specific problem, we are not aware of any existing method that would be able to guarantee meaningful instance-dependent performance bounds.

5.1 Preliminaries

We study the off-policy adversarial bandit game introduced in Definition 4.1.1, assuming that the rewards are bounded as $\mathcal{R} = [0, 1]$.

Notably, the learner does not get to observe its own reward R_t but has to make do with the reward R_t^μ gained by the behavior policy. We allow the adversary to be adaptive in the sense of being able to take into account all past actions of the learner and the behavior policy when selecting the reward function. Also, the learner is allowed to use randomization for selecting its action.

Precisely, in this setting the total information revelead up to the end of round t is given by

$$\mathcal{F}_t = \sigma(A_1, A_1^\mu, g_1, \dots, A_t, A_t^\mu, g_t).$$

Notice that $\mathcal{F}_t$ denotes the full-information, while O_t denotes the information available to the agent. These two differ because the agent has no access to g_t, but only to R_t^μ.

Moreover, we remind that $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \,|\, \mathcal{F}_{t-1}]$ and $\mathbb{P}_t[\cdot] = \mathbb{P}(\cdot \,|\, \mathcal{F}_{t-1})$.

The objective of the learner is to minimize the *expected regret*, with respect to any time-invariant comparator policy $\pi^* \in \Delta_{\mathcal{A}}$, defined as

$$\mathfrak{R}(\pi^*) = \mathbb{E}\left[\sum_{t=1}^{n} \sum_{a} (\pi^*(a) - \pi_t(a)) g_t(a)\right]. \tag{5.1}$$

The expected regret measures the expected gap between the total rewards gained by the learner and the amount gained by a fixed comparator policy π^*.

The most common definition of regret compares the learner's performance to the optimal policy π^* that selects the action $a^* = \operatorname{argmax}_a \sum_{t=1}^{n} \bar{g}_t(a)$.

However, it is easy to see that this comparator strategy may be unsuitable for measuring performance in the setting we consider. Specifically, it is unreasonable to expect strong guarantees against the optimal policy when the behavior policy selects the optimal actions very rarely. Specifically, the adversary can take advantage of the behavior policy covering the action space only partially, and hide the best rewards among the least-frequently sampled actions. In the most extreme case, the behavior policy may not select some actions at all, which clearly makes it impossible for the learner to compete with the optimal policy. Thus, we aim to achieve regret guarantees that scale with the level of mismatch between the behavior and comparator policies, capturing the intuition that comparator strategies that are well covered by the data should be easier to compete with. Concretely, we aim to provide regret bounds that scale with $C^\dagger$ (Equation (4.10)). The intuitive significance of this coverage ratio is that it roughly captures the hardness of estimating the value of the comparator policy π^* using only data from μ. Indeed, a simple argument reveals that the estimation error of the total reward of any given action a scales as $\sqrt{n/\mu(a)}$ in the worst case. Thus, we set out to prove regret guarantees against each comparator π^* that scale proportionally to the worst-case estimation error of order $\sqrt{C^\dagger(\pi^*)n}$.

This section presents our main contributions: a set of algorithms for online off-policy learning and their comparator-dependent performance guarantees that scale with the coverage ratio between the comparator policy and the behavior policy. For the sake of clarity of exposition, we first describe our approach in a relatively simple setting where the number of actions is finite and the behavior policy is known. We then extend the algorithm to be able to deal with unknown behavior policies in Section 5.3 and to linear contextual bandit problems in Section 5.4.

5.2 Known Behavior Policy

Let us first consider the case where the learner has full prior knowledge of μ. The algorithm we propose is an adaptation of the Exp3-IX algorithm first proposed by Kocák, Neu, Valko, and Munos (2014) and later analyzed more generally by Neu (2015). At each time-step t the algorithm computes

the weights

$$w_1(a) = 1,$$
$$w_t(a) = w_{t-1}(a) \exp\left(\eta\, \hat{r}_{t-1}(a)\right) = \exp\left(\eta \sum_{k=1}^{t-1} \hat{r}_k(a)\right),$$

and the normalization factors $W_t = \sum_{a \in \mathcal{A}} w_t(a)$, and uses them to draw the action A_t according to

$$\pi_t(a) = \frac{w_t(a)}{W_t}.$$

Here, η is a positive learning-rate parameter and $\hat{r}$ is the *Implicit eXploration* (IX) estimate of the reward function g_t, modified to use the rewards obtained by the behavior policy μ, since the learner cannot see its own rewards:

$$\hat{r}_t(a) = \frac{R_t^\mu \mathbb{1}\{A_t^\mu = a\}}{\mu(a) + \gamma_t} = \frac{g_t(a) \mathbb{1}\{A_t^\mu = a\}}{\mu(a) + \gamma_t}, \tag{5.2}$$

where $\gamma_t \geq 0$ is an appropriately chosen parameter. The full algorithm is shown as Algorithm 1 .

Input: learning rate η, IX parameters $(\gamma_t)_{t=1}^n$
for $t \leftarrow 1, \ldots, n$ **do**
  compute $w_t(a) = \exp(\eta \sum_{k=1}^{t-1} \hat{r}_k(a)) \quad \forall a \in \mathcal{A}$
  play A_t according to $\pi_t(\cdot) = w_t(\cdot) / \sum_{a \in \mathcal{A}} w_t(a)$
  observe A_t^μ and R_t^μ
  compute $\hat{r}_t(A_t^\mu) = R_t^\mu / \left(\mu(A_t^\mu) + \gamma_t\right)$
end

Algorithm 1: Exp3-IX for Off-Policy Learning

When setting $\gamma_t = 0$, $\hat{r}_t$ is clearly an unbiased estimator of g_t since $\mathbb{E}_t[\mathbb{1}\{A_t^\mu = a\}] = \mu(a)$. Otherwise, for $\gamma_t > 0$, the estimator is biased towards zero which can be seen as a *pessimistic* bias in the sense that it underestimates the true rewards:

$$\mathbb{E}_t[\hat{r}_t(a)] \leq g_t(a).$$

This property is crucially important to achieve our goal to obtain performance guarantees that scale with the mismatch between π^* and μ. We believe that this use of the IX estimator with positive rewards is novel. In previous work, the IX estimator has been used with losses, resulting in an optimistic bias, exploited, for example, by the high-probability analysis of Neu (2015). It is far from obvious that our alternative usage of the IX estimator would induce the right notion of pessimism needed for achieving coverage-dependent results in the off-policy setting. This is established in the following:

Theorem 5.2.1. *For any comparator policy π^*, the expected regret of Exp3-IX initialized with any positive learning rate η and $\gamma_t = \frac{\eta}{2}$, is bounded as*

$$\mathfrak{R}(\pi^*) \leq \frac{\log|\mathcal{A}|}{\eta} + \mathbb{E}\left[\frac{\eta}{2}\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\frac{g_t(a)\pi^*(a)}{\mu(a)+\frac{\eta}{2}}\right]. \tag{5.3}$$

Setting the learning rate to $\eta = \sqrt{\frac{\log|\mathcal{A}|}{n}}$ and to $\eta = \sqrt{\frac{\log|\mathcal{A}|}{C^\dagger(\pi^*)n}}$ respectively gives

$$\mathfrak{R}(\pi^*) \leq \sqrt{n\log|\mathcal{A}|}\left(1+\frac{1}{2}C^\dagger(\pi^*)\right) \tag{5.4}$$

$$\mathfrak{R}(\pi^*) \leq \sqrt{2C^\dagger(\pi^*)n\log|\mathcal{A}|}. \tag{5.5}$$

The proof is based on a set of small but important changes made to the standard Exp3 analysis originally due to Auer et al. (2002), and is deferred to Section 5.5. The bound above successfully achieves our goal of guaranteeing better regret against comparator policies that are well-covered by the behavior policy. In particular, the first bound of Equation (5.4) provides a bound that holds uniformly for all behavior policies without requiring prior commitment to any coverage level, whereas the second bound guarantees improved guarantees against policies with a given coverage level at the price of using a learning-rate parameter that is specific to the desired coverage. Notably, the coverage ratio is of the order $|\mathcal{A}|$ when the comparator policy closely matches the behavior policy, but the actual bound of Equation (5.3) can be much smaller when there are many

actions that the behavior policy selects with probability much smaller than γ.

It is worthwhile to compare this result with what one would obtain by a straightforward adaptation of a standard adversarial bandit algorithm like Exp3 (Auer et al. 2002)—which essentially corresponds to our algorithm with the choice $\gamma = 0$. A standard calculation shows that the regret of this strategy can be upper bounded by

$$\mathfrak{R}(\pi^*) \leq \frac{\log |\mathcal{A}|}{\eta} + \eta \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \mathbb{E}\left[\frac{\pi_t(a)}{\mu(a)}\right].$$

Notice that the right-hand side of this bound does not depend on the comparator policy, which suggests that this method is not quite suitable for achieving our goal. Even worse, the only way to bound the second term in the bound seems to be by $n/\min_a \mu(a)$, which scales inversely with the coverage of the least well-covered action. A pessimistic interpretation of this argument suggests that Exp3 may have huge regret when some actions are not covered appropriately. A more charitable reading is that Exp3 may not be able to take advantage of situations where the comparator policy is well-covered by the behavior policy. We set out to understand this phenomenon empirically in Section 5.6.

The results of Theorem 5.2.1 could be extended to deal with a nonstationary sequence of behavior policies, and the regret bound can be shown to scale with the average of the coverage ratios, as long as the behavior policies are revealed to the learner.

5.3 Unknown Behavior Policy

In the previous section we assumed to have full prior knowledge of the behavior policy μ in order to compute our reward estimator $\hat{r}_t$. In this section, we show that this is not an inherent limitation of our technique and that it can be easily addressed by using a simple plugin estimator $\hat{\mu}_t$

of the behavior policy, which is then used in the definition of $\hat{r}_t$:

$$\hat{\mu}_1(\cdot) = 0, \qquad \hat{\mu}_t(a) = \frac{1}{t-1}\sum_{k=1}^{t-1} \mathbb{1}\{A_k^\mu = a\}, \tag{5.6}$$

$$\hat{r}_t(a) = \frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\hat{\mu}_t(a) + \gamma_t}. \tag{5.7}$$

We then feed these reward estimates to the exponential-weights procedure described in the previous section. As the following theorem shows, the resulting algorithm satisfies essentially the same regret bound as the method that has full knowledge of μ.

Theorem 5.3.1. *For any comparator policy π^*, the expected regret of* Exp3-IX *with learning rate $\eta = \sqrt{\log(|\mathcal{A}|)/n}$ and parameter sequence $\gamma_1 = 1 + \frac{\eta}{2}$, $\gamma_t = \frac{\eta}{2} + \sqrt{\log(|\mathcal{A}|(t-1)^2)/(2t-2)}$, and estimates as in Equation (5.6), is bounded as*

$$\mathfrak{R}(\pi^*) = \mathcal{O}\left(C^\dagger(\pi^*)\sqrt{n\log(|\mathcal{A}|n)}\right). \tag{5.8}$$

The parameter tuning achieving the above bound is similar to what is used in the previous theorem, and does not require the learner to have any problem-specific information that would be difficult to acquire. Details are relegated to Appendix A.2 along with the proof of the theorem.

5.4 Linear Contextual Bandits

We now switch gears and provide an extension to a significantly more advanced setup: that of adversarial linear contextual bandits, first studied by Neu and Olkhovskaya (2020). This combines the online learning with off-policy feedback setting introduced in Section 4.1 and studied in the previous sections, with the linear rewards assumption introduced in Section 4.3.1.

In each round t of this sequential game, the learner first observes a context X_t before making its decision, and the reward function g_t is assumed to be an adversarially chosen function of the context X_t and the action A_t taken by the learner. In particular, the adversary chooses a *reward vector*

$\theta_t \in \mathbb{R}^d$ at each step, which determines the rewards for each context-action pair as

$$g_t(x, a) = \langle \theta_t, \varphi(x, a) \rangle,$$

where $\varphi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is a *feature map* known to both the learner and the adversary. We assume that the contexts live in an abstract space $\mathcal{X}$ and are drawn i.i.d. according to a fixed probability distribution for all t. On the other hand, the adversary has full freedom in choosing the reward functions, as long as it only depends on past observations and in particular does not depend on X_t or A_t. The only restriction we put on the adversary is that we continue to require the rewards to be in the interval $[0, 1]$. Moreover, as in the previous section the learner is not allowed to see its own rewards, but only the ones of an other policy μ running in parallel. In this setting, a policy π is a mapping from contexts to probability distributions over the space of actions.

The objective of the learner is to minimize the regret defined with respect to any time-invariant comparator policy $\pi^* : \mathcal{X} \to \Delta_{\mathcal{A}}$ as:

$$\mathfrak{R}(\pi^*) = \mathbb{E}\left[\sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \left(\pi^*(a \mid X_t) - \pi_t(a \mid X_t)\right) g_t(X_t, a)\right].$$

Our algorithm for this setting is a combination of the context-wise exponential weights method proposed by Neu and Olkhovskaya (2020) with the ideas developed in the previous section. The algorithm design is complicated by the fact that the implicit exploration estimator is not very straightforward to extend to this setting, which necessitates an alternative, but closely related, approach. In particular, we will define an *unbiased* estimator of the reward vector θ_t and feed the resulting reward estimates to calculate policy updates via the Prod update rule proposed by Cesa-Bianchi, Mansour, and Stoltz (2007) (see also Cesa-Bianchi and Lugosi (2006), Section 2.7).

Concretely, following the algorithm design of Neu and Olkhovskaya (2020), we define the estimator

$$\hat{\theta}_t = \mathbf{\Lambda}_\mu^{-1} \varphi(X_t, A_t^\mu) \cdot R_t^\mu, \tag{5.9}$$

where $\mathbf{\Lambda}_\mu^{-1}$ is defined according to Equation (4.12). Since $\varphi(X_t, A_t^\mu)R_t^\mu = \varphi(X_t, A_t^\mu)\varphi(X_t, A_t^\mu)^\intercal\theta_t$, it is easy to see that $\hat{\theta}_t$ is an unbiased estimator of θ_t. These estimators are then used to update a set of weights w_t defined for each context-action pair as

$$w_t(x,a) = \prod_{k=1}^{t-1}(1 + \eta\langle\hat{\theta}_k, \varphi(x,a)\rangle), \tag{5.10}$$

$$W_t(x) = \sum_a w_t(x,a), \tag{5.11}$$

and the policy is then given as $\pi_t(a|x) = \frac{w_t(x,a)}{W_t(x)}$. Notice that this policy can be also seen as another form of pessimistic reward estimation. In-fact, letting $\hat{r}_t$ be an unbiased reward estimator, the PROD-style update of Equation (5.10) can be seen as an EXP3 update with the modified reward estimator $\tilde{r}_t = \frac{1}{\eta}\log(1+\eta\hat{r}_t)$, which clearly lower bounds the original reward estimator throught the inequality $\log(1+z) \leq z$, corresponding to a form of pessimism. Moreover, the policy can be easily implemented without explicitly keeping track of the weights for all (x,a) pairs, as they are well-defined through the sequence of reward-estimate vectors $\hat{\theta}_1, \dots, \hat{\theta}_{t-1}$. For simplicity[1], we assume that the matrix $\mathbf{\Lambda}_\mu$ is known for the learner and has uniformly lower-bounded eigenvalues so that its inverse exists. Following the naming convention of Neu and Olkhovskaya (2020), we call the resulting algorithm LINPROD and show its pseudocode in Algorithm 2.

Similarly to the previous sections, we are aiming for a comparator-dependent performance guarantee that depends on the mismatch of the comparator and the behavior policy. However, this quantity is not straightforward to define in the case that we consider, due to the fact that we consider a potentially infinite space of contexts. In particular, the natural idea of considering

$$\mathbb{E}\left[\sum_a \frac{\pi^*(a|X_t)}{\mu(a|X_t)}\right]$$

[1]These restrictions can be removed using the techniques developed in the previous section, although at the price of a significantly more technical analysis. We opted to preserve clarity of presentation instead.

Input: learning rate η
for $t \leftarrow 1, \dots, n$ **do**
  observe X_t
  compute $w_t(X_t, \cdot) = \prod_{k=1}^{t-1}(1 + \eta\langle\hat{\theta}_k, \varphi(X_t, \cdot)\rangle)$
  draw A_t from $\pi_t(\cdot \mid X_t) = w_t(X_t, \cdot)/\sum_a w_t(X_t, a)$
  observe R_t^μ and $\varphi(X_t, A_t^\mu)$
  compute $\hat{\theta}_t$ as in Equation (5.9)
end

Algorithm 2: LinProd for off-policy learning

as a measure of mismatch is problematic as it can blow up when there exists even a tiny set of contexts where the two policies have no overlap.

Intuitively, it should be possible to estimate the reward vector even when there are states where the policies pick different actions, as long as they are aligned in the feature space in an appropriate sense. To make this intuition formal, we will consider the following alternative notion of *feature coverage ratio*:

$$C^\dagger(\pi^*) = \mathrm{Tr}\big(\mathbf{\Lambda}_\mu^{-1}\,\mathbf{\Lambda}_{\pi^*}\big). \tag{5.12}$$

This notion of coverage appropriately measures the extent to which the feature vectors $\varphi(X_t, A_t^*)$ generated by the comparator policy line up with the features excited by the behavior policy. Similar distribution-mismatch measures are common in the offline RL literature, and in particular the results of Y. Jin, Z. Yang, and Z. Wang (2021) are stated in terms of the same quantity. The following theorem gives a performance guarantee stated in terms of this measure of distribution mismatch.

Theorem 5.4.1. *Let η be any positive learning rate and suppose that it is small enough so that $\lambda_{\min}(\mathbf{\Lambda}_\mu) \geq 2\eta \sup_{x,a} \|\phi(x,a)\|_2^2$ holds. Then, for any comparator policy π^* the expected regret of LinProd is upper-bounded by*

$$\mathfrak{R}(\pi^*) \leq \frac{\log|\mathcal{A}|}{\eta} + \eta n C^\dagger(\pi^*),$$

Setting $\eta = \sqrt{\frac{\log|\mathcal{A}|}{n}}$ and $\eta = \sqrt{\frac{\log|\mathcal{A}|}{C^\dagger(\pi^*)n}}$ and supposing that n is large

enough so that η satisfies the condition, the regret can be further bounded respectively as

$$\mathfrak{R}(\pi^*) \leq \sqrt{n \log |\mathcal{A}|}\,(1 + C^\dagger(\pi^*))\,,$$
$$\mathfrak{R}(\pi^*) \leq 2\sqrt{C^\dagger(\pi^*) n \log |\mathcal{A}|}.$$

The bound mirrors the qualities of Theorem 5.2.1, and in particular it implies good performance when the comparator policy is well-covered by the behavior policy. Under ideal conditions where these policies are close enough, the coverage ratio is of order d, which essentially matches the rate proved by Neu and Olkhovskaya (2020) for the case of standard bandit feedback. The bound then degrades as the two policies drift apart. We recover the best-known bounds for the stochastic setting (Y. Jin, Z. Yang, and Z. Wang 2021). The latter were stated for the setting of off-policy learning in linear MDPs, which includes the stochastic version of our problem as a special case. Note that our algorithm requires knowledge of $\mathbf{\Lambda}_\mu$. However, provided that the context distribution is known, it is possible to use instead an estimate based on matrix geometric resampling, as proposed by Neu and Olkhovskaya 2020.

5.5 Analysis

This section provides the key ideas required for proving our main results. Due to space restrictions, we will only prove Theorem 5.2.1 here and defer the proof of the other two theorems to Appendices A.2 and A.3.

For the analysis, it will be useful to define the unbiased reward estimator

$$\widehat{r}_t^{\mathrm{IW}}(a) = \frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\mu(a)},$$

which essentially corresponds to the biased IX estimator $\widehat{r}_t$ when setting $\gamma_t = 0$. One of the most important properties of the IX estimator that we will repeatedly use is stated in the following inequality:

$$\frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\mu(a) + \gamma_t} \leq \frac{1}{2\gamma_t} \log\left(1 + 2\gamma_t \widehat{r}_t^{\mathrm{IW}}(a)\right). \tag{5.13}$$

The result follows from a simple calculation in the proof of Lemma 1 of Neu (2015) that we reproduce here for the convenience of the reader.

Let $c \in \mathbb{R}_+$ be any non-negative constant. Then,

$$\frac{g_t(a)\mathbb{1}\{A_t = a\}}{\mu(a) + c} \leq \frac{g_t(a)\mathbb{1}\{A_t = a\}}{\mu(a) + c\, g_t(a)} = \frac{\mathbb{1}\{A_t = a\}}{2c} \cdot \frac{2c\, g_t(a)/\mu(a)}{1 + c\, g_t(a)/\mu(a)}$$
$$\leq \frac{1}{2c} \log(1 + 2c\, \widehat{r}_t^{\mathrm{IW}}(a))$$

where the first step follows from $g_t(a) \in [0, 1]$ and the last one from the inequality $\frac{x}{1+x/2} \leq \log(1 + x)$, which holds for all $x \geq 0$.

Notably, the term on the right hand side can be thought of as a reward estimator itself. Combining this reward estimator with the exponential weights policy with $\eta = \gamma$ gives rise to the Prod algorithm of Cesa-Bianchi, Mansour, and Stoltz (2007), which is a fact that some of our proofs will implicitly take advantage of. This observation also motivates our algorithm design for the contextual bandit setting in Section 5.4.

The proof of Theorem 5.2.1 The proof builds on the classical analysis of exponential weights algorithm originally due to Vovk (1990), Littlestone and Warmuth (1994) and Freund and Schapire (1997), and its extension to adversarial bandit problems by Auer et al. (2002). In particular, our starting point is the following lemma that can be proved directly with arguments borrowed from any of these past works:

Lemma 5.5.1.

$$\sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \pi^*(a) \widehat{r}_t(a) \leq \frac{\log |\mathcal{A}|}{\eta}$$
$$+ \frac{1}{\eta} \sum_{t=1}^{n} \log \sum_{a \in \mathcal{A}} \pi_t(a) \exp\left(\eta \widehat{r}_t(a)\right).$$

We include the proof for the sake of completeness in Appendix A.1. To proceed, notice that the above bound can be combined with Equation (5.13)

to obtain

$$\begin{aligned}
\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)\hat{r}_t(a) &\leq \frac{\log|\mathcal{A}|}{\eta} \\
&\quad + \frac{1}{\eta}\sum_{t=1}^{n}\log\sum_{a\in\mathcal{A}}\pi_t(a)\exp\left(\frac{\eta}{2\gamma}\log\left(1+2\gamma\,\hat{r}_t^{\mathrm{IW}}(a)\right)\right) \\
&= \frac{\log|\mathcal{A}|}{\eta} + \frac{1}{\eta}\sum_{t=1}^{n}\log\sum_{a\in\mathcal{A}}\pi_t(a)\left(1+\eta\hat{r}_t^{\mathrm{IW}}(a)\right) \\
&\leq \frac{\log|\mathcal{A}|}{\eta} + \sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi_t(a)\hat{r}_t^{\mathrm{IW}}(a), \qquad (5.14)
\end{aligned}$$

where we used the choice $\gamma = \eta/2$ in the second line and the inequality $\log(1+x) \leq x$ that holds for all $x > -1$ in the last line.

It remains to relate the two sums in the above expression to the total reward of the learner and the comparator policy. To this end, we first notice that for any given action a, we have

$$\begin{aligned}
\mathbb{E}_t[\hat{r}_t(a)] &= \mathbb{E}_t\left[\frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\mu(a)+\gamma}\right] \\
&= \frac{g_t(a)\mu(a)}{\mu(a)+\gamma} = g_t(a) - \frac{\gamma\cdot g_t(a)}{\mu(a)+\gamma}.
\end{aligned} \qquad (5.15)$$

Via the tower rule of expectation, this implies

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)\hat{r}_t(a)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)g_t(a) - \gamma\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\frac{\pi^*(a)g_t(a)}{\mu(a)+\gamma}\right].
\end{aligned}$$

Similarly, since $\mathbb{E}_t[\hat{r}_t^{\mathrm{IW}}(a)] = g_t(a)$, we also have

$$\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi_t(a)\hat{r}_t^{\mathrm{IW}}(a)\right] = \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi_t(a)g_t(a)\right].$$

Putting these two facts together with Equation (5.14), we obtain the result claimed in the theorem.

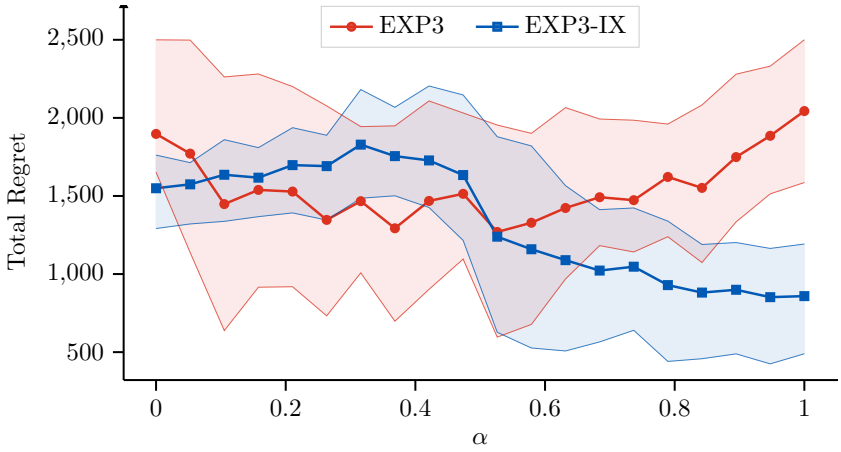

Figure 5.1: Total regret after 1000 steps for different values of the interpolation parameter α. Thick lines represent the mean regret over 100 independent runs, while the shaded area represents the interval between the 25% and 75% quantiles.

5.6 Empirical Results

The goal of this section is to compare the performances of Exp3 and Exp3-IX under different levels of coverage, and verify if indeed our method outperforms Exp3 in situations where the behavior policy is well-aligned with the comparator, as the theory suggests. As pointed out earlier, a naïve analysis of Exp3 suggests that its regret may scale as $\sqrt{n/\min_a \mu(a)}$ in the worst case, and therefore may not be able to take advantage of situations where the behavior policy is well aligned with the comparator. Our proposed method, instead, should be able to do so since it enjoys comparator-dependent bounds.

We instantiate a 100-armed bandit, with Bernoulli rewards for all arms. By default, all rewards have mean 0.5. However, for the first half of the game ($t \leq n/2$), we change the mean reward of the last arm to 0.8, and for the remaining half, the mean of the first arm to 1. Suboptimal arms always have the default mean reward of 0.5. This means that arm 100 is the best for the first half of the game, but eventually gets outperformed

by arm 1. We set the number of rounds n to 10000, the learning rate η of both algorithms to the recommended $\sqrt{\log(|\mathcal{A}|)/n}$ and $\gamma_t = \eta/2$. We repeat the game for a range of behavior policies defined for each α as $\pi_{B,\alpha}(i) \propto (1-\alpha)\frac{i}{|\mathcal{A}|} + \alpha(1 - \frac{i-1}{|\mathcal{A}|})$, for $i \in [0, \dots, |\mathcal{A}|]$, where α varies from 0 to 1. Hence, α closer to 1 means the behavior policy puts large probability mass on the first action, which we use as the comparator in our experiment. We plot the results of the experiment on Section 5.6.

The results clearly match the intuitions that one can derive from our performance guarantees: the regret of Exp3 indeed deteriorates as $\min_a \mu(a)$ approaches 0 at the two extremes $\alpha = 0$ and $\alpha = 1$. In particular, Exp3 fails to take advantage of the favorable case where the optimal policy is well covered, while Exp3-IX performs significantly better in the latter case, as predicted by our theory.

Moreover, it is worth to note that Exp3-IX was originally proposed, in the adversarial bandit literature, as a variant of Exp3 with lower variance, allowing to bound regret with high probability instead of merely in expectation. This variance reduction effect clearly carries over to our setting. However, we were not able to establish high-probability bounds for the adversarial-off-policy setting so far, and leave this question open for future research.

Chapter 6

Importance-Weighted Offline Learning

Offline Policy Optimization (OPO) is the problem of learning a near-optimal policy based on a dataset of historical observations. This problem is of outstanding importance in real-world applications where experimenting directly with the environment is costly, but otherwise large volumes of offline data is available to learn from. Such settings include problems in healthcare (Murphy 2003; Kim et al. 2011; Bertsimas et al. 2017; Rehg, Murphy, and Kumar 2017), advertising (Bottou et al. 2013; Farias and A. A. Li 2019), or recommender systems (L. Li, Chu, et al. 2011; Schnabel et al. 2016).

A popular approach for this setting is *importance-weighted offline learning*, where one optimizes an unbiased estimate of the expected reward, obtained through an appropriately reweighted average of the rewards in the dataset (L. Li, Chu, et al. 2011; Bottou et al. 2013). To deal with unstable nature of these estimators, the influential work of Swaminathan and Joachims (2015) proposed an approach called "counterfactual risk minimization", which consists of adding a regularization term to the optimization problem to down the fluctuations, thus preventing the optimizer to overfit to random noise. Their work has inspired a number of follow-ups that either refined the regularization terms to yield better the-

oretical guarantees (Y. Jin, Ren, et al. 2022; L. Wang, Krishnamurthy, and Slivkins 2023), or developed practical methods with improved empirical performance in large-scale problems London and Sandler (2019); Sakhi, Alquier, and Chopin (2023). In this paper, we contribute to this line of work by studying a simple and robust variant of the standard importance-weighted reward estimators used in past work, and showing tight theoretical performance guarantees for it.

Our main contribution is showing that the so-called *implicit exploration* (IX) estimator (originally proposed by Kocák, Neu, Valko, and Munos 2014 and Neu 2015 in the context of online learning) achieves a massive variance-reducing effect in our offline learning setting, and using this observation to derive performance guarantees that are both significantly tighter and easier to interpret than all previous results in the literature. In particular, we formally show that the regularization effect built into the IX estimator is strong enough so that no further regularizer is required to stabilize the performance of policy optimization. This result is perhaps surprising for the reader familiar with past work on the subject, especially since several of these works made use of IX-like variance reduced estimators without managing to drop the additional regularization. The key observation that allows us to prove our main results is that the tails of importance-weighted estimators are *asymmetric*, which allows us to tightly control the two tails separately via specialized concentration inequalities. This is to be contrasted with previous results that all rely on symmetric confidence intervals that turn out to be needlessly conservative. This new perspective not only allows us to obtain better results but also to simplify the analysis: both of the concentration inequalities we use for the two tails can be derived using elementary techniques in a matter of a few lines[1].

More concretely, our main result is a regret bound that scales with the degree of "overlap" between the comparator policy and the behavior policy, demonstrating better scaling against policies that are covered better by the observed data. Unlike virtually all previous work, our guarantees

[1]In fact, both results are readily available in the literature: one is the main result of Neu (2015) regarding the upper tail of the IX estimator, and another is stated as an exercise in Boucheron, Lugosi, and Massart (2013).

do not require the unrealistic condition that action-sampling probabilities be bounded away from zero for all contexts. Our algorithm can be implemented efficiently using a single call to a cost-sensitive classification oracle, thus effectively reducing the offline policy optimization problem to a standard supervised learning task (which feature is in high regard thanks to the influential works of Langford and T. Zhang 2007; Dudík et al. 2011; Agarwal et al. 2014 in the broader area of contextual bandit learning). For simplicity of exposition, we prove our main result for finite policy classes and show that the regret scales logarithmically with the size of the class. We also provide some extensions to the simple algorithm achieving these results, namely a version that trades oracle-efficiency for a better scaling with the quantity measuring the mismatch between the target and behavior policies, and a "PAC-Bayesian" variant that can make use of prior information on the problem and also works for infinite policy classes. This extends the recent works of London and Sandler (2019); Flynn et al. (2023); Sakhi, Alquier, and Chopin (2023) by providing better generalization bounds and introducing a new family of PAC-Bayesian regret bounds that apparently have not existed so far in the literature. We also illustrate our theoretical findings with a set of experiments conducted on real data, and empirically verify the robustness of our method as compared to some natural baselines.

It is worth mentioning a parallel line of work on contextual bandits that starts from the assumption that the reward function belongs to a known function class, and thus a near-optimal policy can be learned by identifying the true reward function within the class up to sufficient accuracy. This perspective has been adopted by Y. Jin, Z. Yang, and Z. Wang (2021) (as well as a sequence of follow-up works on offline reinforcement learning) who considered function classes that are linear in some low-dimensional features of the context-action pairs. These works provide simple algorithms with strong theoretical performance guarantees, but they are all limited by the strong assumptions that need to be made about the reward function (and it is unclear how sensitive they are to model misspecification). In contrast, the setting we consider assumes access to a policy class and allows the development of algorithms that perform nearly as well as the best policy within the class *without* requiring that the rewards have a simple parametric form. This setting comes with its own set of trade-

offs: the statistical complexity of learning in this setting depends on the complexity of the policy class, and hard problems will evidently require large classes of policies to accommodate best-in-class policies with satisfying performance. Our results in this paper highlight some further open questions in this setting regarding computational-statistical trade-offs—the discussion of which we relegate to Section 8.2.

6.1 Preliminaries

We study the problem of offline learning in stochastic contextual bandits detailed in section Section 4.2.

For simplicity, we suppose that the behavior policy μ is fixed and known, and only note here that extension to adaptive behavior policies is straightforward.

The goal is to use the available data to produce a policy $\tilde{\pi}_n$ achieving the highest possible expected reward. The performance will be measured in terms of *regret* (or *excess risk*) with respect to a comparator policy π^*.

We assume to have access to a policy class $\Pi \subseteq \{\pi : \mathcal{X} \to \Delta_{\mathcal{A}}\}$ and aim to provide regret bounds against all policies within the class. For most of our contributions, we will work with finite policy classes and assume access to a computational oracle that can return optimal policies given an appropriately defined input dataset. This is the setting described in Section 4.3.2, and the oracle we employ is the one of Definition 4.3.2.

We are interested in developing algorithms that access the oracle a small constant number of times while providing formal performance guarantees on the quality of the output policy.

6.2 Pessimistic importance-weighted offline learning in contextual bandits

The generic recipe for offline learning with pessism, detailed in Section 4.5, has been combined with the IW estimator, defined in Equation (4.6),

by Swaminathan and Joachims (2015), Y. Jin, Ren, et al. (2022) and L. Wang, Krishnamurthy, and Slivkins (2023). This "pessimistic importance-weighted offline learning" approach, which we abbreviate as *PIWO learning*, has several downsides, depending on the choice of $B(\pi)$. First, as pointed out recently by (L. Wang, Krishnamurthy, and Slivkins 2023), $\widehat{v}(\pi) - B(\pi)$ may not be necessarily be of the form required by a practical optimization oracle. Even more concerningly, a conservatively chosen adjustment $B(\pi)$ may not only result in loose theoretical guarantees, but also poor empirical performance. Indeed, notice that setting $B(\pi)$ too large may overwhelm the data-dependent value estimates, thus resulting in a policy that effectively ignores the observed data from policies that are relatively poorly covered. In extreme cases, this approach may even favor policies that have never been observed to yield any reward whatsoever over policies with positive estimated reward but high estimated uncertainty.

The cleanest results for this PIWO learning approach have been derived by L. Wang, Krishnamurthy, and Slivkins (2023), who used the adjustment

$$B(\pi) = \beta \sum_{t=1}^{n} \sum_{a} \frac{\pi(a|X_t)}{\mu(a|X_t)}.$$

Their regret bounds are stated in terms of the coverage ratio $C^\diamond$, defined in Equation (4.9). Assuming that the worst-case density ratio (Equation (4.11)) between the two policies is uniformly upper-bounded by α, L. Wang, Krishnamurthy, and Slivkins (2023) obtain, for their oracle-efficient algorithm, a regret bound of the form

$$\mathfrak{R}_n(\pi^*) = \mathcal{O}\left(C^\diamond(\pi^*) \sqrt{\frac{\log\left(|\Pi|/\delta\right)}{n}} + \frac{\alpha \log(|\Pi|/\delta)}{n} \right). \tag{6.1}$$

This bound has the appealing property that its leading term scales as $C^\diamond(\pi^*)/\sqrt{n}$, thus guaranteeing good performance when the comparator policy π^* is well-covered by the behavior policy. The bound can be improved to scale with $\sqrt{C^\diamond(\pi^*)}$ instead of $C^\diamond(\pi^*)$ if one has prior knowledge of the coverage ratio against the target policy π^*. On the negative side, the result effectively requires the strong *uniform coverage* condition which ensures that all actions are sampled at least a constant α fraction of times

in the data set. This condition is typically not met in realistic applications for reasonable values of α, and in particular the bound becomes completely void of meaning if there exists one single context x where some action a is selected with zero probability.

The original algorithm by Swaminathan and Joachims 2015 suffered from the same issue. Recently, Y. Jin, Ren, et al. 2022 were able to relax this uniform-coverage condition by developing a sophisticated concentration inequality that only requires the third moment of the importance weights $\sum_a \frac{\pi^*(a|X_t)}{\mu(a|X_t)}$ to be bounded. Eventually, their bounds only apply to deterministic policies that map each context x to a single action $\pi^*(x)$, and depend on the quantity $\alpha^* = \inf_x \mu(\pi^*(x)|x)$. Their most clearly stated result is Corollary 4.3, where they effectively show

$$\mathfrak{R}_n(\pi^*) = \mathcal{O}\left(\sqrt{\frac{\log\left(|\Pi|T\right)}{\alpha^* n}} \cdot \left(\log\left(\frac{1}{\delta}\right)\right)^{3/2}\right).$$

This bound still remains vacuous if there is one single context where $\mu(\pi^*(x)|x)$ is zero. A further downside of their method pointed out by L. Wang, Krishnamurthy, and Slivkins (2023) is that the proposed algorithm is not directly implementable with a CSC oracle due to the form of the adjustment B_n they use. In the following section, we will develop an algorithm that eliminates all these limitations.

6.3 Pessimism and Variance Reduction via Implicit Exploration

Our main contribution is addressing the limitations of the PIWO learning framework in the previous section by studying a very simple adjustment to the standard IW estimator. Concretely, we adapt the so-called "Implicit eXploration" (IX) estimator of Neu 2015 defined as

$$\hat{v}_n(\pi) = \frac{1}{n}\sum_{t=1}^{n} \frac{\pi(A_t|X_t)}{\mu(A_t|X_t) + \gamma} \cdot R_t, \tag{6.2}$$

where $\gamma \geq 0$ is a hyperparameter of the estimator that we will sometimes refer to as the "IX parameter". This adjustment implicitly acts like mixing

the behavior policy with a uniform exploration policy, thus reducing the random fluctuations of the IW estimator (and justifying the name "implicit exploration"). The price of this stabilization effect is that the estimates are biased towards zero to an extent that can be controlled using the IX parameter γ. Indeed, as a simple calculation shows, the IX estimator satisfies

$$\mathbb{E}\left[\hat{v}_n(\pi)\right] = \rho(\pi) - \gamma C_\gamma(\pi),$$

with the bias term $C_\gamma(\pi)$ given as

$$C_\gamma(\pi) = \mathbb{E}\left[\sum_{a\in\mathcal{A}} \frac{\pi(a|X)}{\mu(a|X)+\gamma} \cdot g(X,a)\right]. \tag{6.3}$$

Since the rewards are assumed to be non-negative, this bias can be interpreted as a *pessimistic* adjustment to an otherwise unbiased estimator, and it is thus reasonable to expect it to have the same effect as the adjustments used in the general PIWO framework[2].

Note that $C_\gamma(\pi)$ is closely related to the policy coverage ratio $C^\diamond(\pi)$ as defined in Equation (4.9), up to the two differences that *i)* it replaces $\mu(X,a)$ by $\mu(X,a)+\gamma$ in the denominator and *(ii)* it is scaled with the rewards $g(X,a)$. Both of these adjustments make it strictly smaller than $C^\diamond(\pi)$ as long as $\gamma > 0$, and notably it always remains bounded as $C_\gamma(\pi) \leq \frac{1}{\gamma}$, no matter how small $\mu(a|x)$ gets. Furthermore, due to the scaling with the rewards, $C_\gamma(\pi)$ is small for policies with low expected reward, and in particular it equals zero for a policy with zero expected reward. In what follows, we will refer to C_γ as the *smoothed coverage ratio*.[3]

Our algorithm consists of simply selecting the policy that maximizes the IX value estimates:

$$\hat{\pi}_n = \arg\max_{\pi\in\Pi} \hat{v}_n(\pi).$$

[2]In fact, the pessimistic bias of the IX estimators has been recently pointed out and utilized by Gabbianelli, Neu, and Papini (2023) in the vaguely related context of online learning with off-policy feedback.

[3]We use this term in the sense of the Laplace smoothing of estimators, not to be confused with the smoothed analysis of algorithms (Spielman and Teng 2001) applied to contextual bandits by Krishnamurthy et al. 2019.

We refer to this algorithm as PIWO-IX, standing for "Pessimistic Importance-Weighted Offline learning with Implicit eXploration". Note that PIWO-IX can be implemented via a single call to the CSC oracle with the gain vectors defined as

$$y_t(a) = \mathbb{1}\{A_t = a\} R_t / (\mu(A_t|X_t) + \gamma).$$

The following theorem states our main result regarding PIWO-IX.

Theorem 6.3.1. *With probability at least* $1 - \delta$*, the regret of PIWO-IX against any comparator policy* $\pi^* \in \Pi$ *satisfies*

$$\mathfrak{R}_n(\pi^*) \leq \frac{\log\left(2|\Pi|/\delta\right)}{\gamma n} + 2\gamma C_\gamma(\pi^*).$$

Furthermore, by setting γ *to* $\sqrt{\frac{\log(2|\Pi|/\delta)}{n}}$*, the bound becomes*

$$\mathfrak{R}_n(\pi^*) \leq \left(2C_\gamma(\pi^*) + 1\right) \sqrt{\frac{\log\left(2|\Pi|/\delta\right)}{n}}.$$

The bound improves on the results of L. Wang, Krishnamurthy, and Slivkins (2023) stated as Equation (4.6) along several dimensions. Most importantly, our result removes the need for the behavior policy to be bounded away from zero, and as such completely does away with the uniform coverage assumptions needed by all previous work on the topic. Another improvement is that our bound tightens the dependence on the coverage ratio from $C^\diamond(\pi^*)$ to the potentially much smaller $C_\gamma(\pi^*)$. A small practical improvement is that PIWO-IX calls the CSC oracle with a sparse input vector which can be computed slightly more efficiently than the dense inputs used by L. Wang, Krishnamurthy, and Slivkins (2023). This sparsity also leads to the practical advantage that PIWO-IX does not output policies that have never been observed to yield nonzero rewards (as long as there are alternatives that do receive positive rewards). We provide further comments on the tightness of the bound above and other properties of PIWO-IX in Section 8.2.

The key idea behind the proof of Theorem 6.3.1 is noticing that the tails of the IX estimator are asymmetric: since $\hat{v}_n$ is a nonnegative random

variable, its only extreme values are all going to be positive. More formally, this means that its lower tail will always be lighter than its upper tail, and thus a tight analysis needs to handle the two tails using different tools. Below, we state two lemmas that separately characterize the lower and upper tails of the IX Equation (6.2). The first of these bounds the upper tail along the lines of Lemma 1 (and Corollary 1) of Neu 2015:

Lemma 6.3.2. *With probability at least* $1-\delta$*, the following holds simultaneously for all* $\pi \in \Pi$*:*

$$\hat{v}_n(\pi) - \rho(\pi) \leq \frac{\log\left(|\Pi|/\delta\right)}{2\gamma n}.$$

The proof is provided in Appendix B for completeness, but is otherwise lifted entirely from Neu 2015. The second lemma provides control of the lower tail of $\hat{v}_n$:

Lemma 6.3.3. *With probability at least* $1-\delta$*, the following holds simultaneously for all* $\pi \in \Pi$*:*

$$\rho(\pi) - \hat{v}_n(\pi) \leq \frac{\log(|\Pi|/\delta)}{2\gamma n} + 2\gamma C_\gamma(\pi).$$

The proof follows from the observation that, since the rewards are non-negative, $\hat{v}_n$ is a non-negative random variable, and as such its lower tail is well-controlled by its second moment (see, e.g., Exercise 2.9 in (Boucheron, Lugosi, and Massart 2013)). The full proof is included in Appendix B for completeness. With the above two lemmas, we can easily prove our main theorem.

Proof of Theorem 6.3.1 The statement follows from combining the two lemmas via a union bound, and exploiting the definition of the algorithm:

$$\begin{aligned}\rho(\hat{\pi}_n) &\geq \hat{v}_n(\hat{\pi}_n) - \frac{\log\left(2|\Pi|/\delta\right)}{2\gamma n} \geq \hat{v}_n(\pi^*) - \frac{\log\left(2|\Pi|/\delta\right)}{2\gamma n} \\ &\geq \rho(\pi^*) - \frac{\log\left(2|\Pi|/\delta\right)}{\gamma n} - 2\gamma C_\gamma(\pi^*).\end{aligned}$$

Concretely, the first of these inequalities follows from Lemma 6.3.2, the second one from the definition of the algorithm, and the third one from Lemma 6.3.3. This concludes the proof. □

6.4 A PAC-Bayesian extension

Our previously stated results require the policy class Π to be finite, and scale with $\log|\Pi|$. While this is a common assumption in past work on the subject (e.g., in (Dudík et al. 2011; Agarwal et al. 2014; L. Wang, Krishnamurthy, and Slivkins 2023)), it is of course not satisfied in most practical scenarios of interest. Several extensions have been proposed in previous work, mostly based on the idea of replacing the union bound over policies by more sophisticated uniform-convergence arguments: for instance, Swaminathan and Joachims (2015) and Y. Jin, Ren, et al. (2022) respectively show bounds that depend on the covering number and the Natarajan dimension of the policy class. In this section, we provide an extension that makes use of so-called *PAC-Bayesian* generalization bounds (McAllester 1998; Audibert 2004; Catoni 2007) that hold for arbitrary policy classes and often lead to meaningful performance guarantees even in large-scale settings of practical interest. We refer to the recent monograph of Alquier (2021) for a gentle introduction into the subject.

Before providing this extension, we will require some additional definitions. In this section, we will consider *randomized* algorithms that output a distribution $\widehat{Q}_n \in \Delta_\Pi$ over policies, and we will be interested in the performance guarantees that hold on expectation with respect to the random choice of $\hat{\pi}_n \sim \widehat{Q}_n$, but still hold with high probablity with respect to the realization of the random data set. We overload our notation slightly by defining $\rho(Q) = \int \rho(\pi)\mathrm{d}Q(\pi)$, $\hat{v}_n(Q) = \int \hat{v}_n(\pi)\mathrm{d}Q(\pi)$, $C_\gamma(Q) = \int C_\gamma(\pi)\mathrm{d}Q(\pi)$, and $\mathfrak{R}_n(Q) = \int \mathfrak{R}_n(\pi)\mathrm{d}Q(\pi)$, which all capture relevant quantities evaluated on expectation under the distribution $Q \in \Delta_\Pi$.

In the context of offline learning, several works have applied PAC-Bayesian techniques to provide concentration bounds for the importance-weighted estimator $\hat{v}_n(Q)$, characterizing its deviations from its true mean $\rho(Q)$ uniformly for all "posteriors" Q—we refer to the recent work of Sakhi,

Alquier, and Chopin (2023) and the survey of Flynn et al. (2023) for an extensive overview of such results. One common feature of these works is that they all provide concentration bounds derived from PAC-Bayesian versions of standard bounds like Hoeffding's or Bernstein's inequality, and as such suffer from the same limitations as the results described in **??**. The biggest such limitation is that all bounds require a uniform coverage assumption $\inf_{x,a} \mu(a|x) \geq \alpha$, or work with biased estimates of $\rho(Q)$ without quantifying the effect of the bias on the learning performance. Instead of deriving regret bounds from the concentration bounds, the focus in these works is to derive implementable algorithms from the concentration bounds and test them extensively in large-scale settings.

Here, we provide a natural extension of PIWO-IX that is derived from PAC-Bayesian principles. For defining our algorithm, we let $P \in \Delta_\Pi$ be an arbitrary "prior" over the policy class Π and define the output distribution as

$$\widehat{Q}_n = \arg\max_{Q \in \Delta_\pi} \left\{ \hat{v}_n(Q) - \frac{\mathrm{KL}\left(Q \| P\right)}{\lambda} \right\},$$

where $\mathrm{KL}\left(Q\|P\right) = \int \log \frac{\mathrm{d}Q}{\mathrm{d}P} \mathrm{d}Q$ is the *Kullback–Leibler divergence* (or *relative entropy*) between the distributions Q and P, and $\lambda > 0$ is a regularization parameter. It is well known that this distribution (often called the *Gibbs posterior*) has a closed-form expression with $\frac{\mathrm{d}\widehat{Q}_n}{\mathrm{d}P}(\pi) = \frac{e^{\lambda \hat{v}_n(\pi)}}{\int e^{\lambda \hat{v}_n(\pi')} \mathrm{d}P(\pi')}$. For practical purposes, we will simply choose $\lambda = 2\gamma n$ below. The following theorem establishes a regret guarantee for the resulting algorithm that we call *PAC-Bayesian PIWO-IX*.

Theorem 6.4.1. *With probability at least* $1-\delta$*, the regret of PAC-Bayesian PIWO-IX against any distribution* $Q^* \in \Delta_\Pi$ *over comparator policies satisfies*

$$\mathfrak{R}_n(Q^*) \leq \frac{\mathrm{KL}\left(Q^*\|P\right) + \log(1/\delta)}{\gamma n} + 2\gamma C_\gamma(Q^*).$$

Furthermore, by setting $\gamma = \sqrt{1/n}$*, the bound becomes*

$$\mathfrak{R}_n(Q^*) \leq \frac{2C_\gamma(Q^*) + \mathrm{KL}\left(Q^*\|P\right) + \log(1/\delta)}{\sqrt{n}}.$$

This bound inherits the key strength of PAC-Bayesian generalization bounds: it holds *uniformly for all competitors* Q^* without requiring a union bound over policies. We warn the reader familiar with PAC-Bayesian bounds though that the role of Q^* here is different from what they may expect: instead of being a data-dependent "posterior", it is a "comparator" distribution that the learner wishes to compete with. Thus, the bound expresses that distributions Q^* that are closer to the "prior" P in terms of relative entropy are "easier" to compete with. As before, the bound scales with the smoothed policy coverage ratio $C_\gamma(Q^*)$, only this time associated with the comparator distribution Q^*. Just like in the bound of Theorem 6.3.1, the bound requires no uniform coverage condition, and in particular continues to hold even if $\inf_{x,a} \mu(a|x)$ approaches zero. To our knowledge, this is the first regret bound for offline learning of such a PAC-Bayesian flavor, and in any case the first PAC-Bayesian bound for this setting that does not require uniform coverage.

The proof of Theorem 6.4.1 relies on the following generalizations of Lemmas 6.3.2 and 6.3.3:

Lemma 6.4.2. *With probability at least* $1-\delta$*, the following holds simultaneously for all* $Q \in \Delta_\Pi$*:*

$$\hat{v}_n(Q) - \rho(Q) \leq \frac{\mathrm{KL}\left(Q\|P\right) + \log\left(1/\delta\right)}{2\gamma n}.$$

Lemma 6.4.3. *With probability at least* $1-\delta$*, the following holds simultaneously for all* $Q \in \Delta_\Pi$*:*

$$\rho(Q) - \hat{v}_n(Q) \leq \frac{\mathrm{KL}\left(Q\|P\right) + \log\left(1/\delta\right)}{2\gamma n} + 2\gamma C_\gamma(Q).$$

The statements follow from combining the proofs of Lemmas 6.3.2 and 6.3.3 with a so-called "change-of-measure" trick commonly used in the PAC-Bayesian literature. We relegate the proofs to Appendix B.3 and only provide the very simple proof of Theorem 6.4.1 here.

Proof of Theorem 6.4.1 The statement follows from combining the above two lemmas via a union bound, and exploiting the definition of the

algorithm:

$$\rho(\widehat{Q}_n) \geq \hat{v}_n(\widehat{Q}_n) - \frac{\mathrm{KL}(\widehat{Q}_n \| P) + \log(1/\delta)}{2\gamma n} \geq \hat{v}_n(Q^*) - \frac{\mathrm{KL}\left(Q^* \| P\right) + \log(1/\delta)}{2\gamma n}$$
$$\geq \rho(Q^*) - \frac{\mathrm{KL}\left(Q^* \| P\right) + \log(1/\delta)}{\gamma n} - 2\gamma C_\gamma(Q^*).$$

Concretely, the first of these inequalities follows from Lemma 6.4.2, the second one from the definition of the algorithm, and the third one from Lemma 6.4.3. This concludes the proof. □

6.5 Adaptivity to the coverage

One shortcoming of the result in Theorem 6.3.1 is that it scales linearly with $C_\gamma(\pi^*)$ even though prior results suggest that a scaling with $\sqrt{C_0(\pi^*)}$ should be possible (Swaminathan and Joachims 2015; L. Wang, Krishnamurthy, and Slivkins 2023). This improvement can be trivially achieved by setting $\gamma = \sqrt{\frac{\log(|\Pi|/\delta)}{C_0(\pi^*)n}}$, but this requires prior knowledge of $C_0(\pi^*)$ which is of course unavailable in practice (at least in the most interesting case where π^* is the optimal policy).

This limitation can be addressed by defining the following *non-uniformly scaled* version of the IX estimator:

$$\hat{v}_n^\dagger(\pi) = \frac{1}{n}\sum_{t=1}^{n} \frac{\pi(A_t|X_t)}{\mu(A_t|X_t) + \gamma_\pi} \cdot R_t - \frac{\log\left(|\Pi|/\delta\right)}{2\gamma_\pi}. \tag{6.4}$$

Here, $\gamma_\pi > 0$ is a *policy-dependent* IX parameter that is potentially different for each policy π. Using this estimator, we define a variant of our main algorithm called *coverage-scaled PIWO-IX* that outputs

$$\hat{\pi}_n = \operatorname*{arg\,min}_{\pi \in \Pi} \hat{v}_n^\dagger(\pi).$$

Notice that, unlike PIWO-IX, this algorithm cannot be directly implemented using a standard optimization oracle due to the policy-dependent IX parameters γ_π. The following theorem is straightforward to prove using our previously established Lemma 6.3.2 Lemma 6.3.3:

Theorem 6.5.1. *With probability at least $1-\delta$, the regret of coverage-scaled PIWO-IX against any comparator policy $\pi^* \in \Pi$ satisfies*

$$\mathfrak{R}_n(\pi^*) \leq \frac{\log\left(2|\Pi|/\delta\right)}{\gamma_{\pi^*} n} + 2\gamma_{\pi^*} C_{\gamma_{\pi^*}}(\pi^*).$$

Furthermore, by setting $\gamma_\pi = \sqrt{\frac{\log(2|\Pi|/\delta)}{2C_0(\pi)n}}$ for each π, the bound becomes

$$\mathfrak{R}_n(\pi^*) \leq \sqrt{\frac{8C_0(\pi^*)\log\left(2|\Pi|/\delta\right)}{n}}.$$

Proof. First observe that the statements of Lemmas 6.3.2 and 6.3.3 can be trivially adjusted to show that the bounds

$$0 \leq \rho(\pi) - \hat{v}_n^\dagger(\pi) \leq \frac{\log(2|\Pi|/\delta)}{\gamma_\pi n} + 2\gamma C_{\gamma_\pi}(\pi).$$

hold simultaneously for all policies with probability at least $1-\delta$. Then, by the definition of the algorithm, we obtain

$$\rho(\hat{\pi}_n) \geq \hat{v}_n^\dagger(\hat{\pi}_n) \geq \hat{v}_n^\dagger(\pi^*) \geq \rho(\pi^*) - \frac{\log\left(2|\Pi|/\delta\right)}{\gamma_{\pi^*} n} - 2\gamma_{\pi^*} C_{\gamma_{\pi^*}}(\pi^*).$$

This concludes the proof of the first claim. The second claim can be verified by noticing that $C_\gamma(\pi^*) \leq C_0(\pi^*)$ for all $\gamma > 0$ and plugging in the choice of γ_π stated in the theorem. □

6.6 Experiments

In this section we provide a set of simple experiments that illustrate our theoretical findings, and in particular to empirically validate the robustness of our algorithm to hyper-parameter selection. We compare our method (PIWO-IX) to the method of L. Wang, Krishnamurthy, and Slivkins (2023) (here referred to as PIWO-PL), and follow an experimental setup that is directly inspired by theirs. Besides PIWO-PL, we also include a commonly used variant of our algorithm that uses the *clipped importance weights* (CIW) estimator defined as

$$\hat{v}_n(\pi) = \frac{1}{n}\sum_{t=1}^{n} \frac{\pi(A_t|X_t)}{\max\left\{\mu(A_t|X_t), \gamma\right\}} \cdot R_t.$$

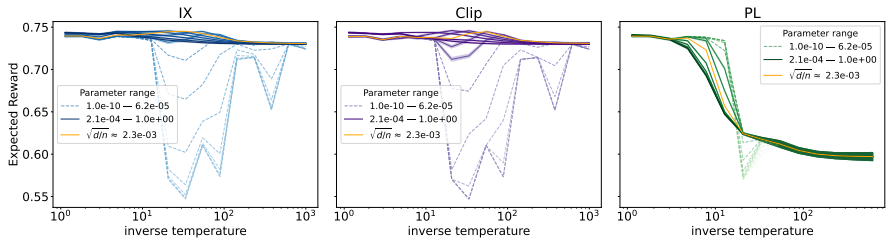

Figure 6.1: The performance of PIWO-IX, PIWO-Clip, and the algorithm of L. Wang, Krishnamurthy, and Slivkins (2023) as a function of the softmax parameter of the behavior policy. Different curves correspond to different hyperparameters for the algorithms, with lighter tones representing smaller hyperparameters and darker tones representing larger ones.

We refer to this metod as PIWO-Clip.

We use the Letter (OpenML ID 247[4]) classification dataset to simulate an offline contextual bandit instance. The dataset contains one milion entries, each consisting of 16 features and a true label, representing one of the $K = 26$ letters of the alphabet. To simulate a contextual bandit instance, we consider the feature vectors as contexts and the true labels as the corresponding optimal actions. To simulate the rewards we build a reward matrix $M \in \mathbb{R}^{K \times K}$ with entries on the diagonal set to 1 and the rest of them uniformly sampled from the $[0, 1)$ interval, and we keep these random parameters fixed for all repetitions. We then set the reward distribution $p(\cdot|x, a)$ for each context-action pair (x, a) as a Bernoulli distribution with parameter $M_{a,a*(x)}$, where $a^*(x)$ denotes the optimal action associated with context x.

The cost-sensitive classification oracle is implemented by fitting a multivariate ridge regressor, with one target for each action[5]. Given any context x, the regressor can be queried to predict the reward for each arm, and a *max* or *softmax* can be used to construct a policy to select the best arm. In order to generate a range of behavior policies, we retain 10% of the

[4] https://www.openml.org/search?type=data&status=active&id=247

[5] The choice of the regularization parameter α did not seem to impact significantly the result of the experiments.

data to train an estimator of the reward for each arm using the regressor described above with the true mean rewards as labels. We then use the predicted rewards to construct 20 softmax behavior policies, by varying the inverse temperature parameter as `logspace(-1, 3, 20)`.

We then collect an offline dataset using each of the behavior policies and train our method PIWO-IX, its variation PIWO-Clip, and the algorithm of L. Wang, Krishnamurthy, and Slivkins (2023), using the CSC oracle described above with an argmax to select the optimal action, and varying their hyper-parameter over a wide range (i.e. `logspace(-10, 0, 20)`). Finally we compute the expected reward for each combination of behavior policy and hyper-parameter, and show the result in Figure 6.1. It can be observed how most choices of hyper-parameters result in good performance for PIWO-IX and PIWO-Clip, while the same cannot be said for PIWO-PL, which is very sensitive to small probabilities in the behavior policy and needs to compensate them with a very careful choice of its hyper-parameter. In particular, we note that in some experiments with large softmax parameters, $\mu(a|x)$ can be as low as 10^{-100} for some context-action pairs, and thus even a seemingly negligible regularization parameter like $\beta = 10^{-20}$ can result in massive pessimistic adjustments. In contrast, PIWO-IX is robust to the presence of such small observation probabilities and continues to work well for a broad range of hyperparameter choices. As expected, PIWO-Clip performs very similarly to PIWO-IX due to the close similarity between these two methods. More details about the experiments are provided in Section 6.7.

6.7 Further details on the experiments

In this section we give more detail on all the experiments we ran. The first step we performed was to use 10% of the data to fit a multivariate ridge regressor $\texttt{reg}(x, a)$ to predict the expected reward of each action, given any context. For each context x and each corresponding optimal action a^* in the data, we selected $M_{\cdot,a^*}$ as the label vector (having one entry for each possible action).

We then used the reamining 90% of the data to perform two sets of experiments. In the first set, which is the one described in the main text

(Section 6.6), we considered 20 softmax behavior policies, varying their inverse temperature parameter η as `logspace(-1, 3, 20)`. That is,

$$\pi_\eta(a|x) \propto \exp(\eta\, \texttt{reg}(x, a)).$$

We repeated each set of experiments 10 times ($i \in [10]$), using a 10-fold validation procedure. That is, the data was first partitioned into 10 non overlapping folds. On each repetition i, 9 folds are used to generate the training data for the algorithms, by simulating the interaction of each behavior policy π_η and the bandit instance. The resulting training dataset $\mathcal{D}_{\eta,i}$ was used to train each algorithm for each possible hyper-parameter choice $h \in$ `logspace(-10, 0, 20))`. Finally, each trained algorithm $\mathfrak{A}_{\eta,i,h}$ is evaluated using the data in the remaining fold, by computing the expected regret using the true mean rewards.

This set of experiments was then repeated for a different set of "bad" behavior policies, which were defined as

$$\pi_\eta(a|x) \propto \exp(-\eta\, \texttt{reg}(x, a)).$$

The results for the two sets of experiments are shown respectively in Figures 6.2 and 6.3. On each figure, the first row of plots shows the expected reward as a function of the inverse temperature parameter η. Each plot on the row is for one of the three different algorithms, and it contains a line for each possible hyper-parameter. The lines are colored using a gradient from lighter to darker to represent increasing hyper-parameter values. In orange we highlighted the learning rate corresponding to $\sqrt{d/n}$, which we use as a crude approximation of the hyper-parameter recommended by theory, $\sqrt{\log|\Pi|/n}$. In addition, values of the hyper-parameters much smaller than $\sqrt{d/n}$ are represented with a dashed line. All lines (excluding for clarity of the representation the dashed ones) have a shaded region representing the standard deviation over the 10 runs. The second row of plots shows the expected regret as a function of the hyper-parameter h. Thus, we can observe a line for each different behavior policy parameter η. Here the lines are lighter for smaller values of η, and darker for bigger values of η.

From the plots, we can infer that PIWO-IX performs well when the behavior policy is "good" and γ is set in a broad proximity of its theoretically recommended value. This behavior appears to be robust as we vary the degree of "goodness" of the policy modulated by the softmax parameter η, and in particular performance stays good even as η approaches its higher extremes and the behavior policy gets more and more deterministic. As expected, PIWO-Clip behaves comparably. In comparison PIWO-PL is a lot less robust in this case and its performance decays as η increases, most likely due to the more and more extreme values of the importance weights arising from some sampling probabilities approaching zero. We note that the the case of "good" behavior policies is the most practical use case, and our experiments suggest that our algorithm performs excellently in this scenario for a wide range of hyperparameters.

In comparison, the picture changes when considering the case of "bad" behavior policies. In this case, PIWO-IX and performs worse and worse as γ is increased, especially for large values of η corresponding to particularly bad behavior policies. This is not surprising given that the policy coverage ratio blows up in this extreme, as less and less mass is put on well-performing actions. Also notice that increasing the regularization parameter γ forces the algorithm to be more and more pessimistic and thus stay closer and closer to the behavior policy, which again results in decaying performance. The performance of PIWO-PL is less consistent in this case, and it is hard to read out patterns that are well-predicted by theory.

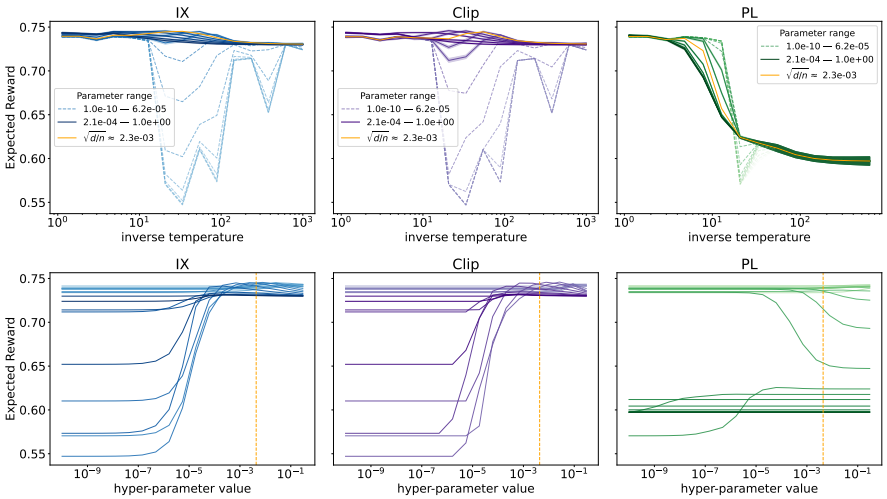

Figure 6.2: Results of PIWO-IX, PIWO-Clip, and PIWO-PL with good behavior policies.

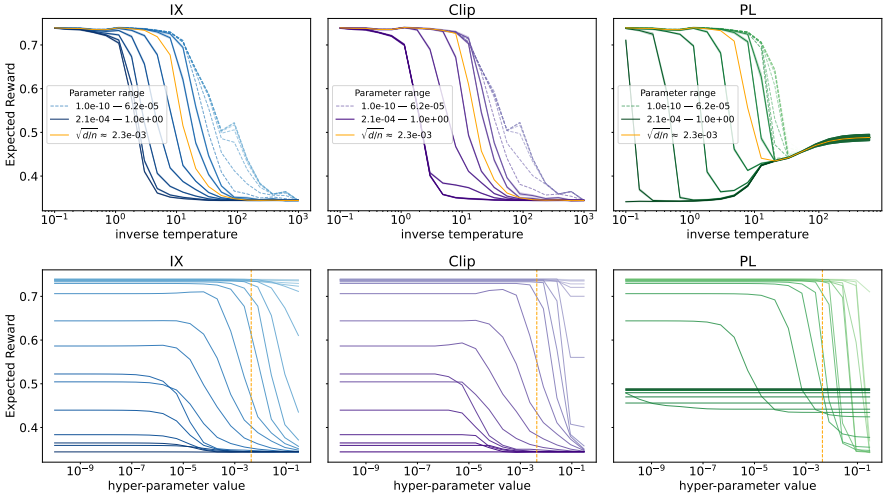

Figure 6.3: Results of PIWO-IX, PIWO-Clip, and PIWO-PL with bad behavior policies.

Chapter 7

Offline Learning in Linear Markov Decision Processes

In this chapter, we study the setting of Offline Reinforcement Learning, where the goal is to learn an ε-optimal policy without being able to interact with the environment, but only using a fixed dataset of transitions collected by a *behavior policy*. Learning from offline data proves to be useful especially when interacting with the environment can be costly or dangerous (Levine et al. 2020).

In this setting, the quality of the best policy learnable by any algorithm is constrained by the quality of the data, implying that finding an optimal policy without further assumptions on the data is not feasible. Therefore, many methods (Munos and Szepesvári 2008; Uehara, Huang, and Jiang 2020) make a *uniform coverage* assumption, requiring that the behavior policy explores sufficiently well the whole state-action space. However, recent work (Liu et al. 2020; Rashidinejad, B. Zhu, et al. 2022) demonstrated that *partial coverage* of the state-action space is sufficient. In particular, this means that the behavior policy needs only to sufficiently explore the state-action pairs visited by the optimal policy.

Moreover, like its online counterpart, modern offline RL faces the problem of learning efficiently in environments with very large state spaces, where function approximation is necessary to compactly represent policies and value functions. Although function approximation, especially with neural networks, is widely used in practice, its theoretical understanding in the context of decision-making is still rather limited, even when considering *linear* function approximation.

In fact, most existing sample complexity results for offline RL algorithms are limited either to the tabular and finite horizon setting, by the uniform coverage assumption or by assuming access to a (convex) optimization oracle — see the top section of Table 7.1 for a summary. Notable exceptions in terms of computational efficiency are the works of Xie, C. Cheng, et al. (2021) and C.-A. Cheng et al. (2022), who provide a computationally efficient version of their method for infinite-horizon discounted MDPs under realizable linear function approximation and partial coverage assumptions. Despite being some of the first concrete implementations, the practical versions of those algorithms differ significantly from their information-theoretic counterparts, and thus the sample-complexity guarantees proven in the corresponding papers do not immediately carry over to them.

More similar to our work are those of W. Zhan et al. (2022), and Rashidinejad, H. Zhu, et al. (2023) who also consider a linear programming approach to offline learning in infinite-horizon discounted MDPs. Yet, like many works which consider the broader general function approximation setting, their method may remain oracle-efficient even in the simpler linear MDP setting – see the caption of Table 7.1. Moreover, all methods referenced so far only work in the finite-horizon or infinite-horizon *discounted* setting, which is inappropriate for modeling practical problems where it is hard to pre-specify a fixed decision-making horizon. This issue is readily addressed by the average-reward framework, which however is known to be much more difficult to handle using techniques familiar from the discounted-reward setting. For example, methods based on approximate dynamic programming like H. Zhu, Rashidinejad, and Jiao (2023) make crucial use of the contractive property of the discounted Bellman operators, which does not generally hold in the average-reward setting (espe-

cially not under the general assumptions we make in our work). Therefore, this work is motivated by the following research question:

Can we design a linear-time algorithm with polynomial sample complexity for the discounted and average-reward infinite-horizon settings, in large state spaces under a partial-coverage assumption?

We answer this question positively by designing a method based on the linear-programming (LP) formulation of sequential decision making (Alan S Manne 1960). Albeit less known than the dynamic-programming formulation (Bellman 1956) that is ubiquitous in RL, it allows us to tackle this problem with the powerful tools of convex optimization. We turn in particular to a relaxed version of the LP formulation (Mehta and S. P. Meyn 2009; Bas-Serrano et al. 2021) that considers action-value functions that are linear in known state-action features. This allows to reduce the dimensionality of the problem from the cardinality of the state space to the number of features. This relaxation still allows to recover optimal policies in *linear MDPs* (L. Yang and M. Wang 2019; C. Jin et al. 2020), a structural assumption that is widely employed in the theoretical study of RL with linear function approximation.

Our algorithm for learning near-optimal policies from offline data is based on primal-dual optimization of the Lagrangian of the relaxed LP. The use of saddle-point optimization in MDPs was first proposed by M. Wang and Y. Chen (2016) for *planning* in small state spaces, and was extended to linear function approximation by Y. Chen, L. Li, and M. Wang (2018); Bas-Serrano and Neu (2020), and Neu and Okolo (2023). We largely take inspiration from this latter work, which was the first to apply saddle-point optimization to the *relaxed* LP. However, primal-dual planning algorithms assume oracle access to a transition model, whose samples are used to estimate gradients. In our offline setting, we only assume access to i.i.d. samples generated by a possibly unknown behavior policy. To adapt the primal-dual optimization strategy to this setting we employ a change of variable, inspired by Nachum and Dai (2020), which allows easy computation of unbiased gradient estimates.

Algorithm	Partial Coverage	Sample Comp.	Comput. Comp.	Function Approx.	Infinite Horizon γ	Infinite Horizon Avg
PEVI (Y. Jin, Z. Yang, and Z. Wang 2021)	✓	$O(\varepsilon^{-2})$	$O(n)$	general	✗	✗
FQI (Munos and Szepesvári 2008)	✗	$O(\varepsilon^{-2})$	oracle based	general	✓	✗
PSPI, practical (Xie, C. Cheng, et al. 2021)	✓	$O(\varepsilon^{-5})$ / $O(\varepsilon^{-3})$	oracle based	general / linear	✓	✗
PRO-RL (W. Zhan et al. 2022)	✓	$O(\varepsilon^{-6})$	oracle based	general	✓	✗
ALMIS (Rashidinejad, H. Zhu, et al. 2023)	✓	$O(\varepsilon^{-2})$	oracle based	general	✓	✗
A-CRAB (H. Zhu, Rashidinejad, and Jiao 2023)	✓	$O(\varepsilon^{-2})$	oracle based	general	✓	✗
PDOR (ours)	✓	$O(\varepsilon^{-4})$	$O(n)$	linear	✓	✓

Table 7.1: Comparison of selected methods for offline RL. The table shows some of the most relevant works for offline RL, and their characteristics. It is important to notice that many of these methods are designed for the general function approximation setting, while we focus on the easier setting of linear MDPs. However, most existing methods make use of oracles, which makes their computational complexity difficult to estimate, and while an efficient implementation can be derived by replacing the oracles appropriately, it is usually not immediate to prove sample complexity results for these practical versions.

7.1 Preliminaries

Our work is based on the linear programming formulation due to Alan S. Manne (1960) (see also Puterman (1994)) which transforms the reinforcement learning problem into the search for an optimal state-action occupancy measure, obtained by solving the following Linear Program (LP):

$$\begin{aligned} &\text{maximize} && \langle \mathbf{r}, \mathbf{p} \rangle \\ &\text{subject to} && \mathbf{E}^\intercal \mathbf{p} = (1-\gamma)\boldsymbol{\nu}_0 + \gamma \mathbf{P}^\intercal \mathbf{p} \\ & && \mathbf{p} \succeq 0 \end{aligned} \tag{7.1}$$

where $\mathbf{E} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times|\mathcal{X}|}$ denotes the matrix with components $\mathbf{E}_{(x,a),x'} \doteq \mathbb{1}\{x = x'\}$. The constraints of this LP are known to characterize the set of valid state-action occupancy measures. Therefore, an optimal solution $\mathbf{p}^*$ of the LP corresponds to the state-action occupancy measure associated to a policy π^* maximizing the expected return, and which is therefore optimal in the MDP. This policy can be extracted as $\pi^*(a \mid x) \doteq p^*(x,a) / \sum_{\bar{a}\in\mathcal{A}} p^*(x,\bar{a})$. However, this linear program cannot be directly solved in an efficient way in large MDPs due to the number of constraints and dimensions of the variables scaling with the size of the state space $\mathcal{X}$. Therefore, taking inspiration from the previous works of Bas-Serrano et al. (2021); Neu and Okolo (2023) we assume the knowledge of a *feature map* φ, which we then use to reduce the dimension of the problem. More specifically we consider the setting of Linear MDPs (L. Yang and M. Wang 2019; C. Jin et al. 2020).

Definition 7.1.1 (Linear MDP)**.** An MDP is called linear if both the transition and reward functions can be expressed as a linear function of a given feature map $\varphi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$. That is, there exist $\psi : \mathcal{X} \to \mathbb{R}^d$ and $\boldsymbol{\theta}_r \in \mathbb{R}^d$ such that, for every $x, x' \in \mathcal{X}$ and $a \in \mathcal{A}$:

$$r(x,a) = \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}_r \rangle, \quad P(x' \mid x, a) = \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\psi}(x') \rangle.$$

We assume that for all x, a, the norms of all relevant vectors are bounded by known constants as $\|\boldsymbol{\varphi}(x,a)\|_2 \le D_{\boldsymbol{\varphi}}$, $\left\|\sum_{x'} \boldsymbol{\psi}(x')\right\|_2 \le D_{\boldsymbol{\psi}}$, and $\|\boldsymbol{\theta}_r\|_2 \le D_{\boldsymbol{\theta}_r}$. Moreover, we represent the feature map with the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{X}\times\mathcal{A}|\times d}$ with rows given by $\boldsymbol{\varphi}(x,a)^\intercal$, and similarly we define $\boldsymbol{\Psi} \in \mathbb{R}^{d\times|\mathcal{X}|}$ as the matrix with columns given by $\boldsymbol{\psi}(x)$.

With this notation we can rewrite the transition matrix as $\mathbf{P} = \mathbf{\Phi}\mathbf{\Psi}$. Furthermore, it is convenient to assume that the dimension d of the feature map cannot be trivially reduced, and therefore that the matrix $\mathbf{\Phi}$ is full-rank. An easily verifiable consequence of the Linear MDP assumption is that state-action value functions can be represented as a linear combinations of φ. That is, there exist $\boldsymbol{\theta}^\pi \in \mathbb{R}^d$ such that:

$$\mathbf{q}^\pi = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}^\pi = \mathbf{\Phi}(\boldsymbol{\theta}_r + \mathbf{\Psi}\mathbf{v}^\pi) = \mathbf{\Phi}\boldsymbol{\theta}^\pi. \tag{7.2}$$

It can be shown that for all policies π, the norm of $\boldsymbol{\theta}^\pi$ is at most $D_{\boldsymbol{\theta}} = D_{\boldsymbol{\theta}_r} + \frac{D_\Psi}{1-\gamma}$ (cf. Lemma B.1 in C. Jin et al. (2020)). We then translate the linear program Equation (7.1) to our setting, with the addition of the new variable $\boldsymbol{\lambda} \in \mathbb{R}^d$, resulting in the following new LP and its corresponding dual:

$$\begin{aligned} \text{maximize} \quad & \langle \boldsymbol{\theta}_r, \boldsymbol{\lambda} \rangle \\ \text{subject to} \quad & \mathbf{E}^\intercal \mathbf{p} = (1-\gamma)\boldsymbol{\nu}_0 + \gamma \mathbf{\Psi}^\intercal \boldsymbol{\lambda} \\ & \boldsymbol{\lambda} = \mathbf{\Phi}^\intercal \mathbf{p} \\ & \mathbf{p} \succeq 0, \end{aligned} \tag{7.3}$$

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \boldsymbol{\nu}_0, \mathbf{v} \rangle \\ \text{subject to} \quad & \boldsymbol{\theta} = \boldsymbol{\theta}_r + \gamma \mathbf{\Psi}\mathbf{v} \\ & \mathbf{E}\mathbf{v} \succeq \mathbf{\Phi}\boldsymbol{\theta}. \end{aligned} \tag{7.4}$$

It can be immediately noticed how the introduction of $\boldsymbol{\lambda}$ did not change neither the set of admissible $\mathbf{p}$s nor the objective, and therefore did not alter the optimal solution. The Lagrangian associated to this set of linear programs is the function:

$$\begin{aligned} \mathfrak{L}(\mathbf{v}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{p}) &= (1-\gamma)\langle \boldsymbol{\nu}_0, \mathbf{v} \rangle + \langle \boldsymbol{\lambda}, \boldsymbol{\theta}_r + \gamma \mathbf{\Psi}\mathbf{v} - \boldsymbol{\theta} \rangle \\ &\quad + \langle \mathbf{p}, \mathbf{\Phi}\boldsymbol{\theta} - \mathbf{E}\mathbf{v} \rangle \qquad (7.5) \\ &= \langle \boldsymbol{\lambda}, \boldsymbol{\theta}_r \rangle + \langle \mathbf{v}, (1-\gamma)\boldsymbol{\nu}_0 + \gamma \mathbf{\Psi}^\intercal \boldsymbol{\lambda} - \mathbf{E}^\intercal \mathbf{p} \rangle \\ &\quad + \langle \boldsymbol{\theta}, \mathbf{\Phi}^\intercal \mathbf{p} - \boldsymbol{\lambda} \rangle. \qquad (7.6) \end{aligned}$$

It is known that finding optimal solutions $(\boldsymbol{\lambda}^\star, \mathbf{p}^\star)$ and $(\mathbf{v}^\star, \boldsymbol{\theta}^\star)$ for the primal and dual LPs is equivalent to finding a saddle point $(\mathbf{v}^\star, \boldsymbol{\theta}^\star, \boldsymbol{\lambda}^\star, \mathbf{p}^\star)$

of the Lagrangian function (Bertsekas 1982). In the next section, we will develop primal-dual methods that aim to find approximate solutions to the above saddle-point problem, and convert these solutions to policies with near-optimality guarantees.

7.2 Algorithm and Main Results

This section introduces the concrete setting we study in this paper, and presents our main contributions.

We consider the offline-learning scenario where the agent has access to a dataset $D = (W_t)_{t=1}^n$, collected by a behavior policy μ, and composed of n random observations of the form $W_t = (X_t^0, X_t, A_t, R_t, X_t')$. The random variables $X_t^0, (X_t, A_t)$ and X_t' are sampled, respectively, from the initial-state distribution ν_0, the discounted occupancy measure of the behavior policy, denoted as p^μ, and from $P(\cdot \mid X_t, A_t)$. Finally, R_t denotes the reward $r(X_t, A_t)$. We assume that all observations W_t are generated independently of each other, and will often use the notation $\boldsymbol{\varphi}_t = \boldsymbol{\varphi}(X_t, A_t)$.

Our strategy consists in finding approximately good solutions for the Equations (7.3) and (7.4) using stochastic optimization methods, which require access to unbiased gradient estimates of the Lagrangian (Equation (7.6)). The main challenge we need to overcome is constructing suitable estimators based only on observations drawn from the behavior policy.

We address this challenge by using the matrix $\boldsymbol{\Lambda}_\mu$, defined in Equation (4.12) (supposed to be invertible for the sake of argument for now), and rewriting the gradient with respect to $\boldsymbol{\lambda}$ as

$$\begin{aligned}
&\nabla_{\boldsymbol{\lambda}} \mathfrak{L}(\boldsymbol{\lambda}, \mathbf{p}; \mathbf{v}, \boldsymbol{\theta}) = \boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi} \mathbf{v} - \boldsymbol{\theta} \\
&= \boldsymbol{\Lambda}_\mu^{-1} \boldsymbol{\Lambda}_\mu \left(\boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi} \mathbf{v} - \boldsymbol{\theta} \right) \\
&= \boldsymbol{\Lambda}_\mu^{-1} \mathbb{E} \left[\boldsymbol{\varphi}(X_t, A_t) \boldsymbol{\varphi}(X_t, A_t)^\intercal \left(\boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi} \mathbf{v} - \boldsymbol{\theta} \right) \right] \\
&= \boldsymbol{\Lambda}_\mu^{-1} \mathbb{E} \left[\boldsymbol{\varphi}(X_t, A_t) \left(R_t + \gamma \mathbf{v}(X_t') - \langle \boldsymbol{\theta}, \boldsymbol{\varphi}(X_t, A_t) \rangle \right) \right].
\end{aligned}$$

This suggests that the vector within the expectation can be used to build an unbiased estimator of the desired gradient. A downside of using this

estimator is that it requires knowledge of $\mathbf{\Lambda}_\mu$. However, this can be sidestepped by a reparametrization trick inspired by Nachum and Dai (2020): introducing the parametrization $\boldsymbol{\beta} = \mathbf{\Lambda}_\mu^{-1}\boldsymbol{\lambda}$, the objective can be rewritten as

$$\begin{aligned}\mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}; \mathbf{v}, \boldsymbol{\theta}) = (1-\gamma)\langle \boldsymbol{\nu}_0, \mathbf{v}\rangle + \langle \boldsymbol{\beta}, \mathbf{\Lambda}_\mu(\boldsymbol{\theta}_r + \gamma \mathbf{\Psi}\mathbf{v} - \boldsymbol{\theta})\rangle \\ + \langle \mathbf{p}, \mathbf{\Phi}\boldsymbol{\theta} - \mathbf{E}\mathbf{v}\rangle.\end{aligned}$$

This can be indeed seen to generalize the tabular reparametrization of Nachum and Dai (2020) to the case of linear function approximation. Notably, our linear reparametrization does not change the structure of the saddle-point problem, but allows building an unbiased estimator of $\nabla_{\boldsymbol{\beta}}\mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}; \mathbf{v}, \boldsymbol{\theta})$ without knowledge of $\mathbf{\Lambda}_\mu$ as

$$\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} = \boldsymbol{\varphi}(X_t, A_t)\left(R_t + \gamma \mathbf{v}(X_t') - \langle \boldsymbol{\theta}, \boldsymbol{\varphi}(X_t, A_t)\rangle\right).$$

In what follows, we will use the more general parametrization $\boldsymbol{\beta} = \Lambda^{-c}\lambda$, with $c \in \{1/2, 1\}$, and construct a primal-dual stochastic optimization method that can be implemented efficiently in the offline setting based on the observations above. Using $c = 1$ allows to run our algorithm without knowledge of $\mathbf{\Lambda}_\mu$, that is, without knowing the behavior policy that generated the dataset, while using $c = 1/2$ results in a tighter bound[1], at the price of having to assume knowledge of $\mathbf{\Lambda}_\mu$.

Our algorithm (presented as Algorithm 3) is inspired by the method of Neu and Okolo (2023), originally designed for planning with a generative model. The algorithm has a double-loop structure, where at each iteration t we run one step of stochastic gradient ascent for $\boldsymbol{\beta}$, and also an inner loop which runs K iterations of stochastic gradient descent on $\boldsymbol{\theta}$ making sure that $\langle \boldsymbol{\varphi}(x, a), \boldsymbol{\theta}_t\rangle$ is a good approximation of the true action-value function of π_t. Iterations of the inner loop are indexed by k. The main idea of the algorithm is to compute the unbiased estimators $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k}$ and $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}$ of the gradients $\nabla_{\boldsymbol{\theta}}\mathfrak{L}(\boldsymbol{\beta}_t, \mathbf{p}_t; \cdot, \boldsymbol{\theta}_{t,k})$ and $\nabla_{\boldsymbol{\beta}}\mathfrak{L}(\boldsymbol{\beta}_t, \cdot; \mathbf{v}_t, \boldsymbol{\theta}_t)$, and use them to update the respective variables iteratively. We then define a softmax

[1] By "tighter bound" we refer to dependence on the coverage ratio introduced in Definition 4.7.1. We give more details on this in Section 8.3.

policy π_t at each iteration t using the $\boldsymbol{\theta}$ parameters as

$$\pi_t(a \mid x) = \frac{\exp\left(\alpha \sum_{i=1}^{t-1} \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}_i \rangle\right)}{\sum_{a'} \exp\left(\alpha \sum_{i=1}^{t-1} \langle \boldsymbol{\varphi}(x,a'), \boldsymbol{\theta}_i \rangle\right)}$$

The other higher-dimensional variables $(\mathbf{p}_t, \mathbf{v}_t)$ are defined symbolically in terms of $\boldsymbol{\beta}_t$, $\boldsymbol{\theta}_t$ and π_t, and used only as auxiliary variables for computing the estimates $\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k}$ and $\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}$. Specifically, we set these variables as

$$v_t(x) = \sum_{a \in \mathcal{A}} \pi_t(a \mid x) \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}_t \rangle, \tag{7.7}$$

$$\begin{aligned} p_{t,k}(x,a) = \pi_t(a \mid x)((1-\gamma)\mathbb{1}\{X^0_{t,k} = x\} \\ + \gamma \langle \boldsymbol{\varphi}_{t,k}, \boldsymbol{\Lambda}_\mu^{c-1} \boldsymbol{\beta}_t \rangle \mathbb{1}\{X'_{t,k} = x\}). \end{aligned} \tag{7.8}$$

Finally, the gradient estimates can be defined as

$$\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} = \boldsymbol{\Lambda}_\mu^{c-1} \boldsymbol{\varphi}_t \left(R_t + \gamma v_t(X'_t) - \langle \boldsymbol{\varphi}_t, \boldsymbol{\theta}_t \rangle\right), \tag{7.9}$$

$$\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k} = \boldsymbol{\Phi}^\top \mathbf{p}_{t,k} - \boldsymbol{\Lambda}_\mu^{c-1} \boldsymbol{\varphi}_{t,k} \langle \boldsymbol{\varphi}_{t,k}, \boldsymbol{\beta}_t \rangle. \tag{7.10}$$

These gradient estimates are then used in a projected gradient ascent/descent scheme, with the ℓ_2 projection operator denoted by Π. The feasible sets of the two parameter vectors are chosen as ℓ_2 balls of radii D_θ and D_β, denoted respectively as $\mathbb{B}(D_\theta)$ and $\mathbb{B}(D_\beta)$. Notably, the algorithm does not need to compute $v_t(x)$, $p_{t,k}(x,a)$, or $\pi_t(a|x)$ for all states x, but only for the states that are accessed during the execution of the method. In particular, π_t does not need to be computed explicitly, and it can be efficiently represented by the single d-dimensional parameter vector $\sum_{i=1}^t \boldsymbol{\theta}_i$.

Due to the double-loop structure, each iteration t uses K samples from the dataset D, adding up to a total of $n = KT$ samples over the course of T iterations. Each gradient update calculated by the method uses a constant number of elementary vector operations, resulting in a total computational complexity of $O(|\mathcal{A}|dn)$ elementary operations. At the end, our algorithm outputs a policy selected uniformly at random from the T iterations.

7.2.1 Main result

We are now almost ready to state our main result. Before doing so, we first need to discuss the quantities appearing in the guarantee, and provide an

Input: Learning rates α, ζ, η, initial points
$\boldsymbol{\theta}_0 \in \mathbb{B}(D_\theta), \boldsymbol{\beta}_1 \in \mathbb{B}(D_\beta), \pi_1$, and data $D = (W_t)_{t=1}^n$
for $t = 1$ ***to*** T **do**
  Initialize $\boldsymbol{\theta}_{t,1} = \boldsymbol{\theta}_{t-1}$
  for $k = 1$ ***to*** $K - 1$ **do**
    Obtain sample $W_{t,k} = (X^0_{t,k}, X_{t,k}, A_{t,k}, X'_{t,k})$
    $\mathbf{p}_{t,k} = \pi_t \circ [(1-\gamma)\mathbf{e}_{X^0_{t,k}} + \gamma\langle \boldsymbol{\varphi}(X_{t,k}, A_{t,k}), \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle \mathbf{e}_{X'_{t,k}}]$
    $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,i} = \boldsymbol{\Phi}^\intercal \mathbf{p}_{t,k} - \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}(X_{t,k}, A_{t,k})\langle \boldsymbol{\varphi}(X_{t,k}, A_{t,k}), \boldsymbol{\beta}_t\rangle$
    $\boldsymbol{\theta}_{t,k+1} = \Pi_{\mathbb{B}(D_\theta)}(\boldsymbol{\theta}_{t,k} - \eta\hat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,i})$ *// Stochastic gradient descent*
  end
  $\boldsymbol{\theta}_t = \frac{1}{K}\sum_{k=1}^K \boldsymbol{\theta}_{t,k}$
  Obtain sample $W_t = (X^0_t, X_t, A_t, X'_t)$
  $\mathbf{v}_t = \mathbf{E}^\intercal(\pi_t \circ \boldsymbol{\Phi}\boldsymbol{\theta}_t)$
  $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} = \boldsymbol{\Lambda}c - 1\boldsymbol{\varphi}(X_t, A_t)(R_t + \gamma\mathbf{v}_t(X'_t) - \langle \boldsymbol{\varphi}(X_t, A_t), \boldsymbol{\theta}_t\rangle)$
  $\boldsymbol{\beta}_{t+1} = \Pi_{\mathbb{B}(D_\beta)}(\boldsymbol{\beta}_t + \zeta\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t})$ *// Stochastic gradient ascent*
  $\pi_{t+1} = \sigma(\alpha\sum_{i=1}^t \boldsymbol{\Phi}\boldsymbol{\theta}_i)$ *// Policy update*
end
Output: π_J with $J \sim \mathcal{U}(T)$

Algorithm 3: Primal-Dual Offline RL (PDOR)

intuitive explanation for them.

Similarly to previous work, we capture the partial coverage assumption by expressing the rate of convergence to the optimal policy in terms of a *coverage ratio* that measures the mismatch between the behavior and the optimal policy. Several definitions of coverage ratio are surveyed by Uehara and Sun (2022). In this work, we employ the notion of *feature* coverage ratio for linear MDPs defined in Definition 4.7.1.

We defer a detailed discussion of this ratio to Section 8.3, where we compare it with similar notions in the literature. We are now ready to state our main result.

Theorem 7.2.1. *Consider a linear MDP (Definition 7.1.1) such that* $\boldsymbol{\theta}^\pi \in \mathbb{B}(D_{\boldsymbol{\theta}})$ *for all* $\pi \in \Pi$. *Further, suppose that* $C_{\varphi,c}(\pi^*) \leq D_{\boldsymbol{\beta}}$. *Then, for*

any comparator policy $\pi^* \in \Pi$*, the policy output by Algorithm 3 satisfies:*

$$\mathbb{E}\left[\langle \mathbf{p}^{\pi^*} - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r}\rangle\right] \leq \frac{2D_{\boldsymbol{\beta}}^2}{\zeta T} + \frac{\log|\mathcal{A}|}{\alpha T} + \frac{2D_{\boldsymbol{\theta}}^2}{\eta K} + \frac{\zeta G_{\boldsymbol{\beta},c}^2}{2} + \frac{\alpha D_{\boldsymbol{\theta}}^2 D_{\boldsymbol{\varphi}}^2}{2} + \frac{\eta G_{\boldsymbol{\theta},c}^2}{2},$$

where:

$$G_{\boldsymbol{\theta},c}^2 = 3D_{\boldsymbol{\varphi}}^2\left((1-\gamma)^2 + (1+\gamma^2)D_{\boldsymbol{\beta}}^2\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}\right), \tag{7.11}$$

$$G_{\boldsymbol{\beta},c}^2 = 3(1+(1+\gamma^2)D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2)D_{\boldsymbol{\varphi}}^{2(2c-1)}. \tag{7.12}$$

In particular, using learning rates

$$\eta = \frac{2D_{\boldsymbol{\theta}}}{G_{\boldsymbol{\theta},c}\sqrt{K}}, \quad \zeta = \frac{2D_{\boldsymbol{\beta}}}{G_{\boldsymbol{\beta},c}\sqrt{T}}, \quad \alpha = \frac{\sqrt{2\log|\mathcal{A}|}}{D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}}\sqrt{T}},$$

and setting

$$K = T \cdot \frac{2D_{\boldsymbol{\beta}^2}G_{\boldsymbol{\beta},c}^2 + D_{\boldsymbol{\theta}}^2 D_{\boldsymbol{\varphi}}^2 \log|\mathcal{A}|}{2D_{\boldsymbol{\theta}}^2 G_{\boldsymbol{\theta},c}^2}$$

we achieve $\mathbb{E}\left[\langle \mathbf{p}^{\pi^*} - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r}\rangle\right] \leq \epsilon$ *with a number of samples* n_ϵ *that is*

$$O\left(\epsilon^{-4}D_{\boldsymbol{\theta}}^4 D_{\boldsymbol{\varphi}}^4 D_{\boldsymbol{\beta}}^4 \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}\log|\mathcal{A}|\right).$$

By Remark 7.2.2 below, we have that n_ϵ is simply of order

$$O\left(\varepsilon^{-4}D_{\boldsymbol{\theta}}^4 D_{\boldsymbol{\varphi}}^{8c} D_{\boldsymbol{\beta}}^4 d^{2-2c}\log|\mathcal{A}|\right).$$

Remark 7.2.2. When $c = 1/2$, the factor $\operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})$ is just d, the feature dimension, and $\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1} = 1$. When $c = 1$ and $\boldsymbol{\Lambda}_\mu$ is unknown, both $\left\|\boldsymbol{\Lambda}_\mu\right\|_2$ and $\operatorname{Tr}(\boldsymbol{\Lambda}_\mu)$ should be replaced by their upper bound $D_{\boldsymbol{\varphi}}^2$. Then, for $c \in \{1/2, 1\}$, we have that $\operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1} \leq D_{\boldsymbol{\varphi}}^{8c-4}d^{2-2c}$.

The main theorem can be simplified by making some standard assumptions, formalized by the following corollary.

Corollary 7.2.3. *Assume that the bound of the feature vectors* $D_{\boldsymbol{\varphi}}$ *is of order* $O(1)$*, that* $D_{\boldsymbol{\theta}_r} = D_{\boldsymbol{\psi}} = \sqrt{d}$ *and that* $D_{\boldsymbol{\beta}} = c \cdot C_{\varphi,c}(\pi^*)$ *for some positive universal constant* c*. Then, under the same assumptions of Theorem 7.2.1,* n_ε *is of order*

$$O\Big(\frac{d^4 C_{\varphi,c}(\pi^*)^2 \log|\mathcal{A}|}{d^{2c}(1-\gamma)^4\varepsilon^4}\Big).$$

7.3 Analysis

This section explains the rationale behind some of the technical choices of our algorithm, and sketches the proof of our main result.

First, we explicitly rewrite the expression of the Lagrangian (Equation (7.6)), after performing the change of variable $\boldsymbol{\lambda} = \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}$:

$$\begin{aligned}\mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}; \mathbf{v}, \boldsymbol{\theta}) &= (1-\gamma)\langle \boldsymbol{\nu}_0, \mathbf{v}\rangle + \langle \boldsymbol{\beta}, \boldsymbol{\Lambda}_\mu^c(\boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi}\mathbf{v} - \boldsymbol{\theta})\rangle \\ &\quad + \langle \mathbf{p}, \boldsymbol{\Phi}\boldsymbol{\theta} - \mathbf{E}\mathbf{v}\rangle \qquad (7.13) \\ &= \langle \boldsymbol{\beta}, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\theta}_r\rangle + \langle \mathbf{v}, (1-\gamma)\boldsymbol{\nu}_0 + \gamma \boldsymbol{\Psi}^\intercal \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta} - \mathbf{E}^\intercal \mathbf{p}\rangle \\ &\quad + \langle \boldsymbol{\theta}, \boldsymbol{\Phi}^\intercal \mathbf{p} - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}\rangle. \qquad (7.14)\end{aligned}$$

We aim to find an approximate saddle-point of the above convex-concave objective function. One challenge that we need to face is that the variables $\mathbf{v}$ and $\mathbf{p}$ have dimension proportional to the size of the state space $|\mathcal{X}|$, so making explicit updates to these parameters would be prohibitively expensive in MDPs with large state spaces. To address this challenge, we choose to parametrize $\mathbf{p}$ in terms of a policy π and $\boldsymbol{\beta}$ through the symbolic assignment $\mathbf{p} = \mathbf{p}_{\boldsymbol{\beta},\pi}$, where

$$p_{\boldsymbol{\beta},\pi}(x,a) \doteq \pi(a|x)\Big[(1-\gamma)\nu_0(x) + \gamma\langle \boldsymbol{\psi}(x), \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}\rangle\Big].$$

This choice can be seen to satisfy the first constraint of the primal Equation (7.3), and thus the gradient of the Equation (7.14) evaluated at $\mathbf{p}_{\boldsymbol{\beta},\pi}$ with respect to $\mathbf{v}$ can be verified to be 0. This parametrization makes it possible to express the Lagrangian as a function of only $\boldsymbol{\theta}, \boldsymbol{\beta}$ and π as

$$\begin{aligned}f(\boldsymbol{\theta}, \boldsymbol{\beta}, \pi) &\doteq \mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}_{\boldsymbol{\beta},\pi}; \mathbf{v}, \boldsymbol{\theta}) \\ &= \langle \boldsymbol{\beta}, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\theta}_r\rangle + \langle \boldsymbol{\theta}, \boldsymbol{\Phi}^\intercal \mathbf{p}_{\boldsymbol{\beta},\pi} - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}\rangle. \qquad (7.15)\end{aligned}$$

For convenience, we also define the quantities $\boldsymbol{\nu}_{\boldsymbol{\beta}} = \mathbf{E}^\intercal \mathbf{p}_{\boldsymbol{\beta},\pi}$ and $v_{\boldsymbol{\theta},\pi}(s) \doteq \sum_a \pi(a|s)\,\langle \boldsymbol{\theta}, \boldsymbol{\varphi}(x,a)\rangle$, which enables us to rewrite f as

$$\begin{aligned}f(\boldsymbol{\theta}, \boldsymbol{\beta}, \pi) &= \langle \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}, \boldsymbol{\theta}_r - \boldsymbol{\theta}\rangle + \langle \mathbf{v}_{\boldsymbol{\theta},\pi}, \boldsymbol{\nu}_{\boldsymbol{\beta}}\rangle \\ &= (1-\gamma)\langle \boldsymbol{\nu}_0, \mathbf{v}_{\boldsymbol{\theta},\pi}\rangle \\ &\quad + \langle \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}, \boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi}\mathbf{v}_{\boldsymbol{\theta},\pi} - \boldsymbol{\theta}\rangle. \qquad (7.16)\end{aligned}$$

The above choices allow us to perform stochastic gradient / ascent over the low-dimensional parameters $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ and the policy π. In order to calculate an unbiased estimator of the gradients, we first observe that the choice of $p_{t,k}$ in Algorithm 3 is an unbiased estimator of $p_{\boldsymbol{\beta}_t,\pi_t}$:

$$\begin{aligned}
\mathbb{E}_{t,k}[p_{t,k}(x,a)] &= \pi_t(a \mid x)\Big((1-\gamma)\mathbb{P}(X^0_{t,k} = x) \\
&\qquad + \mathbb{E}_{t,k}[\mathbb{1}\{X'_{t,k} = x\}\langle \boldsymbol{\varphi}_t, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle]\Big) \\
&= \pi_t(a \mid x)\Big((1-\gamma)\nu_0(x) \\
&\qquad + \gamma \sum_{\bar{x},\bar{a}} p^\mu(\bar{x},\bar{a})P(x \mid \bar{x},\bar{a})\boldsymbol{\varphi}(\bar{x},\bar{a})^\intercal \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\Big) \\
&= \pi_t(a \mid x)\Big((1-\gamma)\nu_0(x) + \gamma\boldsymbol{\psi}(x)^\intercal \boldsymbol{\Lambda}_\mu \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\Big) \\
&= p_{\boldsymbol{\beta}_t,\pi_t}(x,a),
\end{aligned}$$

where we used the fact that $P(x \mid \bar{x},\bar{a}) = \langle \boldsymbol{\psi}(x), \boldsymbol{\varphi}(\bar{x},\bar{a})\rangle$, and the definition of $\boldsymbol{\Lambda}_\mu$. This in turn facilitates proving that the gradient estimate $\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k}$, defined in Equation (7.10), is indeed unbiased:

$$\begin{aligned}
\mathbb{E}_{t,k}[\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k}] &= \boldsymbol{\Phi}^\intercal \mathbb{E}_{t,k}[\mathbf{p}_{t,k}] - \boldsymbol{\Lambda}_\mu^{c-1}\mathbb{E}_{t,k}[\boldsymbol{\varphi}_{t,k}\boldsymbol{\varphi}_{t,k}^\intercal]\boldsymbol{\beta}_t \\
&= \boldsymbol{\Phi}^\intercal \mathbf{p}_{\boldsymbol{\beta}_t,\pi_t} - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t = \nabla_{\boldsymbol{\theta}}\mathfrak{L}(\boldsymbol{\beta}_t, \mathbf{p}_t; \mathbf{v}_t, \cdot).
\end{aligned}$$

A similar proof is used for $\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}$ and is detailed in Appendix C.3.

Our analysis is based on arguments by Neu and Okolo (2023), carefully adapted to the reparametrized version of the Lagrangian presented above. The proof studies the following central quantity that we refer to as *dynamic duality gap*:

$$\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}^*_{1:T}) \doteq \frac{1}{T}\sum_{t=1}^{T}(f(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}^*_t)).$$

Here, $(\boldsymbol{\theta}_t, \boldsymbol{\beta}_t, \pi_t)$ are the iterates of the algorithm, $\boldsymbol{\theta}^*_{1:T} = (\boldsymbol{\theta}^*_t)_{t=1}^T$ a sequence of comparators for $\boldsymbol{\theta}$, and finally $\boldsymbol{\beta}^*$ and π^* are fixed comparators for $\boldsymbol{\beta}$ and π, respectively. Our first key lemma relates the suboptimality of the output policy to $\mathcal{G}_T$ for a specific choice of comparators.

Lemma 7.3.1. *Let* $\boldsymbol{\theta}_t^* \doteq \boldsymbol{\theta}^{\pi_t}$, π^* *be any policy, and* $\beta^* = \boldsymbol{\Lambda}_\mu^{-c} \boldsymbol{\Phi}^\top \mathbf{p}^{\pi^*}$. *Then,* $\mathbb{E}\left[\langle \mathbf{p}^{\pi^*} - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r} \rangle\right] = \mathcal{G}_T(\beta^*, \pi^*; \boldsymbol{\theta}_{1:T}^*)$.

The proof is relegated to Appendix C.1. Our second key lemma rewrites the gap $\mathcal{G}_T$ for *any* choice of comparators as the sum of three regret terms:

Lemma 7.3.2. *With the choice of comparators of Lemma 7.3.1*

$$\begin{aligned}\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}_{1:T}^*) = \frac{1}{T} \sum_{t=1}^{T} \Big(&\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, g_{\boldsymbol{\theta},t} \rangle \\ &+ \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, g_{\boldsymbol{\beta},t} \rangle \\ &+ \sum_s \nu^{\pi^*}(s) \sum_a (\pi^*(a|s) - \pi_t(a|s)) \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}(x,a) \rangle \Big),\end{aligned}$$

where $g_{\boldsymbol{\theta},t} = \boldsymbol{\Phi}^\top \mathbf{p}_{\boldsymbol{\beta}_t, \pi_t} - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t$ *and* $g_{\boldsymbol{\beta},t} = \boldsymbol{\Lambda}_\mu^c (\boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi} v_{\boldsymbol{\theta}_t, \pi_t} - \boldsymbol{\theta}_t)$.

The proof is presented in Appendix C.2. To conclude the proof we bound the three terms appearing in Lemma 7.3.2. The first two of those are bounded using standard gradient descent/ascent analysis (Lemmas C.3.1 and C.3.2), while for the latter we use mirror descent analysis (Lemma C.3.3). The details of these steps are reported in Appendix C.3.

7.4 Extension to Average-Reward MDPs

In this section, we briefly explain how to extend our approach to offline learning in *average reward MDPs*, establishing the first sample complexity result for this setting. After introducing the setup, we outline a remarkably simple adaptation of our algorithm along with its performance guarantees for this setting. The reader is referred to Appendix E for the full details, and to Chapter 8 of Puterman (1994) for a more thorough discussion of average-reward MDPs.

In the average reward setting we aim to optimize the objective $\rho^\pi(x) = \liminf_{T \to \infty} \frac{1}{T} \mathbb{E}_\pi[\sum_{t=1}^T r(x_t, a_t) \mid x_1 = x]$, representing the long-term average reward of policy π when started from state $x \in \mathcal{X}$. Unlike the discounted setting, the average reward criterion prioritizes long-term frequency over proximity of good rewards due to the absence of discounting

which expresses a preference for earlier rewards. As is standard in the related literature, we will assume that ρ^π is well-defined for any policy and is independent of the start state, and thus will use the same notation to represent the scalar average reward of policy π. Due to the boundedness of the rewards, we clearly have $\rho^\pi \in [0, 1]$. Similarly to the discounted setting, it is possible to define quantities analogous to the value and action value functions as the solutions to the Bellman equations $\mathbf{q}^\pi = \mathbf{r} - \rho^\pi \mathbf{1} + \mathbf{P}\mathbf{v}^\pi$, where $\mathbf{v}^\pi$ is related to the action-value function as $v^\pi(x) = \sum_a \pi(a|x) q^\pi(x, a)$. We will make the following standard assumption about the MDP (see Section 17.4 of S. Meyn and Tweedie (1996)):

Assumption 7.4.1. For all stationary policies π, the Bellman equations have a solution $\mathbf{q}^\pi$ satisfying $\sup_{x,a} q^\pi(x, a) - \inf_{x,a} q^\pi(x, a) < D_q$.

Furthermore, we will continue to work with the linear MDP assumption of Definition 7.1.1, and will additionally make the following minor assumption:

Assumption 7.4.2. The all ones vector $\mathbf{1}$ is contained in the column span of the feature matrix $\mathbf{\Phi}$. Furthermore, let $\boldsymbol{\varrho} \in \mathbb{R}^d$ such that for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$, $\langle \boldsymbol{\varphi}(x, a), \boldsymbol{\varrho} \rangle = 1$.

Using these insights, it is straightforward to derive a linear program akin Equation (7.1) that characterize the optimal occupancy measure and thus an optimal policy in average-reward MDPs. Starting from this formulation and proceeding as in Sections 7.1 and 7.3, we equivalently restate this optimization problem as finding the saddle-point of the reparametrized Lagrangian defined as follows:

$$\begin{aligned}\mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}; \rho, \mathbf{v}, \boldsymbol{\theta}) = \rho &+ \langle \boldsymbol{\beta}\,, \mathbf{\Lambda} c[\boldsymbol{\theta}_r + \mathbf{\Psi}\mathbf{v} - \boldsymbol{\theta} - \rho\boldsymbol{\varrho}] \rangle \\ &+ \langle \mathbf{p}\,, \mathbf{\Phi}\boldsymbol{\theta} - \mathbf{E}\mathbf{v} \rangle.\end{aligned}$$

As previously, the saddle point can be shown to be equivalent to an optimal occupancy measure under the assumption that the MDP is linear in the sense of Definition 7.1.1. Notice that the above Lagrangian slightly differs from that of the discounted setting in Equation (7.13) due to the additional optimization parameter ρ, but otherwise our main algorithm can be directly generalized to this objective. We present details of the

derivations and the resulting algorithm in Appendix E. The following theorem states the performance guarantees for this method.

Theorem 7.4.3. *Given a linear MDP (Definition 7.1.1) satisfying Assumption 7.4.2 and such that* $\boldsymbol{\theta}^\pi \in \mathbb{B}(D_{\boldsymbol{\theta}})$ *for any policy* π*. Assume that the coverage ratio is bounded* $C_{\boldsymbol{\varphi},c}(\pi^*) \leq D_{\boldsymbol{\beta}}$*. Then, for any comparator policy* π^**, the policy output by an appropriately tuned instance of Algorithm 4 satisfies*

$$\mathbb{E}\left[\langle \mathbf{p}^{\pi^*} - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r}\rangle\right] \leq \varepsilon$$

with a number of samples n_ϵ *that is*

$$O\left(\varepsilon^{-4} D_{\boldsymbol{\theta}}^4 D_{\boldsymbol{\varphi}}^{12c-2} D_{\boldsymbol{\beta}}^4 d^{2-2c} \log|\mathcal{A}|\right).$$

As compared to the discounted case, this additional dependence of the sample complexity on $D_{\boldsymbol{\varphi}}$ is due to the extra optimization variable ρ. We provide the full proof of this theorem along with further discussion in Appendix E.

7.5 Detailed Computations for Comparing Coverage Ratios

In this section, after reviewing the different versions of coverage ratio discussed in the paper, we prove several inequalities that hold between them. For ease of comparison, we only consider discounted linear MDPs (Definition 7.1.1).

Definition 7.5.1. Recall the following definitions of coverage ratio given by different authors in the offline RL literature:

1. $C_{\boldsymbol{\varphi},c}(\pi^*) = \mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\varphi}(X,A)\right]^\top \boldsymbol{\Lambda}_\mu^{-2c} \mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\varphi}(X,A)\right]$ (Ours)
2. $C^\dagger(\pi^*) = \mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\varphi}(X,A)^\mathsf{T} \boldsymbol{\Lambda}_\mu^{-1} \boldsymbol{\varphi}(X,A)\right]$ (e.g., Y. Jin, Z. Yang, and Z. Wang (2021))
3. $C^\diamond(\pi^*) = \sup_{y\in\mathbb{R}^d} \frac{y^\mathsf{T}\mathbb{E}_{X,A\sim p^*}[\boldsymbol{\varphi}(X,A)\boldsymbol{\varphi}(X,A)^\mathsf{T}]y}{y^\mathsf{T}\mathbb{E}_{X,A\sim p^\mu}[\boldsymbol{\varphi}(X,A)\boldsymbol{\varphi}(X,A)^\mathsf{T}]y}$ (e.g., Uehara and Sun (2022))
4. $C_{\mathcal{F},\pi}(\pi^*) = \max_{f\in\mathcal{F}} \frac{\|f-\mathcal{T}^\pi f\|_{p^*}^2}{\|f-\mathcal{T}^\pi f\|_{p^\mu}^2}$ (e.g., Xie, C. Cheng, et al. (2021)),

where $c \in \{1, 2\}$, $\mathbf{\Lambda}_\mu = \mathbb{E}_{X,A\sim p^\mu}[\boldsymbol{\varphi}(X, A)\boldsymbol{\varphi}(X, A)^\intercal]$ (assumed invertible), $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$, and $\mathcal{T}^\pi : \mathcal{F} \to \mathbb{R}$ defined as $(\mathcal{T}^\pi f)(x, a) = r(x, a) + \gamma \sum_{x',a'} p(x' \mid x, a)\pi(a' \mid x')f(x', a')$ is the Bellman operator associated to policy π.

In the following, we construct a problem instance where $C^\dagger$ can be arbitrarily larger than $C_{\varphi,c}$, regardless of the value of c, thanks to the single-direction property of our coverage ratio discussed in Section 8.3.

Proposition 7.5.2. *There exists a linear MDP with two states, two actions and feature dimension $d = 3$, such that, for every $\epsilon \in (0, 1)$, there exists a behavior policy μ, such that $C_{\varphi,c}(\pi^*)$ is bounded by a constant independent of ϵ for all $c \in \{1/2, 1\}$, while $C^\dagger(\pi^*) = \Omega(\epsilon^{-1})$, where π^* is the unique deterministic optimal policy of the MDP.*

Proof. Let $|\mathcal{X}| = \{x_1, x_2\}$ and $\mathcal{A} = \{a_1, a_2\}$. Consider the following 3-dimensional feature map where $\boldsymbol{\varphi}_{ij}$ is short for $\boldsymbol{\varphi}(x_i, a_j)$:

$$\boldsymbol{\varphi}_{11} = [4, 0, 1]^\top, \qquad \boldsymbol{\varphi}_{12} = [1, 1, 1]^\top,$$
$$\boldsymbol{\varphi}_{21} = [0, 4, 1]^\top, \qquad \boldsymbol{\varphi}_{22} = [-1, -1, 1]^\top.$$

Following the notation of Definition 7.1.1, let $\boldsymbol{\psi}(x_1) = \boldsymbol{\psi}(x_2) = [0, 0, 1/2]^\top$ and $\boldsymbol{\theta}_r = [1, 1, 0]^\top$, obtaining $p(x_k|x_i, a_j) = 1/2$ for all $i, j, k \in [2]$, and the following reward function:

$$r(x_1, a_1) = 4, \qquad r(x_1, a_2) = 2,$$
$$r(x_2, a_1) = 4, \qquad r(x_2, a_2) = -2.$$

Finally, let $\nu_0(x_1) = \nu_0(x_2) = 1/2$. It is easy to see that, for any discount factor $\gamma > 0$, the MDP admits a unique deterministic optimal policy, $\pi^*(x_1) = \pi^*(x_2) = a_1$, with optimal value $\rho^* = 4(1 - \gamma)$. The state-action occupancy measure induced by this optimal policy is

$$p^*(x_1, a_1) = p^*(x_2, a_1) = \frac{1}{2}, \qquad p^*(x_1, a_2) = p^*(x_2, a_2) = 0.$$

Now fix an $\epsilon \in (0, 1)$. Let the behavior policy be

$$\mu(a_1|x_1) = \epsilon, \qquad \mu(a_2|x_1) = 1 - \epsilon,$$
$$\mu(a_1|x_2) = \epsilon, \qquad \mu(a_2|x_2) = 1 - \epsilon.$$

The state-action occupancy measure induced by the behavior policy is

$$p^\mu(x_1, a_1) = \frac{\epsilon}{2}, \qquad p^\mu(x_1, a_2) = \frac{1-\epsilon}{2},$$
$$p^\mu(x_2, a_1) = \frac{\epsilon}{2}, \qquad p^\mu(x_2, a_2) = \frac{1-\epsilon}{2}.$$

The feature covariance matrix under μ is then

$$\mathbf{\Lambda}_\mu = \mathbb{E}_{X,A\sim p^\mu}[\boldsymbol{\varphi}(X,A)\boldsymbol{\varphi}(X,A)^\top] = \begin{bmatrix} 1+7\epsilon & 1-\epsilon & 2\epsilon \\ 1-\epsilon & 1+7\epsilon & 2\epsilon \\ 2\epsilon & 2\epsilon & 1 \end{bmatrix},$$

from which we obtain the coverage ratio

$$C^\dagger(\pi^*) = \mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\varphi}(X,A)^\intercal \mathbf{\Lambda}_\mu^{-1} \boldsymbol{\varphi}(X,A)\right] = \frac{1+9\epsilon}{\epsilon(1+4\epsilon)} = \Omega(\epsilon^{-1}). \quad (7.17)$$

To compute $C_{\varphi,c}(\pi^*)$, note that the expected feature vector under π^* is

$$\overline{\boldsymbol{\varphi}}(\pi^*) = \mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\varphi}(X,A)\right] = [2,2,1]^\top.$$

Hence:

$$C_{\varphi,1/2}(\pi^*) = \overline{\boldsymbol{\varphi}}(\pi^*)^\top \mathbf{\Lambda}_\mu^{-1} \overline{\boldsymbol{\varphi}}(\pi^*) = \frac{5}{1+4\epsilon} \leq 5, \quad (7.18)$$

$$C_{\varphi,1}(\pi^*) = \overline{\boldsymbol{\varphi}}(\pi^*)^\top \mathbf{\Lambda}_\mu^{-2} \overline{\boldsymbol{\varphi}}(\pi^*) = \frac{3}{(1+4\epsilon)^2} \leq 3 < 5. \quad (7.19)$$

□

The previous proof admits a simple geometric interpretation: for $\epsilon \to 0$, the span of the features visited by the behavior policy degenerates to $\text{span}(\{\boldsymbol{\varphi}_{12}, \boldsymbol{\varphi}_{22}\})$, which belongs to a 2-dimensional subspace of $\mathbb{R}^3$, while the optimal features span the whole $\mathbb{R}^3$. So, according to the notion of coverage from Y. Jin, Z. Yang, and Z. Wang 2021, the data fail to cover the span of the optimal features. However, the average optimal feature $\overline{\boldsymbol{\varphi}}(\pi^*)$ belongs to the very same subspace covered by the data, which is enough

according to our notion of coverage. In particular, $\overline{\boldsymbol{\varphi}}(\pi^*) = 3/2\boldsymbol{\varphi}_{12} - 1/2\boldsymbol{\varphi}_{22}$.

The following is a generalization of the low-variance property discussed in Section 8.3.

Proposition 7.5.3. *Let $\mathbb{V}[Z] = \mathbb{E}[\|Z - \mathbb{E}[Z]\|^2]$ for a random vector Z. Then, for any pair of policies π^*, μ*

$$C_{\varphi,c}(\pi^*) = \mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\varphi}(X,A)^\intercal \boldsymbol{\Lambda}_\mu^{-2c} \boldsymbol{\varphi}(X,A)\right] - \mathbb{V}_{X,A\sim p^*}\left[\boldsymbol{\Lambda}_\mu^{-c}\boldsymbol{\varphi}(X,A)\right].$$

In particular, $C_{\varphi,1/2}(\pi^*) \leq C^\dagger(\pi^*)$ for all π^*, μ.

Proof. We just rewrite $C_{\varphi,c}$ from Definition 7.5.1 as

$$C_{\varphi,c}(\pi^*) = \left\|\mathbb{E}_{X,A\sim p^*}\left[\boldsymbol{\Lambda}_\mu^{-c}\boldsymbol{\varphi}(X,A)\right]\right\|^2.$$

The result follows from the elementary property of variance $\mathbb{V}[Z] = \mathbb{E}[\|Z\|^2] - \|\mathbb{E}[Z]\|^2$. The second statement follows from the non-negativity of the variance, but can also be obtained directly via Jensen's inequality. □

Proposition 7.5.4. $C^\diamond(\pi^*) \leq C^\dagger(\pi^*) \leq dC^\diamond(\pi^*)$.

Proof. Let $(X^*, A^*) \sim p^*$ and $\boldsymbol{\Lambda}_* = \boldsymbol{\Lambda}_{\pi^*}$. First, we rewrite $C^\dagger$ as

$$\begin{aligned}
C^\dagger(\pi^*) &= \mathbb{E}\left[\boldsymbol{\varphi}(X^*,A^*)^\intercal \boldsymbol{\Lambda}_\mu^{-1}\boldsymbol{\varphi}(X^*,A^*)\right] \\
&= \mathbb{E}\left[\mathrm{Tr}(\boldsymbol{\varphi}(X^*,A^*)^\intercal \boldsymbol{\Lambda}_\mu^{-1}\boldsymbol{\varphi}(X^*,A^*))\right] \\
&= \mathbb{E}\left[\mathrm{Tr}(\boldsymbol{\varphi}(X^*,A^*)\boldsymbol{\varphi}(X^*,A^*)^\intercal \boldsymbol{\Lambda}_\mu^{-1})\right] && (7.20)\\
&= \mathrm{Tr}(\boldsymbol{\Lambda}_*\boldsymbol{\Lambda}_\mu^{-1}) && (7.21)\\
&= \mathrm{Tr}(\boldsymbol{\Lambda}_\mu^{-1/2}\boldsymbol{\Lambda}_*\boldsymbol{\Lambda}_\mu^{-1/2}), && (7.22)
\end{aligned}$$

where we have used the cyclic property of the trace (twice) and linearity of trace and expectation. Note that, since $\boldsymbol{\Lambda}_\mu$ is positive definite, it admits a unique positive definite matrix $\boldsymbol{\Lambda}_\mu^{1/2}$ such that $\boldsymbol{\Lambda}_\mu = \boldsymbol{\Lambda}_\mu^{1/2}\boldsymbol{\Lambda}_\mu^{1/2}$. We rewrite

$C^\diamond$ in a similar fashion

$$\begin{aligned} C^\diamond(\pi^*) &= \sup_{y \in \mathbb{R}^d} \frac{y^\intercal \mathbf{\Lambda}_* y}{y^\intercal \mathbf{\Lambda}_\mu y} \\ &= \sup_{z \in \mathbb{R}^d} \frac{z^\intercal \mathbf{\Lambda}_\mu^{-1/2} \mathbf{\Lambda}_* \mathbf{\Lambda}_\mu^{-1/2} z}{z^\intercal z} \qquad (7.23) \\ &= \lambda_{\max}(\mathbf{\Lambda}_\mu^{-1/2} \mathbf{\Lambda}_* \mathbf{\Lambda}_\mu^{-1/2}), \qquad (7.24) \end{aligned}$$

where $\lambda_{\max}$ denotes the maximum eigenvalue of a matrix. We have used the fact that both $\mathbf{\Lambda}_*$ and $\mathbf{\Lambda}_\mu$ are positive definite and the min-max theorem. Since the quadratic form $\mathbf{\Lambda}_\mu^{-1/2} \mathbf{\Lambda}_* \mathbf{\Lambda}_\mu^{-1/2}$ is also positive definite, and the trace is the sum of the (positive) eigenvalues, we get the desired result. □

Proposition 7.5.5 (cf. the proof of Theorem 3.2 from (Xie, C. Cheng, et al. 2021))**.** *Let* $\mathcal{F} = \{f_{\boldsymbol{\theta}} : (x, a) \mapsto \langle \boldsymbol{\varphi}(x, a), \boldsymbol{\theta} \rangle \mid \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d\}$ *where* φ *is the feature map of the linear MDP. Then*

$$C_{\mathcal{F},\pi}(\pi^*) \leq C^\diamond(\pi^*),$$

with equality if $\Theta = \mathbb{R}^d$*.*

Proof. Fix any policy π and let $\mathcal{T} = \mathcal{T}^\pi$. By linear Bellman completeness of linear MDPs (C. Jin et al. 2020), $\mathcal{T} f \in \mathcal{F}$ for any $f \in \mathcal{F}$. For $f_{\boldsymbol{\theta}} : (x, a) \mapsto \langle \boldsymbol{\varphi}(x, a), \boldsymbol{\theta} \rangle$, let $\mathcal{T}\boldsymbol{\theta} \in \Theta$ be defined so that $\mathcal{T} f_{\boldsymbol{\theta}} : (x, a) \mapsto \langle \boldsymbol{\varphi}(x, a), \mathcal{T}\boldsymbol{\theta} \rangle$. Then

$$\begin{aligned} C_{\mathcal{F},\pi}(\pi^*) &= \max_{f \in \mathcal{F}} \frac{\mathbb{E}_{X,A \sim p^*}\left[\left(f(X, A) - \mathcal{T} f(X, A)\right)^2\right]}{\mathbb{E}_{X,A \sim p^\mu}\left[\left(f(X, A) - \mathcal{T} f(X, A)\right)^2\right]} \qquad (7.25) \\ &\leq \max_{\boldsymbol{\theta} \in \mathbb{R}^d} \frac{\mathbb{E}_{X,A \sim p^*}\left[\langle \boldsymbol{\varphi}(X, A), \boldsymbol{\theta} - \mathcal{T}\boldsymbol{\theta} \rangle^2\right]}{\mathbb{E}_{X,A \sim p^\mu}\left[\langle \boldsymbol{\varphi}(X, A), \boldsymbol{\theta} - \mathcal{T}\boldsymbol{\theta} \rangle^2\right]} \qquad (7.26) \\ &= \max_{y \in \mathbb{R}^d} \frac{\mathbb{E}_{X,A \sim p^*}\left[\langle \boldsymbol{\varphi}(X, A), y \rangle^2\right]}{\mathbb{E}_{X,A \sim p^\mu}\left[\langle \boldsymbol{\varphi}(X, A), y \rangle^2\right]} \qquad (7.27) \\ &= \max_{y \in \mathbb{R}^d} \frac{y^\intercal \mathbb{E}_{X,A \sim p^*}\left[\boldsymbol{\varphi}(X, A) \boldsymbol{\varphi}(X, A)^\intercal\right] y}{y^\intercal \mathbb{E}_{X,A \sim p^\mu}\left[\boldsymbol{\varphi}(X, A) \boldsymbol{\varphi}(X, A)^\intercal\right] y}, \qquad (7.28) \end{aligned}$$

where the inequality in Equation (7.26) holds with equality if $\Theta = \mathbb{R}^d$. □

Chapter 8

Conclusions

8.1 Online Learning with Off-Policy Feedback

We introduced a new online learning setting where the learner is only allowed to observe off-policy feedback generated by a fixed behavior policy. We have proposed an algorithm with comparator-dependent regret bounds of order $C^\dagger(\pi^*)\sqrt{n}$, depending on a naturally defined coverage ratio parameter $C^\dagger(\pi^*)$ that characterizes the mismatch between the behavior and the comparator policies. Many questions remain open regarding the potential tightness of this result. First, we have shown that the bounds can be improved to $O(\sqrt{C^\dagger(\pi^*)n})$, if one wishes to restrict their attention to comparators whose coverage level is at a fixed level $C^\dagger(\pi^*)$. However, the tuning required for achieving this result depends on the desired coverage level. It is an interesting open problem to find out if this requirement can be relaxed, and bounds of order $\sqrt{C^\dagger(\pi^*)n}$ can be simultaneously achieved for all comparators π^* by a single algorithm. We conjecture that this question can be addressed by a careful adaptation of existing techniques for adaptive online learning, and in particular we believe that adapting the methodology of Koolen and Erven (2015) should be especially suitable for achieving this goal.

Questions regarding the best achievable performances for our newly defined problem are even more exciting. As an adaptation of the results

of Xiao et al. (2021) show via an online-to-batch reduction, the minimax regret of any algorithm for this setting has to scale as $\sqrt{n/\min_a \mu(a)}$, suggesting that our naïve adaptation of Exp3 is already minimax optimal. In our view, this makes it all the more interesting to identify characteristics of individual problem instances that make faster learning possible, and we believe that comparator-dependent regret bounds scaling with the coverage ratio are only one of many possible flavors of adaptive performance guarantees. One concrete question that we are particularly interested in is a better understanding of the "Pareto regret frontier" of achievable regrets, roughly corresponding to the set of comparator-dependent regret bounds that are achievable by any algorithm. Clearly, the bounds we achieve are just singular elements of this set. We conjecture that bounds of order $\sqrt{C^\dagger(\pi^*)n}$ are indeed on the regret frontier. Whether this is indeed true or if there are other distinguished entries on the Pareto frontier with desirable properties remains to be seen. All in all, our results highlight that off-policy learning is a field of study that's ripe with open questions that can be interesting for the online learning community that is typically very keen on instance-dependent analysis.

A more ambitious question for future research is if our techniques can be extended to more challenging settings, and especially online learning in Markov decision processes (Even-Dar, Kakade, and Mansour 2009; Neu, György, and Szepesvári 2010; Neu, György, Szepesvári, and Antos 2014). We think that an extension to this setting would be particularly valuable, given the recent flurry of interest in offline reinforcement learning. In this context, we could potentially exploit the unique feature of our algorithm design that, unlike all other methods, it does not rely on explicit uncertainty quantification for calculating its pessimistic updates. This could mean a major advantage over traditional off-policy RL methods that rely on uncertainty quantification to build confidence sets over abstract objects (like the entire transition function of the Markov process), which is a notoriously hard problem, especially in the infinite-horizon setting. In contrast, as our results in Section 5.4 highlight, the pessimistic nature of our method is realized through an update rule that is slightly more conservative than the standard exponential-weights update rule. We believe that this insight can be very useful for developing new methods for offline RL, even more so since they appear to be directly compatible with the

primal-dual off-policy learning methods of Nachum, Chow, et al. (2019); Nachum, Dai, et al. (2019); Uehara, Huang, and Jiang (2020).

8.2 Importance-Weighted Offline Learning

We now provide some additional discussion on our results, related work, and open problems.

No more uniform coverage. The bounds we have proved are tighter than any that are known in the literature, and they have the particular strength that they do not require the action probabilities to be strictly bounded away from zero. Virtually all previous bounds require this "uniform-coverage" assumption, largely due to their excessive reliance on textbook concentration results like Bernstein's, Bennett's, or Freedman's inequalities. The only result we are aware of that does not explicitly suffer from this limitation is by Y. Jin, Ren, et al. (2022), who rely on a very sophisticated new proof technique which eventually does not yield easily interpretable performance bounds due to the appearance of some higher moments of the importance weights. The key to our stronger results is the observation that the tails of importance-weighted reward estimators are *asymmetric*: their lower tails are always light, and thus one only has to tame the upper tails via pessimistic adjustments. This simple observation allows us to derive very tight bounds using a few lines of elementary derivations. If there is any moral to this story, then it is that one should always avoid using two-sided concentration inequalities for importance-weighted estimators (at least as long as the rewards are positive).

Implicit exploration and clipped importance weighting. A perhaps more traditional way to control the tails of importance-weighted estimators is the clipped importance weighting (CIW) estimator we have defined in Section 6.6. Variants of this estimator have been studied at least since the work of Ionides (2008) and vigorously applied in the offline learning literature (Bottou et al. 2013; Flynn et al. 2023; Sakhi, Alquier, and Chopin 2023). Interestingly, despite its broad usage, we are not aware of any work in this context that has worked out expressions for the bias of the CIW estimator, much less derived a regret bound for the resulting

offline learning scheme. We believe that our results for the closely related IX estimator should essentially all apply to the CIW estimator and indeed our experiments show that they behave nearly identically in the settings we have tested. Nevertheless, we suspect that analyzing this estimator would end up being considerably more involved than our own analysis, but of course we would love to be proved wrong by future work.

Reward-scaled coverage ratios. A subtle improvement of our bounds as opposed to the ones of L. Wang, Krishnamurthy, and Slivkins (2023) is that they depend on the *reward-scaled* version of the coverage ratio. This implies that bounds expressed in terms the scaled ratio $C_\gamma(\pi^*)$ can be much tighter than ones expressed in terms of $C(\pi^*)$ when the rewards of the comparator policy π^* "tend to be small" in an appropriate sense. Note that this is a significant improvement in practical applications like online recommendation systems, where expected rewards correspond to clickthrough rates, which are very close to zero even for the very best ad campaigns. In the special case where rewards are negatively correlated with the importance weights (which may intuitively happen if the behavior policy is "reasonably good" in the sense that it puts larger weights on good actions), the coverage ratio against the optimal *deterministic* policy π^* can be shown to satisfy $C_\gamma(\pi^*) \leq \rho(\pi^*)C(\pi^*)$, thus improving greatly over standard bounds that depend on $C(\pi^*)$. Bounds that improve for small expected rewards are known in the bandit literature at least since the work of Auer et al. (2002), and we are curious if guarantees like the above can be proved under more general conditions for offline learning as well.

Lower bounds. The "optimality" of pessimistic offline learning methods is a contentious topic that we prefer not to discuss here in much detail. In particular, even in the simplest case of offline learning in multi-armed bandits, Xiao et al. (2021) have shown that a large range of algorithms including pessimistic, greedy, and optimistic methods satisfies the standard notion of minimax optimality, and there is thus nothing special about pessimistic methods in these terms. Putting this alarming concern aside, pessimistic algorithms tend to have the property that their regret scales with the minimax sample complexity of *estimating* the value of the comparator policy (Y. Jin, Z. Yang, and Z. Wang 2021; Xiao et al. 2021). In

our case, it is not entirely clear if this statement continues to be true. In the special case of multi-armed bandits with binary rewards and a deterministic comparator policy, our bound matches the lower bound proved by L. Li, Munos, and Szepesvári (2015) (up to a $\log K$ factor). That said, already in the case of stochastic comparator policies, our upper bounds no longer match the minimax sample complexity of estimation. Finding out if better algorithms with matching regret guarantees can be developed is a very interesting research question that we leave open for now.

Computational-statistical tradeoffs. As we show in this paper, it is possible to develop oracle-efficient algorithms with good statistical guarantees. However, these algorithms don't seem to demonstrate the correct scaling with the problem complexity unless prior knowledge of problem parameters is provided to the algorithm. This limitation can be bypassed by a more involved algorithm we describe in Appendix 6.5, but the resulting method cannot apparently be implemented via a single call to the optimization oracle. Whether or not this computational-statistical tradeoff is inherent to the problem is unclear at this point and warrants further research.

Further refinements. Our results can be extended in a number of straightforward ways by building on previous developments in the literature. For instance, the dependence on $\log|\Pi|$ appearing in our main results can be most likely replaced by other complexity measures like covering numbers or the Natarajan dimension of the policy class, by adapting the techniques of either Swaminathan and Joachims (2015) or Y. Jin, Ren, et al. (2022). Similar bounds can be recovered by our PAC-Bayesian guarantees presented in Section 6.4 by building on techniques of Audibert (2004); Catoni (2007) (see also (Grünwald, Steinke, and Zakynthinou 2021)). Another very simple generalization that our framework can readily handle is the case of adaptive behavior policies, where each sample point (X_t, A_t, R_t) can be generated by a different behavior policy μ_t that may potentially depend on all past observations. The concentration bounds of Lemmas 6.3.2 and 6.3.3 can be very easily adapted to deal with such observations, and accordingly a version of our main result can be proved with the quantity $\mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n}\sum_{a}\frac{\pi^*(a|X_t)}{\mu_t(a|X_t)+\gamma}\cdot g(X_t,a)\right]$ taking the role of $C_\gamma(\pi^*)$.

We hope that the simplicity of our techniques will enable further progress on the topic of importance-weighted offline learning, and in particular that further interesting extensions will be uncovered by future work.

8.3 Offline Learning in Markov Decision Processes

In this section, we compare our results with the most relevant ones from the literature, with a particular focus on discussing the relations between the coverage ratios used in our work and the ones used in related literature. Our Table 7.1 can be used as a reference. As a complement to this section, we refer the interested reader to the recent work by Uehara and Sun (2022), which provides a survey of offline RL methods with their coverage and structural assumptions. Detailed computations can be found in Appendix 7.5.

An important property of our method is that it only requires partial coverage. This sets it apart from classic batch RL methods like fitted Q-iteration (Ernst, Geurts, and Wehenkel 2005; Munos and Szepesvári 2008; J. Chen and Jiang 2019), whose analysis requires a stronger uniform-coverage assumption. Interestingly, our results defy the common wisdom in the related literature that suggests that obtaining guarantees under weaker partial-coverage assumptions requires the use of pessimistic adjustments (e.g., (Y. Jin, Z. Yang, and Z. Wang 2021; Xie, C. Cheng, et al. 2021))—indeed, notice that our algorithm does not implement any form of explicit pessimism. In fact, as we argue below, the notion of coverage that our guarantees depend on is in many senses much weaker than the most commonly used notions appearing in the literature.

Let us review some existing notions of coverage and contrast them to our notion. Y. Jin, Z. Yang, and Z. Wang (2021) (Theorem 4.4) rely on a *feature* coverage ratio which can be written as

$$C^\dagger(\pi^*) = \mathbb{E}_{X,A\sim\mu^*}\left[\boldsymbol{\varphi}(X,A)^\intercal \boldsymbol{\Lambda}_\mu^{-1}\boldsymbol{\varphi}(X,A)\right]. \tag{8.1}$$

By Jensen's inequality, our $C_{\varphi,1/2}$ (Definition 4.7.1) is never larger than $C^\dagger$, and more precisely one can show that

$$C_{\varphi,1/2}(\pi^*) = C^\dagger(\pi^*) - \mathbb{V}_{X,A\sim\mu^*}\left[\boldsymbol{\Lambda}_\mu^{-1/2}\boldsymbol{\varphi}(X,A)\right],$$

where $\mathbb{V}[Z] = \mathbb{E}[\|Z - \mathbb{E}[Z]\|^2]$ for a random vector Z. Furthermore, a simple geometric argument shows that the difference can be very large. A boundedness condition on $C^\dagger(\pi^*)$ requires the column space of $\mathbf{\Lambda}_\mu$ to span the subspace of $\mathbb{R}^d$ spanned by optimal features, $\text{span}\{\boldsymbol{\varphi}(x,a) \mid \mu^{\pi^*}(x,a) > 0\}$. In contrast, our coverage assumption only requires $\overline{\boldsymbol{\varphi}}(\pi^*) \in \text{range}(\mathbf{\Lambda}_\mu)$. Intuitively, we only require the behavior policy to witness a single direction in feature space (the average feature vector under π^*) compared to a whole, potentially d-dimensional, subspace. This can make a big difference, especially when d is large. Although it is not true in general that $C_{\varphi,1}$ is smaller than $C^\dagger(\pi^*)$, the single-direction property continues to hold for $c = 1$. In Appendix 7.5, we show an example where $C^\dagger$ can be arbitrarily larger than both $C_{\varphi,1/2}$ and $C_{\varphi,1}$.

This kind of coverage ratio has appeared in the literature before, but only for finite-horizon problems. Concretely, Zanette, Wainwright, and Brunskill (2021) propose a computationally intense algorithm that demonstrates a regret bound scaling with a quantity essentially equivalent to our $C_{\varphi,1/2}$. Uehara and Sun (2022) and X. Zhang et al. (2022) use a coverage ratio that is conceptually similar to Equation (8.1),

$$C^\diamond(\pi^*) = \sup_{y \in \mathbb{R}^d} \frac{y^\intercal \mathbf{\Lambda}^*_\mu y}{y^\intercal \mathbf{\Lambda}_\mu y}, \tag{8.2}$$

where $\mathbf{\Lambda}^*_\mu = \mathbb{E}_{X,A \sim \mu^*}[\boldsymbol{\varphi}(X,A)\boldsymbol{\varphi}(X,A)^\intercal]$. Some linear algebra shows that $C^\diamond \leq C^\dagger \leq dC^\diamond$. Therefore, chaining the previous inequalities we know that $C_{\varphi,1/2} \leq C^\dagger \leq dC^\diamond$. It should be noted that the algorithm from Uehara and Sun (2022) also works with unknown features, at the cost of being computationally inefficient. The algorithm from X. Zhang et al. (2022) instead is limited to the finite-horizon setting.

We can gain some further insight from the special case of tabular MDPs, although it is hard to compare our ratio with existing ones there, because in this setting, error bounds are commonly stated in terms of $\sup_{x,a} {}^{\mu^*(x,a)}/_{\mu_B(x,a)}$, often introducing an explicit dependency on the number of states (e.g., Liu et al. 2020). However, looking at how the coverage ratio specializes to the tabular setting can still provide some insight. First, $C_{\varphi,1/2}(\pi^*) = \sum_{x,a} {}^{(\mu^*(x,a))^2}/_{\mu_B(x,a)}$, which of course is smaller than the more standard $C^\dagger(\pi^*) = \sum_{x,a} {}^{\mu^*(x,a)}/_{\mu_B(x,a)}$. Interestingly, $C_{\varphi,1/2}(\pi^*) = 1 + \mathcal{X}^2(\mu^* \| \mu_B)$,

where $\mathcal{X}^2$ denotes the chi-square divergence, a crucial quantity in off-distribution learning based on importance sampling Cortes, Mansour, and Mohri 2010. An analogous quantity was used by L. Li, Munos, and Szepesvári (2014) to characterize the sample complexity of off-policy policy evaluation. Unfortunately, $C_{\varphi,1}(\pi^*) = \sum_{x,a}(\mu^*(x,a)/\mu_B(x,a))^2$ is non-comparable with $C^\dagger$ in general, and larger than $C_{\varphi,1/2}$. A similar quantity to $C_{\varphi,1}$ was used by Lykouris et al. (2021) in the context of RL with adversarial corruptions.

The most directly comparable works to ours are those of Xie, C. Cheng, et al. (2021) and C.-A. Cheng et al. (2022), which are the only known practical methods to consider function approximation in the infinite-horizon setting, with minimal assumptions on the dataset. They both use the coverage ratio $C_{\mathcal{F}}(\pi^*) = \max_{f\in\mathcal{F}} \|f-\mathcal{T}f\|^2_{\mu^*}/\|f-\mathcal{T}f\|^2_{\mu_B}$, where $\mathcal{F}$ is a function class and $\mathcal{T}$ the Bellman operator. This can be shown to reduce to Equation (8.2) for linear MDPs (cf. Appendix 7.5). However, the specialized bound of Xie, C. Cheng, et al. (2021) (Theorem 3.2) scales with the potentially larger ratio from Equation (8.1). Both their algorithms have superlinear computational complexity and a sample complexity of $O(\varepsilon^{-5})$. While the authors make plausible arguments in their paper that their method can be efficiently implemented in the linear setting and may obtain a sample complexity of order of order ε^{-2}, these statements are not supported with rigorous proofs. Hence, our result is technically the first *provably* computationally effective method that achieves a rate better than $O(\varepsilon^{-5})$, with the additional benefit of using a single-direction coverage ratio as discussed in the above paragraphs.

The above discussion outlines two major open problems that we leave open for future work. First, we highlight that so far, no computationally efficient algorithm exists for our setting that achieves the minimax optimal sample complexity rate of $O(\varepsilon^{-2})$ (Xiao et al. 2021; Rashidinejad, B. Zhu, et al. 2022). Regarding our own algorithm, it is clear that the extra $O(\varepsilon^{-2})$ factor in our bounds is due to the nested-loop structure of the algorithm. How to remove this component from our algorithm design is currently unclear, but we suspect that that borrowing ideas from the literature on optimistic descent methods (Korpelevich 1976; Rakhlin and Sridharan 2013) or two-timescale stochastic approximation (Borkar 1997) may bring

us closer to an answer. A second limitation of our contribution is that, in order to scale with $C_{\varphi,1/2}$, our method requires prior knowledge of $\mathbf{\Lambda}_\mu$. We believe that this limitation can be relaxed at the price of a significantly more involved analysis, for instance by setting aside some fraction of the data set to estimate $\mathbf{\Lambda}_\mu$ (or directly $\mathbf{\Lambda}_\mu^{-1}$, using techniques from (Neu and Olkhovskaya 2020; Neu and Olkhovskaya 2021)). We opted to focus on this slightly stylized scenario to maintain the clarity of our technical contribution. That said, as long as one is happy with a bound that scales with $C_{\varphi,1}$, a simple and elegant version of our algorithm can provide such bounds without prior knowledge of Λ. Whether or not a it is possible to unify the advantages of the two versions of our algorithm is an exciting question for future research.

Bibliography

[1] Daniel G Horvitz and Donovan J Thompson. "A generalization of sampling without replacement from a finite universe". In: *Journal of the American statistical Association* 47.260 (1952), pp. 663–685.

[2] Richard Bellman. *Dynamic programming*. Tech. rep. RAND CORP SANTA MONICA CA, 1956.

[3] Alan S Manne. "Linear programming and sequential decisions". In: *Management Science* 6.3 (1960), pp. 259–267.

[4] Alan S. Manne. "Linear Programming and Sequential Decisions". In: *Manage. Sci.* 6.3 (Apr. 1960), pp. 259–267. ISSN: 0025-1909. DOI: 10.1287/mnsc.6.3.259. URL: https://doi.org/10.1287/mnsc.6.3.259.

[5] GM Korpelevich. "The extragradient method for finding saddle points and other problems". In: *Matecon* 12 (1976), pp. 747–756.

[6] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982. ISBN: 978-0-12-093480-5.

[7] A. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley Interscience, 1983.

[8] Vladimir G. Vovk. "Aggregating Strategies". In: *COLT*. Morgan Kaufmann, 1990, pp. 371–386.

[9] Richard Weber. "On the Gittins Index for Multiarmed Bandits". In: *The Annals of Applied Probability* 2.4 (Nov. 1992). DOI: 10.1214/aoap/1177005588.

[10] Nick Littlestone and Manfred K. Warmuth. "The Weighted Majority Algorithm". In: *Inf. Comput.* 108.2 (1994), pp. 212–261.

[11] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. USA: John Wiley & Sons, Inc., 1994. ISBN: 0471619779.

[12] S.P. Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, 1996.

[13] Vivek S Borkar. "Stochastic approximation with two time scales". In: *Systems & Control Letters* 29.5 (1997), pp. 291–294.

[14] Yoav Freund and Robert E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". In: *J. Comput. Syst. Sci.* 55.1 (1997), pp. 119–139.

[15] David A. McAllester. "Some PAC-Bayesian Theorems". In: *COLT*. ACM, 1998, pp. 230–234.

[16] Aldo Rustichini. "Minimizing regret: The general case". In: *Games and Economic Behavior* 29.1-2 (1999), pp. 224–243.

[17] Daniel A. Spielman and Shang-Hua Teng. "Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time". In: *STOC*. ACM, 2001, pp. 296–305.

[18] Peter Auer et al. "The Nonstochastic Multiarmed Bandit Problem". In: *SIAM J. Comput.* 32.1 (2002), pp. 48–77.

[19] Susan A Murphy. "Optimal dynamic treatment regimes". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 65.2 (2003), pp. 331–355.

[20] Martin Zinkevich. "Online Convex Programming and Generalized Infinitesimal Gradient Ascent". In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*. 2003.

[21] Jean-Yves Audibert. "PAC-Bayesian statistical learning theory". PhD thesis. Université Paris VI, 2004.

[22] Damien Ernst, Pierre Geurts, and Louis Wehenkel. "Tree-Based Batch Mode Reinforcement Learning". In: *J. Mach. Learn. Res.* 6 (2005), pp. 503–556.

[23] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[24] Olivier Catoni. "PAC-Bayesian Supervised Classification". In: *Lecture Notes-Monograph Series. IMS* 1277 (2007).

[25] Nicolò Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. “Improved second-order bounds for prediction with expert advice”. In: *Mach. Learn.* 66.2-3 (2007), pp. 321–352.

[26] John Langford and Tong Zhang. “The Epoch-Greedy Algorithm for Multi-armed Bandits with Side Information”. In: *NeurIPS*. Curran Associates, Inc., 2007, pp. 817–824.

[27] András Antos, Csaba Szepesvári, and Rémi Munos. “Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path”. In: *Mach. Learn.* 71.1 (2008), pp. 89–129.

[28] Edward L Ionides. “Truncated importance sampling”. In: *Journal of Computational and Graphical Statistics* 17.2 (2008), pp. 295–311.

[29] Rémi Munos and Csaba Szepesvári. “Finite-Time Bounds for Fitted Value Iteration”. In: *J. Mach. Learn. Res.* 9 (2008), pp. 815–857.

[30] Kamalika Chaudhuri, Yoav Freund, and Daniel J. Hsu. “A Parameter-free Hedging Algorithm”. In: *NeurIPS*. Curran Associates, Inc., 2009, pp. 297–305.

[31] Eyal Even-Dar, Sham M. Kakade, and Yishay Mansour. “Online Markov Decision Processes”. In: *Math. Oper. Res.* 34.3 (2009), pp. 726–736.

[32] Prashant G. Mehta and Sean P. Meyn. “Q-learning and Pontryagin’s Minimum Principle”. In: *CDC*. IEEE, 2009, pp. 3598–3605.

[33] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. “Learning Bounds for Importance Weighting”. In: *NeurIPS*. Curran Associates, Inc., 2010, pp. 442–450.

[34] Gergely Neu, András György, and Csaba Szepesvári. “The Online Loop-free Stochastic Shortest-Path Problem”. In: *COLT*. Omnipress, 2010, pp. 231–243.

[35] Miroslav Dudík et al. “Efficient Optimal Learning for Contextual Bandits”. In: *UAI*. AUAI Press, 2011, pp. 169–178.

[36] Edward S. Kim et al. In: *Cancer discovery* 1.1 (2011), pp. 44–53.

[37] Lihong Li, Wei Chu, et al. “Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms”. In: *WSDM*. ACM, 2011, pp. 297–306.

[38] Shie Mannor and Ohad Shamir. “From Bandits to Experts: On the Value of Side-Observations”. In: *NeurIPS*. 2011, pp. 684–692.

[39] Baqun Zhang et al. “Estimating optimal treatment regimes from a classification perspective”. In: *Stat* 1.1 (2012), pp. 103–114.

[40] Yingqi Zhao et al. “Estimating individualized treatment rules using outcome weighted learning”. In: *Journal of the American Statistical Association* 107.499 (2012), pp. 1106–1118.

[41] Léon Bottou et al. “Counterfactual reasoning and learning systems: the example of computational advertising”. In: *J. Mach. Learn. Res.* 14.1 (2013), pp. 3207–3260.

[42] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities - A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

[43] Wouter M. Koolen. “The Pareto Regret Frontier”. In: *NeurIPS*. 2013, pp. 863–871.

[44] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, et al. “Playing Atari with Deep Reinforcement Learning”. In: *CoRR* abs/1312.5602 (2013). arXiv: 1312.5602. URL: http://arxiv.org/abs/1312.5602.

[45] Alexander Rakhlin and Karthik Sridharan. “Optimization, learning, and games with predictable sequences”. In: *Advances in Neural Information Processing Systems*. 2013, pp. 3066–3074.

[46] Alekh Agarwal et al. “Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits”. In: *ICML*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 1638–1646.

[47] Gábor Bartók et al. “Partial Monitoring - Classification, Regret Bounds, and Algorithms”. In: *Math. Oper. Res.* 39.4 (2014), pp. 967–997.

[48] Pierre Gaillard, Gilles Stoltz, and Tim van Erven. “A second-order bound with excess losses”. In: *COLT*. Vol. 35. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 176–196.

[49] Tomáš Kocák, Gergely Neu, Michal Valko, and Rémi Munos. “Efficient learning by implicit exploration in bandit problems with side observations”. In: *NeurIPS*. 2014, pp. 613–621.

[50] Lihong Li, Rémi Munos, and Csaba Szepesvári. “On Minimax Optimal Offline Policy Evaluation”. In: *CoRR* abs/1409.3653 (2014).

[51] Gergely Neu, András György, Csaba Szepesvári, and András Antos. "Online Markov Decision Processes Under Bandit Feedback". In: *IEEE Trans. Autom. Control.* 59.3 (2014), pp. 676–691.

[52] Amir Sani, Gergely Neu, and Alessandro Lazaric. "Exploiting easy data in online optimization". In: *NeurIPS*. 2014, pp. 810–818.

[53] Noga Alon, Nicolò Cesa-Bianchi, Ofer Dekel, et al. "Online Learning with Feedback Graphs: Beyond Bandits". In: *COLT*. Vol. 40. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 23–35.

[54] Wouter M. Koolen and Tim van Erven. "Second-order Quantile Methods for Experts and Combinatorial Games". In: *COLT*. Vol. 40. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1155–1175.

[55] Tor Lattimore. "The Pareto Regret Frontier for Bandits". In: *NeurIPS*. 2015, pp. 208–216.

[56] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: https://doi.org/10.1038/nature14539.

[57] Lihong Li, Rémi Munos, and Csaba Szepesvári. "Toward Minimax Off-policy Value Estimation". In: *AISTATS*. Vol. 38. JMLR Workshop and Conference Proceedings. JMLR.org, 2015.

[58] Haipeng Luo and Robert E. Schapire. "Achieving All with No Parameters: AdaNormalHedge". In: *COLT*. Vol. 40. JMLR Workshop and Conference Proceedings. JMLR.org, 2015, pp. 1286–1304.

[59] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 1476-4687. DOI: 10.1038/nature14236. URL: https://doi.org/10.1038/nature14236.

[60] Gergely Neu. "Explore no more: Improved high-probability regret bounds for non-stochastic bandits". In: *NeurIPS*. 2015, pp. 3168–3176.

[61] Adith Swaminathan and Thorsten Joachims. "Batch learning from logged bandit feedback through counterfactual risk minimization". In: *J. Mach. Learn. Res.* 16 (2015), pp. 1731–1755.

[62] Ying-Qi Zhao et al. “Doubly robust learning for estimating individualized treatment with censored data”. In: *Biometrika* 102.1 (2015), pp. 151–168.

[63] Tomáš Kocák, Gergely Neu, and Michal Valko. “Online Learning with Noisy Side Observations”. In: *AISTATS*. Vol. 51. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1186–1194.

[64] Francesco Orabona and Dávid Pál. “Coin Betting and Parameter-Free Online Learning”. In: *NeurIPS*. 2016, pp. 577–585.

[65] Tobias Schnabel et al. “Recommendations as Treatments: Debiasing Learning and Evaluation”. In: *ICML*. Vol. 48. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 1670–1679.

[66] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. ISSN: 1476-4687. DOI: 10.1038/nature16961. URL: https://doi.org/10.1038/nature16961.

[67] Mengdi Wang and Yichen Chen. “An online primal-dual method for discounted Markov decision processes”. In: *CDC*. IEEE, 2016, pp. 4516–4521.

[68] Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, et al. “Nonstochastic Multi-Armed Bandits with Graph-Structured Feedback”. In: *SIAM J. Comput.* 46.6 (2017), pp. 1785–1826.

[69] Dimitris Bertsimas et al. “Personalized diabetes management using electronic medical records”. In: *Diabetes care* 40.2 (2017), pp. 210–217.

[70] Gergely Neu, Anders Jonsson, and Vicenç Gómez. “A unified view of entropy-regularized Markov decision processes”. In: *arXiv preprint arXiv:1705.07798* (2017).

[71] James M Rehg, Susan A Murphy, and Santosh Kumar. “Mobile health”. In: Springer, 2017.

[72] Xin Zhou et al. “Residual weighted learning for estimating individualized treatment rules”. In: *Journal of the American Statistical Association* 112.517 (2017), pp. 169–187.

[73] Yichen Chen, Lihong Li, and Mengdi Wang. “Scalable Bilinear Learning Using State and Action Features”. In: *ICML*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 833–842.

[74] Ashok Cutkosky and Francesco Orabona. “Black-Box Reductions for Parameter-free Online Learning in Banach Spaces”. In: *COLT*. Vol. 75. Proceedings of Machine Learning Research. PMLR, 2018, pp. 1493–1529.

[75] Nathan Kallus. “Balanced Policy Evaluation and Learning”. In: *NeurIPS*. 2018, pp. 8909–8920.

[76] Toru Kitagawa and Aleksey Tetenov. “Who should be treated? empirical welfare maximization methods for treatment choice”. In: *Econometrica* 86.2 (2018), pp. 591–616.

[77] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[78] Jinglin Chen and Nan Jiang. “Information-Theoretic Considerations in Batch Reinforcement Learning”. In: *ICML*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1042–1051.

[79] Vivek F. Farias and Andrew A. Li. “Learning Preferences with Side Information”. In: *Manag. Sci.* 65.7 (2019), pp. 3131–3149.

[80] Akshay Krishnamurthy et al. “Contextual bandits with continuous actions: Smoothing, zooming, and adapting”. In: *COLT*. Vol. 99. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2025–2027.

[81] Tor Lattimore and Csaba Szepesvári. “Cleaning up the neighborhood: A full classification for adversarial partial monitoring”. In: *ALT*. Vol. 98. Proceedings of Machine Learning Research. PMLR, 2019, pp. 529–556.

[82] Ben London and Ted Sandler. “Bayesian Counterfactual Risk Minimization”. In: *ICML*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 4125–4133.

[83] Ofir Nachum, Yinlam Chow, et al. “DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections”. In: (2019), pp. 2315–2325.

[84] Ofir Nachum, Bo Dai, et al. “AlgaeDICE: Policy Gradient from Arbitrary Experience”. In: *CoRR* abs/1912.02074 (2019).

[85] Francesco Orabona. “A modern introduction to online learning”. In: *arXiv preprint arXiv:1912.13213* (2019).

[86] Aleksandrs Slivkins. “Introduction to Multi-Armed Bandits”. In: *Foundations and Trends® in Machine Learning* 12.1-2 (2019), pp. 1–286. ISSN: 1935-8237. DOI: 10.1561/2200000068. URL: http://dx.doi.org/10.1561/2200000068.

[87] Lin Yang and Mengdi Wang. “Sample-Optimal Parametric Q-Learning Using Linearly Additive Features”. In: *ICML*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6995–7004.

[88] Joan Bas-Serrano and Gergely Neu. “Faster saddle-point optimization for solving large-scale Markov decision processes”. In: *L4DC*. Vol. 120. Proceedings of Machine Learning Research. PMLR, 2020, pp. 413–423.

[89] Chi Jin et al. “Provably efficient reinforcement learning with linear function approximation”. In: *COLT*. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2137–2143.

[90] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. DOI: 10.1017/9781108571401.

[91] Sergey Levine et al. “Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems”. In: *CoRR* abs/2005.01643 (2020).

[92] Sergey Levine et al. “Offline reinforcement learning: Tutorial, review, and perspectives on open problems”. In: *arXiv preprint arXiv:2005.01643* (2020).

[93] Yao Liu et al. “Provably Good Batch Off-Policy Reinforcement Learning Without Great Exploration”. In: *NeurIPS*. 2020.

[94] Ofir Nachum and Bo Dai. “Reinforcement Learning via Fenchel-Rockafellar Duality”. In: (Jan. 7, 2020). arXiv: 2001.01866 [cs.LG].

[95] Gergely Neu and Julia Olkhovskaya. “Efficient and robust algorithms for adversarial linear contextual bandits”. In: *COLT*. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3049–3068.

[96] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020. ISBN: 9780134610993. URL: http://aima.cs.berkeley.edu/.

[97] Masatoshi Uehara, Jiawei Huang, and Nan Jiang. “Minimax Weight and Q-Function Learning for Off-Policy Evaluation”. In: *ICML*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Ma-

chine Learning Research. PMLR, 13–18 Jul 2020, pp. 9659–9668. URL: https://proceedings.mlr.press/v119/uehara20a.html.

[98] Pierre Alquier. "User-friendly introduction to PAC-Bayes bounds". In: *arXiv preprint arXiv:2110.11216* (2021).

[99] Susan Athey and Stefan Wager. "Policy learning with observational data". In: *Econometrica* 89.1 (2021), pp. 133–161.

[100] Joan Bas-Serrano et al. "Logistic Q-Learning". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, 13–15 Apr 2021, pp. 3610–3618. URL: https://proceedings.mlr.press/v130/bas-serrano21a.html.

[101] Jacob Buckman, Carles Gelada, and Marc G. Bellemare. "The Importance of Pessimism in Fixed-Dataset Policy Optimization". In: *ICLR*. OpenReview.net, 2021.

[102] Peter Grünwald, Thomas Steinke, and Lydia Zakynthinou. "PAC-Bayes, MAC-Bayes and Conditional Mutual Information: Fast rate bounds that handle general VC classes". In: *COLT*. Vol. 134. Proceedings of Machine Learning Research. PMLR, 2021, pp. 2217–2247.

[103] Ying Jin, Zhuoran Yang, and Zhaoran Wang. "Is Pessimism Provably Efficient for Offline RL?" In: *ICML*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 5084–5096.

[104] Ying Jin, Zhuoran Yang, and Zhaoran Wang. "Is pessimism provably efficient for offline rl?" In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5084–5096.

[105] Thodoris Lykouris et al. "Corruption-robust exploration in episodic reinforcement learning". In: *COLT*. Vol. 134. Proceedings of Machine Learning Research. PMLR, 2021, pp. 3242–3245.

[106] Gergely Neu and Julia Olkhovskaya. "Online learning in MDPs with linear function approximation and bandit feedback". In: *NeurIPS*. 2021, pp. 10407–10417.

[107] Paria Rashidinejad, Banghua Zhu, et al. "Bridging Offline Reinforcement Learning and Imitation Learning: A Tale of Pessimism". In: *NeurIPS*. 2021, pp. 11702–11716.

[108] Paria Rashidinejad, Banghua Zhu, et al. “Bridging Offline Reinforcement Learning and Imitation Learning: A Tale of Pessimism”. In: (2021), pp. 11702–11716.

[109] Masatoshi Uehara and Wen Sun. “Pessimistic Model-based Offline Reinforcement Learning under Partial Coverage”. In: *arXiv preprint arXiv:2107.06226* (2021).

[110] Chenjun Xiao et al. “On the Optimality of Batch Policy Optimization Algorithms”. In: *ICML*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 11362–11371.

[111] Tengyang Xie, Ching-An Cheng, et al. “Bellman-consistent Pessimism for Offline Reinforcement Learning”. In: *NeurIPS*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 6683–6694. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/34f98c7c5d7063181da890ea8d25265a-Paper.pdf.

[112] Tengyang Xie and Nan Jiang. “Batch Value-function Approximation with Only Realizability”. In: *ICML*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 11404–11413.

[113] Andrea Zanette, Martin J. Wainwright, and Emma Brunskill. “Provable Benefits of Actor-Critic Methods for Offline Reinforcement Learning”. In: *NeurIPS*. 2021, pp. 13626–13640.

[114] Ruohan Zhan et al. “Policy learning with adaptively collected data”. In: *arXiv preprint arXiv:2105.02344* (2021).

[115] Ching-An Cheng et al. “Adversarially Trained Actor Critic for Offline Reinforcement Learning”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 17–23 Jul 2022, pp. 3852–3878. URL: https://proceedings.mlr.press/v162/cheng22b.html.

[116] Ying Jin, Zhimei Ren, et al. “Policy learning “without” overlap: Pessimism and generalized empirical Bernstein’s inequality”. In: *arXiv preprint arXiv:2212.09900* (2022).

[117] Gene Li, Cong Ma, and Nati Srebro. “Pessimism for Offline Linear Contextual Bandits using ℓ_p Confidence Sets”. In: *NeurIPS*. 2022.

[118] Paria Rashidinejad, Banghua Zhu, et al. “Bridging Offline Reinforcement Learning and Imitation Learning: A Tale of Pessimism”. In: *IEEE Trans. Inf. Theory* 68.12 (2022), pp. 8156–8196.

[119] Masatoshi Uehara and Wen Sun. “Pessimistic Model-based Offline Reinforcement Learning under Partial Coverage”. In: *ICLR*. OpenReview.net, 2022.

[120] Wenhao Zhan et al. “Offline Reinforcement Learning with Realizability and Single-policy Concentrability”. In: *Proceedings of Thirty Fifth Conference on Learning Theory*. Ed. by Po-Ling Loh and Maxim Raginsky. Vol. 178. Proceedings of Machine Learning Research. PMLR, Feb. 2022, pp. 2730–2775. URL: https://proceedings.mlr.press/v178/zhan22a.html.

[121] Xuezhou Zhang et al. “Corruption-robust Offline Reinforcement Learning”. In: *AISTATS*. Vol. 151. Proceedings of Machine Learning Research. PMLR, 2022, pp. 5757–5773.

[122] Hamish Flynn et al. “PAC-Bayes Bounds for Bandit Problems: A Survey and Experimental Comparison”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[123] Germano Gabbianelli, Gergely Neu, and Matteo Papini. “Online Learning with Off-Policy Feedback”. In: *ALT*. Ed. by Shipra Agrawal and Francesco Orabona. Vol. 201. Proceedings of Machine Learning Research. PMLR, 20 Feb–23 Feb 2023, pp. 620–641. URL: https://proceedings.mlr.press/v201/gabbianelli23a.html.

[124] Gergely Neu and Nneka Okolo. “Efficient Global Planning in Large MDPs via Stochastic Primal-Dual Optimization”. In: *ALT*. Vol. 201. Proceedings of Machine Learning Research. PMLR, 2023, pp. 1101–1123.

[125] Paria Rashidinejad, Hanlin Zhu, et al. “Optimal Conservative Offline RL with General Function Approximation via Augmented Lagrangian”. In: *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/pdf?id=ZsvWb6mJnMv.

[126] Otmane Sakhi, Pierre Alquier, and Nicolas Chopin. “PAC-Bayesian Offline Contextual Bandits With Guarantees”. In: *ICML*. Vol. 202.

Proceedings of Machine Learning Research. PMLR, 2023, pp. 29777–29799.

[127] Lequn Wang, Akshay Krishnamurthy, and Aleksandrs Slivkins. "Oracle-Efficient Pessimism: Offline Policy Optimization in Contextual Bandits". In: *arXiv preprint arXiv:2306.07923* (2023).

[128] Zhengyuan Zhou, Susan Athey, and Stefan Wager. "Offline Multi-Action Policy Learning: Generalization and Optimization". In: *Oper. Res.* 71.1 (2023), pp. 148–183.

[129] Hanlin Zhu, Paria Rashidinejad, and Jiantao Jiao. *Importance Weighted Actor-Critic for Optimal Conservative Offline Reinforcement Learning*. 2023. arXiv: 2301.12714 [cs.LG].

Appendix A

Proofs for Chapter 5

A.1 The proof of Lemma 5.5.1

We study the evolution of the potential function $\frac{1}{\eta}\log\frac{W_{n+1}}{W_1}$. On the one hand, we have for any action $\bar{a}$ that

$$\frac{1}{\eta}\log\frac{W_{n+1}}{W_1} = \frac{1}{\eta}\log\left(\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} w_{n+1}(a)\right) \geq \frac{1}{\eta}\log\left(\frac{1}{|\mathcal{A}|}\, w_{n+1}(\bar{a})\right) \quad \text{(A.1)}$$
$$= \sum_{t=1}^{n}\hat{r}_t(a) - \frac{\log|\mathcal{A}|}{\eta}.$$

Multiplying this bound with $\pi^*(\bar{a})$ and summing up over actions gives the lower bound

$$\frac{1}{\eta}\log\frac{W_{n+1}}{W_1} \geq \sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)\hat{r}_t(a) - \frac{1}{\eta}\log|\mathcal{A}|. \quad \text{(A.2)}$$

On the other hand, the potential can be rewritten as follows:

$$\frac{1}{\eta}\log\frac{W_{n+1}}{W_1} = \frac{1}{\eta}\sum_{t=1}^{n}\log\frac{W_{t+1}}{W_t}$$
$$= \frac{1}{\eta}\sum_{t=1}^{n}\log\frac{\sum_{a\in\mathcal{A}} w_t(a)e^{\eta\hat{r}_t(a)}}{W_t} = \frac{1}{\eta}\sum_{t=1}^{n}\log\sum_{a\in\mathcal{A}}\pi_t(a)e^{\eta\hat{r}_t(a)}.$$

Putting the two expressions together concludes the proof.

A.2 The proof of Theorem 5.3.1

In this proof, we have to face the added challenge of having to account for the possible inaccuracy of our estimator of μ. To this end, we define a sequence of "good events" under which the policy estimate is well-concentrated and analyze the regret under this event and its complement, using that the good event should hold with high probability. Concretely, we define the failure probability $\delta_t \in (0,1)$, the tolerance parameter ε_t, and the t-th good event as follows:

$$\varepsilon_1 = 1\,, \quad \varepsilon_t = \sqrt{\frac{\log(|\mathcal{A}|/\delta_t)}{2(t-1)}} \quad E_t = \{|\widehat{\mu}_t(a) - \mu(a)| \le \varepsilon_t \ (\forall a \in \mathcal{A})\}. \tag{A.3}$$

TODO: resume from here

An application of Hoeffding's inequality shows that E_t holds with probability at least $1-\delta_t$. Now, setting $\gamma_t = \varepsilon_t + \eta/2$, we can observe that under event E_t, we have

$$\widehat{r}_t(a) = \frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\widehat{\mu}_t + \gamma_t} \le \frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\mu(a) + \eta/2} \le \frac{1}{\eta}\log(1 + \eta\widehat{r}_t^{\mathrm{IW}}(a)). \tag{A.4}$$

We proceed by noticing that the bound of Lemma 5.5.1 continues to apply, and that we can bound the term appearing on the right-hand side as follows:

$$\begin{aligned}
&\mathbb{E}_t\left[\frac{1}{\eta}\log\sum_{a\in\mathcal{A}}\pi_t(a)\exp\left(\eta\widehat{r}_t(a)\right)\right] \\
&\quad\le \mathbb{1}\{E_t\}\mathbb{E}_t\left[\frac{1}{\eta}\log\sum_{a\in\mathcal{A}}\pi_t(a)\exp\left(\eta\widehat{r}_t(a)\right)\right] + \mathbb{1}\{\overline{E}_t\}\frac{2}{\eta} \\
&\quad\le \mathbb{1}\{E_t\}\mathbb{E}_t\left[\sum_{a\in\mathcal{A}}\pi_t(a)\widehat{r}_t^{\mathrm{IW}}(a)\right] + \mathbb{1}\{\overline{E}_t\}\frac{2}{\eta} \le \sum_{a\in\mathcal{A}}\pi_t(a)g_t(a) + \mathbb{1}\{\overline{E}_t\}\frac{2}{\eta},
\end{aligned}$$

where in the first line we used that $e^{\eta\widehat{r}_t(a)} \le e^{\eta/\gamma_t} \le e^2$ and in the second line we used the bound of Equation (A.4), the fact that E_t is $\mathcal{F}_{t-1}$-measurable, that $\mathbb{E}_t[\widehat{r}_t^{\mathrm{IW}}(a)] = g_t(a)$, and finally upper bounded the indicator $\mathbb{1}\{E_t\}$ by one. Taking marginal expectations and summing up for

all t, we get

$$\mathbb{E}\left[\frac{1}{\eta}\sum_{t=1}^{n}\log\sum_{a\in\mathcal{A}}\pi_t(a)\exp\left(\eta\hat{r}_t(a)\right)\right] \le \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi_t(a)g_t(a)\right] + \frac{2}{\eta}\sum_{t=1}^{n}\delta_t,$$

where we used $\mathbb{E}\left[\mathbb{1}\{\overline{E}_t\}\right] \le \delta_t$.

It thus remains to relate the term on the left-hand side of the bound of Lemma 5.5.1. To do this, we similarly write

$$\begin{aligned}\mathbb{E}_t[\hat{r}_t(a)] &\ge \mathbb{1}\{E_t\}\mathbb{E}_t[\hat{r}_t(a)] = \mathbb{1}\{E_t\}\mathbb{E}_t\left[\frac{g_t(a)\mathbb{1}\{A_t^\mu = a\}}{\hat{\mu}_t(a)+\gamma_t}\right] \\ &\ge \mathbb{1}\{E_t\}\cdot\frac{g_t(a)\mu(a)}{\mu(a)+\varepsilon_t+\gamma_t} \ge \mathbb{1}\{E_t\}g_t(a) - \frac{\varepsilon_t+\gamma_t}{\mu(a)},\end{aligned}$$

where in the first inequality we exploited that $\hat{r}_t(a)$ is nonnegative, in the second one we used that E_t is $\sigma(\tilde{h}_t)$-measurable and the defining property of the good event, and in the last one we simplified some expressions. Thus, we have

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)g_t(a)\right] \\ &\quad\le \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)\left(\hat{r}_t(a) + (1-\mathbb{1}\{E_t\})g_t(a) + \frac{\varepsilon_t+\gamma_t}{\mu(a)}\right)\right] \\ &\quad= \mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a)\hat{r}_t(a)\right] + \sum_{t=1}^{n}\delta_t + \sum_{a\in\mathcal{A}}\frac{\pi^*(a)}{\mu(a)}\sum_{t=1}^{n}\left(2\varepsilon_t+\frac{\eta}{2}\right),\end{aligned}$$

where in the last line we recalled that $\gamma_t = \varepsilon_t + \eta/2$. Putting the two bounds together, we arrive to

$$\begin{aligned}&\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\left(\pi^*(a)g_t(a) - \pi_t(a)g_t(a)\right)\right] \\ &\qquad\le \frac{\log|\mathcal{A}|}{\eta} + \left(1+\frac{2}{\eta}\right)\sum_{t=1}^{n}\delta_t + \left(\frac{\eta n}{2} + 2\sum_{t=1}^{n}\varepsilon_t\right)\sum_{a\in\mathcal{A}}\frac{\pi^*(a)}{\mu(a)}\end{aligned}$$

Finally, we set $\delta_1 = 0$, $\delta_t = (t-1)^{-2}$ so that we have $\sum_{t=1}^{n} \delta_t = \pi^2/6 \leq 2$ and we can write

$$\sum_{t=1}^{n} \varepsilon_t = 1 + \sum_{t=1}^{n-1} \sqrt{\frac{\log(|\mathcal{A}|t^2)}{2t}} \leq 2\sqrt{n \log(|\mathcal{A}|n)},$$

where we also used the standard upper bound $\sum_{t=1}^{n} 1/\sqrt{t} \leq 2\sqrt{n}$. Putting everything together, we finally get

$$\mathbb{E}\left[\sum_{t=1}^{n} \sum_{a \in \mathcal{A}} (\pi^*(a) g_t(a) - \pi_t(a) g_t(a))\right] \leq \frac{16 + \log|\mathcal{A}|}{\eta} + \left(\frac{\eta n}{2} + 2\sqrt{n \log(|\mathcal{A}|n)}\right) C^{\dagger}(\pi^*) + 2. \quad \text{(A.5)}$$

Setting $\eta = \sqrt{\frac{\log|\mathcal{A}|}{n}}$ concludes the proof.

A.3 The proof of Theorem 5.4.1

The proof combines ideas from the previous two proofs with ideas from Neu and Olkhovskaya (2020) to deal with the contextual aspect of the problem setting. In the following, let $\hat{r}_t(x,a) = \langle \hat{\theta}_t, \varphi(x,a) \rangle$. As a starting point, we fix a context $x \in \mathcal{X}$ and define the estimated regret in context x against comparator π^* as

$$\hat{\mathfrak{R}}(\pi^*, x) = \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} (\pi^*(a \mid x) - \pi_t(a \mid x))\, \hat{r}_t(x,a). \quad \text{(A.6)}$$

The following lemma gives a bound on the above quantity:

Lemma A.3.1. *Suppose that $\eta \hat{r}_t(x,a) \geq -1/2$ holds for all x, a. Then, for any fixed x and π^*,*

$$\hat{\mathfrak{R}}(\pi^*, x) \leq \frac{\log|\mathcal{A}|}{\eta} + \eta \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \pi^*(a \mid x)\, (\hat{r}_t(x,a))^2. \quad \text{(A.7)}$$

The proof follows from a careful combination of techniques by Cesa-Bianchi, Mansour, and Stoltz (2007) and Neu and Olkhovskaya (2020),

and is deferred to Appendix A.4.1. We proceed by noting that for any fixed x, the second term in the bound can be bounded as follows:

$$\begin{aligned}
&\mathbb{E}_t[\hat{r}_t(x,a)^2] \\
&\quad = \mathbb{E}_t\left[\left(R_t^\mu\right)^2 \varphi(x,a)^\intercal \mathbf{\Lambda}_\mu^{-1} \varphi(X_t, A_t^\mu) \varphi(X_t, A_t^\mu)^\top \mathbf{\Lambda}_\mu^{-1} \varphi(x,a)\right] \\
&\quad \leq \varphi(x,a)^\top \mathbf{\Lambda}_\mu^{-1} \mathbf{\Lambda}_\mu \mathbf{\Lambda}_\mu^{-1} \varphi(x,a) \\
&\quad = \varphi(x,a)^\intercal \mathbf{\Lambda}_\mu^{-1} \varphi(x,a) \\
&\quad = \operatorname{Tr}(\mathbf{\Lambda}_\mu^{-1} \varphi(x,a) \varphi(x,a)^\intercal),
\end{aligned} \tag{A.8}$$

where we have used $R_t^\mu \leq 1$ in the inequality. Furthermore, in order to use the lemma, we first need to verify that its precondition is satisfied. To this end, notice that

$$|\hat{r}_t(x,a)| = \left|R_t^\mu \phi(x,a)^\intercal \mathbf{\Lambda}_\mu^{-1} \phi(X_t, A_t^\mu)\right| \leq \frac{\sup_{x,a} \|\phi(x,a)\|_2^2}{\lambda_{\min}(\mathbf{\Lambda}_\mu)},$$

which follows from a straightforward applicaiton of the Cauchy–Schwarz inequality. Thus, the condition on η we impose in the theorem guarantees that $\eta|\hat{r}_t(x,a)| \leq 1/2$. Now we are in position to invoke Lemma A.3.1, albeit with a specific choice for the context x. Specifically, we let X_0 be a "ghost sample" drawn independently from the context distribution for the sake analysis, and apply Lemma A.3.1 to obtain

$$\hat{\mathfrak{R}}(\pi^*, X_0) \leq \frac{\log|\mathcal{A}|}{\eta} + \eta \sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \pi^*(a \mid X_0) \operatorname{Tr}(\mathbf{\Lambda}_\mu^{-1} \varphi(X_0,a) \varphi(X_0,a)^\intercal). \tag{A.9}$$

Then, a straightforward calculation inspired by the analysis of Neu and Olkhovskaya (2020) shows that the left-hand side is related to the expected regret as

$$\mathbb{E}\left[\hat{\mathfrak{R}}(\pi^*, X_0)\right] = \mathbb{E}\left[\sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \left(\pi^*(a|X_t) - \pi_t(a|X_t)\right) g_t(X_t, a)\right]. \tag{A.10}$$

For completeness, we include this calculation in Appendix A.4. The same technique can be used to deal with the term on the right-hand side as

follows:

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi^*(a\mid X_0)\,\mathrm{Tr}\left(\mathbf{\Lambda}_\mu^{-1}\phi(X_0,a)\phi(X_0,a)^\intercal\right)\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{n}\mathrm{Tr}\left(\mathbf{\Lambda}_\mu^{-1}\sum_{a\in\mathcal{A}}\pi^*(a\mid X_0)\phi(X_0,a)\phi(X_0,a)^\intercal\right)\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{n}\mathrm{Tr}\left(\mathbf{\Lambda}_\mu^{-1}\mathbf{\Lambda}_\pi^*\right)\right]=n\cdot C^\dagger(\pi^*).
\end{aligned}$$

Thus, taking expectations of both sides of Equation (A.9) and using the above two results concludes the proof. □

A.4 The proof of the regret decomposition of Equation (A.10)

We start by fixing an arbitrary x and defining the following notion of pseudo-regret in context x:

$$\tilde{\mathfrak{R}}(\pi^*,x)=\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\left(\pi^*(a\mid x)-\pi_t(a\mid x)\right)g_t(x,a).$$

We first note that $\mathbb{E}[\hat{\mathfrak{R}}(\pi^*,x)]=\mathbb{E}[\tilde{\mathfrak{R}}(\pi^*,x)]$ holds thanks to the unbiasedness of $\hat{r}_t$ and the independence of π_t and $\hat{r}_t$. In particular, this follows from the following derivation:

$$\begin{aligned}
\mathbb{E}[\hat{\mathfrak{R}}(\pi^*,x)]&=\mathbb{E}\left[\sum_{t=1}^{n}\mathbb{E}_t\Big[\sum_{a\in\mathcal{A}}\left(\pi^*(a\mid x)-\pi_t(a\mid x)\right)\hat{r}_t(x,a)\Big]\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\left(\pi^*(a\mid x)-\pi_t(a\mid x)\right)\mathbb{E}_t\Big[\hat{r}_t(x,a)\Big]\right]\\
&=\mathbb{E}\left[\sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\left(\pi^*(a\mid x)-\pi_t(a\mid x)\right)g_t(x,a)\right]=\mathbb{E}[\tilde{\mathfrak{R}}(\pi^*,x)],
\end{aligned}$$

where we used the tower rule of expectation in the first step, the fact that π_t is $\sigma(\tilde{h}_t)$-measurable in the second step, and the unbiasedness of the reward estimator in the last step. To relate $\mathbb{E}[\tilde{\mathfrak{R}}(\pi^*, x)]$ and the true expected Regret $\mathfrak{R}(\pi^*)$, we consider the random variable $\tilde{\mathfrak{R}}(\pi^*, X_0)$ with X_0 being a ghost sample drawn from the context distribution independently from the history of contexts $(X_t)_{t=1}^n$. Then, we can write the expectation of this random variable as

$$\begin{aligned}
\mathbb{E}[\tilde{\mathfrak{R}}(\pi^*, X_0)] &= \mathbb{E}\left[\sum_{t=1}^{n} \sum_{a \in \mathcal{A}} \left(\pi^*(a \mid X_0) - \pi_t(a \mid X_0)\right) g_t(X_0, a)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{E}_t\left[\sum_{a \in \mathcal{A}} \left(\pi^*(a \mid X_0) - \pi_t(a \mid X_0)\right) g_t(X_0, a)\right]\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{E}_t\left[\sum_{a \in \mathcal{A}} \left(\pi^*(a \mid X_t) - \pi_t(a \mid X_t)\right) g_t(X_t, a)\right]\right] = \mathfrak{R}(\pi^*),
\end{aligned}$$

where the second line uses the tower rule of expectation and the third one the fact that X_0 is distributed identically with X_t given $\tilde{h}_t$. This concludes the proof. □

A.4.1 Proof of Lemma A.3.1

The proof is inspired by the classic Prod analysis of Cesa-Bianchi, Mansour, and Stoltz (2007), and follows from similar arguments as the proof of Lemma 5.5.1. The main adjustment we need to these proofs is that now we have to include contexts in our derivations. To this end, let us fix one context $x \in \mathcal{X}$ and suppose that the condition of the theorem is satisfied: $\eta\, \hat{r}_t(x, a) \geq -1/2$ for all actions $a \in \mathcal{A}$.

As before, we will study the evolution of the potential function $\frac{1}{\eta} \log \frac{W_{n+1}(x)}{W_1(x)}$. For every action $\bar{a} \in \mathcal{A}$ we have:

$$\begin{aligned}
\frac{1}{\eta} \log W_{n+1}(x) &= \frac{1}{\eta} \log \sum_{a \in \mathcal{A}} \prod_{t=1}^{n} (1 + \eta \hat{r}_t(x, a)) \geq \frac{1}{\eta} \log \prod_{t=1}^{n} (1 + \eta \hat{r}_t(x, \bar{a})) \\
&= \frac{1}{\eta} \sum_{t=1}^{n} \log(1 + \eta \hat{r}_t(x, \bar{a})) \geq \sum_{t=1}^{n} \left(\hat{r}_t(x, \bar{a}) - \eta \left(\hat{r}_t(x, \bar{a})\right)^2\right),
\end{aligned}$$

where we used our condition on the magnitude of the reward estimates twice: once to use $(1-\eta\hat{r}_t(x,a)) \geq 0$ in the first line and once when using the elementary inequality $\log(1+z) \geq z - z^2$ that holds for all $z \geq -1/2$ in the second line. Moreover, we can upper-bound the potential as

$$\begin{aligned}
\frac{1}{\eta}\log W_{n+1}(x) &= \frac{1}{\eta}\log W_1 + \frac{1}{\eta}\log\prod_{t=1}^{n}\frac{W_{t+1}(x)}{W_t(x)} \\
&= \frac{\log|\mathcal{A}|}{\eta} + \frac{1}{\eta}\sum_{t=1}^{n}\log\frac{W_{t+1}(x)}{W_t(x)} \\
&= \frac{\log|\mathcal{A}|}{\eta} + \frac{1}{\eta}\sum_{t=1}^{n}\log\sum_{a\in\mathcal{A}}\frac{\hat{r}_t(x,a)}{W_t(x)}(1+\eta\,\hat{r}_t(x,a)) \qquad \text{(def. of } W_{t+1}\text{)} \\
&= \frac{\log|\mathcal{A}|}{\eta} + \frac{1}{\eta}\sum_{t=1}^{n}\log\sum_{a\in\mathcal{A}}\pi_t(a\mid x)(1+\eta\,\hat{r}_t(x,a)) \qquad \text{(def. of } \pi_t\text{)} \\
&= \frac{\log|\mathcal{A}|}{\eta} + \frac{1}{\eta}\sum_{t=1}^{n}\log\Big(1+\eta\sum_{a\in\mathcal{A}}\pi_t(a\mid x)\hat{r}_t(x,a)\Big) \\
&\leq \frac{\log|\mathcal{A}|}{\eta} + \sum_{t=1}^{n}\sum_{a\in\mathcal{A}}\pi_t(a\mid x)\hat{r}_t(x,a),
\end{aligned}$$

where we used the inequality $\log(1+z) \leq z$ that holds for all $z > -1$.

Combining the lower bound and upper bound, we obtain

$$\sum_{t=1}^{n}\Big(\hat{r}_t(x,\bar{a}) - \sum_{a\in\mathcal{A}}\pi_t(a\mid x)\hat{r}_t(x,a)\Big) \leq \frac{\log|\mathcal{A}|}{\eta} + \eta\sum_{t=1}^{n}\left(\hat{r}_t(x,\bar{a})\right)^2.$$

Multiplying both sides by $\pi^*(\bar{a}\mid x)$ and summing over all actions $\bar{a}\in\mathcal{A}$ yields the desired result. □

Appendix B

Proofs for Chapter 6

In this section, we prove our main technical lemmas. To facilitate this effort, we introduce the shorthand notations

$$\hat{r}_t^{\mathrm{IW}}(\pi) = \frac{\pi(A_t|X_t)}{\mu(A_t|X_t)} R_t, \qquad \text{and} \qquad \hat{r}_t(\pi) = \frac{\pi(A_t|X_t)}{\mu(A_t|X_t) + \gamma} R_t,$$

and note that $\hat{v}^{\mathrm{IW}}(\pi) = \frac{1}{n}\sum_{t=1}^n \hat{r}_t^{\mathrm{IW}}(\pi)$ and $\hat{v}_n = \frac{1}{n}\sum_{t=1}^n \hat{r}_t(\pi)$, and also recall that $\mathbb{E}[\hat{r}_t^{\mathrm{IW}}(\pi)] = \rho(\pi)$ and $\mathbb{E}[\hat{r}_t(\pi)] = \rho(\pi) - \gamma C_\gamma(\pi)$ holds for all t.

B.1 The proof of Lemma 6.3.2

Fix an arbitrary $\pi \in \Pi$. We start by using the elementary inequality $\log(1+y) \geq \frac{y}{1+y/2}$ that holds for all $y \geq 0$ to show that

$$\begin{aligned}\hat{r}_t(\pi) &= \frac{\pi(A_t|X_t)R_t}{\mu(A_t|X_t) + \gamma} \leq \frac{\pi(A_t|X_t)R_t}{\mu(A_t|X_t) + \gamma\pi(A_t|X_t)R_t} \\ &= \frac{1}{2\gamma} \cdot \frac{2\gamma\hat{r}_t^{\mathrm{IW}}(\pi)}{1 + \gamma\hat{r}_t^{\mathrm{IW}}(\pi)} \leq \frac{\log\left(1 + 2\gamma\hat{r}_t^{\mathrm{IW}}(\pi)\right)}{2\gamma}.\end{aligned}$$

This implies that

$$\mathbb{E}[e^{2\gamma\hat{r}_t(\pi)}] \leq \mathbb{E}[1 + 2\gamma\hat{r}_t^{\mathrm{IW}}(\pi)] = 1 + 2\gamma\rho(\pi) \leq e^{2\gamma\rho(\pi)},$$

where the last step follows from the inequality $e^y \geq 1 + y$ that holds for all $y \in \mathbb{R}$. Using the independence of all observations, this implies $\mathbb{E}[e^{2\gamma \sum_{t=1}^{n} (\hat{r}_t(\pi) - \rho(\pi))}] \leq 1$, and thus an application of Markov's inequality yields

$$\mathbb{P}\left[\sum_{t=1}^{n} (\hat{r}_t(\pi) - \rho(\pi)) \geq \varepsilon\right] = \mathbb{P}\left[e^{2\gamma \sum_{t=1}^{n} (\hat{r}_t(\pi) - \rho(\pi))} \geq e^{2\gamma\varepsilon}\right] \leq e^{-2\gamma\varepsilon}$$

for any $\varepsilon \geq 0$. Setting $\varepsilon = \frac{\log(|\Pi|/\delta)}{2\gamma}$ and taking a union bound over all policies concludes the proof. □

B.2 The proof of Lemma 6.3.3

Fix an arbitrary $\pi \in \Pi$. We start by noting that for any nonnegative random variable Y, and for any positive λ, we have

$$\mathbb{E}[e^{-\lambda Y}] \leq \mathbb{E}[1 - \lambda Y + \lambda^2 Y^2/2] \leq e^{-\lambda \mathbb{E}[Y] + \lambda^2 \mathbb{E}[Y^2]/2},$$

where the first inequality follows from $e^{-y} \leq 1 - y + y^2/2$ that holds for all $y \geq 0$ and the second from $e^y \geq 1 + y$ that holds for all $y \in \mathbb{R}$. Apply this inequality with $Y = \hat{r}_t(\pi)$ and note that

$$\begin{aligned}
\mathbb{E}\left[\left(\hat{r}_t(\pi)\right)^2\right] &= \mathbb{E}\left[\frac{\left(\pi(A_t|X_t)\right)^2}{(\mu(A_t|X_t) + \gamma)^2} \cdot R_t^2\right] \\
&\leq \mathbb{E}\left[\sum_a \mathbb{I}_{\{A_t = a\}} \frac{\pi(a|X_t)}{(\mu(a|X_t) + \gamma)^2} \cdot g(X_t, a)\right] \\
&\leq \mathbb{E}\left[\sum_a \frac{\pi(a|X_t)}{\mu(a|X_t) + \gamma} \cdot g(X_t, a)\right] = C_\gamma(\pi),
\end{aligned}$$

where the first inequality used the boundedness of the rewards to show $\mathbb{E}\left[R_t^2\right] \leq \mathbb{E}\left[R_t\right] = \mathbb{E}\left[g(X_t, a)\right]$ and $\left(\pi(a|X_t)\right)^2 \leq \pi(a|X_t)$, and the second inequality used that $\mathbb{E}\left[\mathbb{I}_{\{A_t = a\}} \middle| X_t\right] = \mu(a|X_t)$.

Using the independence of all observations, this implies

$$\mathbb{E}[e^{\lambda \sum_{t=1}^{n} (\mathbb{E}[\hat{r}_t(\pi)] - \hat{r}_t(\pi) - \lambda C_\gamma(\pi)/2)}] \leq 1.$$

Recalling that $\mathbb{E}[\hat{r}_t(\pi)] = \rho(\pi) - \gamma C_\gamma(\pi)$, an application of Markov's inequality yields

$$\mathbb{P}\left[\sum_{t=1}^{n}\left(\rho(\pi) - \hat{r}_t(\pi) - (\gamma + \lambda/2)\, C_\gamma(\pi)\right) \geq \varepsilon\right] \leq e^{-2\gamma\varepsilon}.$$

Setting $\lambda = 2\gamma$ and $\varepsilon = \frac{\log(|\Pi|/\delta)}{2\gamma}$, and finally taking a union bound over all policies concludes the proof. □

B.3 The proofs of Lemmas 6.4.2 and 6.4.3

To prove Lemma 6.4.2, let us first fix an arbitrary $Q \in \Delta_\Pi$, and recall from the proof of Lemma 6.3.2 that $\mathbb{E}[e^{2\gamma\hat{r}_t(\pi)}] \leq e^{2\gamma\rho(\pi)}$, holds for all fixed π. Thus, since P is independent of the random observations, we also have

$$\mathbb{E}\left[\int e^{2\gamma\sum_{t=1}^{n}(\hat{r}_t(\pi)-\rho(\pi))}\mathrm{d}P(\pi)\right] \leq 1.$$

Now, let us introduce the notation $\rho_\pi(Q,P) = \log\frac{\mathrm{d}Q}{\mathrm{d}P}(\pi)$ and write

$$\begin{aligned}
&\mathbb{P}\left[\int\left(\sum_{t=1}^{n}(\hat{r}_t(\pi)-\rho(\pi)) - \frac{\rho_\pi(Q,P)}{2\gamma}\right)\mathrm{d}Q(\pi) \geq \varepsilon\right] \\
&\quad\leq \mathbb{E}\left[e^{2\gamma\int\left(\sum_{t=1}^{n}(\hat{r}_t(\pi)-\rho(\pi)) - \frac{\rho_\pi(Q,P)}{2\gamma}\right)\mathrm{d}Q(\pi)}\right] e^{-2\gamma\varepsilon} \\
&\quad\leq \mathbb{E}\left[\int e^{2\gamma\left(\sum_{t=1}^{n}(\hat{r}_t(\pi)-\rho(\pi)) - \frac{\rho_\pi(Q,P)}{2\gamma}\right)}\mathrm{d}Q(\pi)\right] e^{-2\gamma\varepsilon} \\
&\quad= \mathbb{E}\left[\int e^{2\gamma\left(\sum_{t=1}^{n}(\hat{r}_t(\pi)-\rho(\pi))\right)}\frac{\mathrm{d}P}{\mathrm{d}Q}(\pi)\mathrm{d}Q(\pi)\right] e^{-2\gamma\varepsilon} \\
&\quad= \mathbb{E}\left[\int e^{2\gamma\left(\sum_{t=1}^{n}(\hat{r}_t(\pi)-\rho(\pi))\right)}\mathrm{d}P(\pi)\right] e^{-2\gamma\varepsilon} \leq e^{-2\gamma\varepsilon}.
\end{aligned}$$

Here, the first step follows from Markov's inequality, the second from Jensen's inequality for the convex function $y \mapsto e^{2\gamma y}$, the third from the definition of $\rho_\pi(Q,P)$, the fourth from the definition of the Radon–Nykodim derivative $\frac{\mathrm{d}P}{\mathrm{d}Q}$, and the last step from the inequality that we have established above. Noticing that $\int \rho_\pi(Q,P)\mathrm{d}Q(\pi) = \mathrm{KL}\,(Q\|P)$ and setting $\varepsilon = \frac{\log(1/\delta)}{2\gamma}$ concludes the proof of Lemma 6.4.2. The proof

of Lemma 6.4.3 then follows analogously by recalling from the proof of Lemma 6.3.3 that $\mathbb{E}[e^{2\gamma(\rho(\pi)-\tilde{r}_t(\pi)-2\gamma C_\gamma(\pi))}] \leq 1$, and then following the same steps as above. □

Appendix C

Proofs for the discounted setting of Chapter 7

C.1 Proof of Lemma 7.3.1

Using the choice of comparators described in the lemma, we have

$$\begin{aligned}\nu_{\boldsymbol{\beta}^*}(s) &= (1-\gamma)\nu_0(s) + \gamma\langle \boldsymbol{\psi}(s), \boldsymbol{\Lambda}_\mu^c \boldsymbol{\Lambda}_\mu^{-c} \boldsymbol{\Phi}^\top p^{\pi^*}\rangle \\ &= (1-\gamma)\nu_0(s) + \sum_{s',a'} P(s|s',a') p^{\pi^*}(s',a') = \nu^{\pi^*}(s),\end{aligned}$$

hence $p_{\boldsymbol{\beta}^*,\pi^*} = \mathbf{p}^{\pi^*}$. From Equation (7.15) it is easy to see that

$$\begin{aligned}f(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}_t) &= \langle \boldsymbol{\Lambda}_\mu^{-c} \boldsymbol{\Phi}^\top \mathbf{p}^*, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\theta}_r\rangle + \langle \boldsymbol{\theta}_t, \boldsymbol{\Phi}^\top \mathbf{p}^* - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\Lambda}_\mu^{-c} \boldsymbol{\Phi}^\top \mathbf{p}^*\rangle \\ &= \langle p^{\pi^*}, \boldsymbol{\Phi}\boldsymbol{\theta}_r\rangle = \langle \mathbf{p}^*, \mathbf{r}\rangle.\end{aligned}$$

Moreover, we also have

$$\begin{aligned}v_{\boldsymbol{\theta}_t^*, \pi_t}(s) &= \sum_a \pi_t(a|s) \langle \boldsymbol{\theta}^{\pi_t}, \boldsymbol{\varphi}(x,a)\rangle \\ &= \sum_a \pi_t(a|s) q^{\pi_t}(s,a) = v^{\pi_t}(s,a).\end{aligned}$$

Then, from Equation (7.16) we obtain

$$\begin{aligned}
f(\boldsymbol{\theta}_t^*, \boldsymbol{\beta}_t, \pi_t) &= (1-\gamma)\langle \boldsymbol{\nu}_0, v^{\pi_t}\rangle + \langle \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^c(\boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi}\mathbf{v}^{\pi_t} - \boldsymbol{\theta}^{\pi_t})\rangle \\
&= (1-\gamma)\langle \boldsymbol{\nu}_0, v^{\pi_t}\rangle + \langle \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^{c-1}\mathbb{E}_{X,A\sim p^\mu}\left[\boldsymbol{\varphi}(X,A)\boldsymbol{\varphi}(X,A)^\intercal(\boldsymbol{\theta}_r + \gamma\boldsymbol{\Psi}\mathbf{v}^{\pi_t} - \boldsymbol{\theta}^{\pi_t})\right]\rangle \\
&= (1-\gamma)\langle \boldsymbol{\nu}_0, v^{\pi_t}\rangle \\
&\quad + \langle \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^{c-1}\mathbb{E}_{X,A\sim p^\mu}\left[[r(X,A) + \gamma\left\langle p(\cdot|X,A), \mathbf{v}^{\pi_t}\right\rangle - \mathbf{q}^{\pi_t}(X,A)]\boldsymbol{\varphi}(X,A)\right]\rangle \\
&= (1-\gamma)\langle \boldsymbol{\nu}_0, v^{\pi_t}\rangle = \langle p^{\pi_t}, \mathbf{r}\rangle,
\end{aligned}$$

where the fourth equality uses that the value functions satisfy the Bellman equation $\mathbf{q}^\pi = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}^\pi$ for any policy π. The proof is concluded by noticing that, since $\boldsymbol{\pi}_{\text{out}}$ is sampled uniformly from $\{\pi_t\}_{t=1}^T$,

$$\mathbb{E}\left[\langle \mathbf{p}^{\boldsymbol{\pi}_{\text{out}}}, \mathbf{r}\rangle\right] = \frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\langle \mathbf{p}^{\pi_t}, \mathbf{r}\rangle\right].$$

□

C.2 Proof of Lemma 7.3.2

We start by rewriting the terms appearing in the definition of $\mathcal{G}_T$:

$$\begin{aligned}
f(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t^*) &= f(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}^*, \pi_t; \boldsymbol{\theta}_t) \\
&\quad + f(\boldsymbol{\beta}^*, \pi_t; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t) \\
&\quad + f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t^*). \qquad \text{(C.1)}
\end{aligned}$$

To rewrite this as the sum of the three regret terms, we first note that

$$f(\boldsymbol{\beta}, \pi; \boldsymbol{\theta}) = \langle \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}, \boldsymbol{\theta}_r - \boldsymbol{\theta}_t\rangle + \langle \nu_{\boldsymbol{\beta}}, v_{\boldsymbol{\theta}_t,\pi}\rangle,$$

which allows us to write the first term of Equation (C.1) as

$$\begin{aligned}
f(\boldsymbol{\beta}^*, \pi^*; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}^*, \pi_t; \boldsymbol{\theta}_t) &= \langle \boldsymbol{\Lambda}_\mu^c(\boldsymbol{\beta}^* - \boldsymbol{\beta}^*), \boldsymbol{\theta}_r - \boldsymbol{\theta}_t\rangle + \langle \nu_{\boldsymbol{\beta}^*}, v_{\boldsymbol{\theta}_t,\pi^*} - v_{\boldsymbol{\theta}_t,\pi_t}\rangle \\
&= \langle \nu_{\boldsymbol{\beta}^*}, \sum_a (\pi^*(a|\cdot) - \pi_t(a|\cdot))\langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}(\cdot, a)\rangle\rangle,
\end{aligned}$$

and we have already established in the proof of Lemma E.2.1 that $\boldsymbol{\nu}_{\boldsymbol{\beta}^*}$ is equal to $\boldsymbol{\nu}^{\pi^*}$ for our choice of comparator.

Similarly, we use Equation (7.16) to rewrite the second term of Equation (C.1) as

$$f(\boldsymbol{\beta}^*, \pi_t; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t) = \\ (1-\gamma)\langle \boldsymbol{\nu}_0, v_{\boldsymbol{\theta}_t, \pi_t} - v_{\boldsymbol{\theta}_t, \pi_t} \rangle + \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^c (\boldsymbol{\theta}_r + \gamma \boldsymbol{\Psi} v_{\boldsymbol{\theta}_t, \pi_t} - \boldsymbol{\theta}_t) \rangle \\ = \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \boldsymbol{\kappa}_{\beta,t} \rangle.$$

Finally, we use Equation (7.15) to rewrite the third term of Equation (C.1) as

$$f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \boldsymbol{\theta}_t^*) = \langle \boldsymbol{\beta}_t - \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\theta}_r \rangle + \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\Phi}^\top \mathbf{p}_{\boldsymbol{\beta}_t, \pi_t} - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t \rangle \\ = \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\kappa}_{\theta,t} \rangle.$$

C.3 Regret bounds for stochastic gradient descent / ascent

Lemma C.3.1. *For any dynamic comparator $\boldsymbol{\theta}_{1:T} \in D_{\boldsymbol{\theta}^\vee}$, the iterates $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T$ of Algorithm 3 satisfy the following regret bound:*

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\kappa}_{\theta,t} \rangle\right] \leq \frac{2T D_{\boldsymbol{\theta}}^2}{\eta K} + \frac{3\eta T D_{\boldsymbol{\varphi}}^2 \left((1-\gamma)^2 + (1+\gamma^2) D_\beta^2 \left\| \boldsymbol{\Lambda}_\mu \right\|_2^{2c-1} \right)}{2}.$$

Proof. First, we use the definition of $\boldsymbol{\theta}_t$ as the average of the inner-loop iterates from Algorithm 3, together with linearity of expectation and bilinearity of the inner product.

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\kappa}_{\theta,t} \rangle\right] = \sum_{t=1}^{T} \frac{1}{K} \underbrace{\mathbb{E}\left[\sum_{k=1}^{K} \langle \boldsymbol{\theta}_{t,k} - \boldsymbol{\theta}_t^*, \boldsymbol{\kappa}_{\theta,t} \rangle\right]}_{\text{regret}_t}. \tag{C.2}$$

We then appeal to standard stochastic gradient descent analysis to bound each term regret_t separately.

We have already proven in Section 7.3 that the gradient estimator for $\boldsymbol{\theta}$ is unbiased, that is, $\mathbb{E}_{t,k}[\hat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k}] = \boldsymbol{\kappa}_{\theta,t}$. It is also useful to recall here that

$\hat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,k}$ does *not* depend on $\boldsymbol{\theta}_{t,k}$. Next, we show that its second moment is bounded. From Equation (7.10), plugging in the definition of $p_{t,k}$ from Equation (7.8) and using the abbreviations $\boldsymbol{\varphi}^0_{t,k} = \sum_a \pi_t(a|x^0_{t,k})\boldsymbol{\varphi}(x^0_{t,k}, a)$, $\boldsymbol{\varphi}_t = \boldsymbol{\varphi}(x_{t,k}, a_{t,k})$, and $\boldsymbol{\varphi}'_{t,k} = \sum_a \pi_t(a|x^0_{t,k})\boldsymbol{\varphi}(x'_{t,k}, a)$, we have:

$$
\begin{aligned}
&\mathbb{E}_{t,k}[\left\|\hat{\boldsymbol{\kappa}}_{\boldsymbol{\theta},t,i}\right\|^2] \\
&\quad= \mathbb{E}_{t,k}[\left\|(1-\gamma)\boldsymbol{\varphi}^0_{t,k} + \gamma\boldsymbol{\varphi}'_{t,k}\langle\boldsymbol{\varphi}_{tk}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle - \boldsymbol{\varphi}_{t,k}\langle\boldsymbol{\varphi}_{tk}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle\right\|^2] \\
&\quad\leq 3(1-\gamma)^2 D_{\boldsymbol{\varphi}}^2 + 3\gamma^2\mathbb{E}_{t,k}[\left\|\boldsymbol{\varphi}'_{t,k}\langle\boldsymbol{\varphi}_{tk}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle\right\|^2] + 3\mathbb{E}_{t,k}[\left\|\boldsymbol{\varphi}_{t,k}\langle\boldsymbol{\varphi}_{tk}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle\right\|^2] \\
&\quad\leq 3(1-\gamma)^2 D_{\boldsymbol{\varphi}}^2 + 3(1+\gamma^2)D_{\boldsymbol{\varphi}}^2\mathbb{E}_{t,k}[\langle\boldsymbol{\varphi}_{tk}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\rangle^2] \\
&\quad= 3(1-\gamma)^2 D_{\boldsymbol{\varphi}}^2 + 3(1+\gamma^2)D_{\boldsymbol{\varphi}}^2\boldsymbol{\beta}_t^\top\boldsymbol{\Lambda}_\mu^{c-1}\mathbb{E}_{t,k}[\boldsymbol{\varphi}_{tk}\boldsymbol{\varphi}_{tk}^\top]\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t \\
&\quad= 3(1-\gamma)^2 D_{\boldsymbol{\varphi}}^2 + 3(1+\gamma^2)D_{\boldsymbol{\varphi}}^2\left\|\boldsymbol{\beta}_t\right\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}}.
\end{aligned}
$$

We can then apply Lemma D.0.1 with the latter expression as G^2, $\mathbb{B}(D_{\boldsymbol{\theta}})$ as the domain, and η as the learning rate, obtaining:

$$
\begin{aligned}
\mathbb{E}_t\left[\sum_{k=1}^K\langle\boldsymbol{\theta}_{t,k} - \boldsymbol{\theta}^*_t, \boldsymbol{\kappa}_{\theta,t}\rangle\right] &\leq \frac{\left\|\boldsymbol{\theta}_{t,1} - \boldsymbol{\theta}^*_t\right\|_2^2}{2\eta} + \frac{3\eta K D_{\boldsymbol{\varphi}}^2\left((1-\gamma)^2 + (1+\gamma^2)\left\|\boldsymbol{\beta}_t\right\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}}\right)}{2} \\
&\leq \frac{2D_\theta^2}{\eta} + \frac{3\eta K D_{\boldsymbol{\varphi}}^2\left((1-\gamma)^2 + (1+\gamma^2)\left\|\boldsymbol{\beta}_t\right\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}}\right)}{2}.
\end{aligned}
$$

Plugging this into Equation (C.2) and bounding $\left\|\boldsymbol{\beta}_t\right\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}} \leq D_{\boldsymbol{\beta}}^2\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}$, we obtain the final result. □

Lemma C.3.2. *For any comparator $\boldsymbol{\beta} \in D_{\boldsymbol{\beta}}$, the iterates $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$ of Algorithm 3 satisfy the following regret bound:*

$$
\mathbb{E}\left[\sum_{t=1}^T\langle\boldsymbol{\beta}^* - \boldsymbol{\beta}_t, \boldsymbol{\kappa}_{\beta,t}\rangle\right] \leq \frac{2D_{\boldsymbol{\beta}}^2}{\zeta} + \frac{3\zeta T(1 + (1+\gamma^2)D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2)\operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})}{2}.
$$

Proof. We again employ stochastic gradient descent analysis. We first prove that the gradient estimator for $\boldsymbol{\beta}$ is unbiased. Recalling the defini-

tion of $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}$ from Equation (7.9),

$$\begin{aligned}
\mathbb{E}\left[\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} \mid \mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] &= \mathbb{E}\left[\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t\left(R_t + \gamma v_t(X_t') - \langle \boldsymbol{\varphi}_t, \boldsymbol{\theta}_t\rangle\right) \mid \mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= \boldsymbol{\Lambda}_\mu^{c-1}(\mathbb{E}_t[\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\top]\boldsymbol{\theta}_r + \gamma\mathbb{E}_t[\boldsymbol{\varphi}_t v_t(X_t')] - \mathbb{E}_t[\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\top]\boldsymbol{\theta}_t) \\
&= \boldsymbol{\Lambda}_\mu^{c-1}(\boldsymbol{\Lambda}_\mu\boldsymbol{\theta}_r + \gamma\mathbb{E}_t[\boldsymbol{\varphi}_t v_t(X_t')] - \boldsymbol{\Lambda}_\mu\boldsymbol{\theta}_t) \\
&= \boldsymbol{\Lambda}_\mu^{c-1}(\boldsymbol{\Lambda}_\mu\boldsymbol{\theta}_r + \gamma\mathbb{E}_t[\boldsymbol{\varphi}_t\mathbf{P}(\cdot|X_t, A_t)\mathbf{v}_t] - \boldsymbol{\Lambda}_\mu\boldsymbol{\theta}_t) \\
&= \boldsymbol{\Lambda}_\mu^{c-1}(\boldsymbol{\Lambda}_\mu\boldsymbol{\theta}_r + \gamma\mathbb{E}_t[\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\top]\boldsymbol{\Psi}\mathbf{v}_t - \boldsymbol{\Lambda}_\mu\boldsymbol{\theta}_t) \\
&= \boldsymbol{\Lambda}_\mu^{c}(\boldsymbol{\theta}_r + \gamma\boldsymbol{\Psi}v_{\boldsymbol{\theta}_t,\pi_t} - \boldsymbol{\theta}_t) = \boldsymbol{\kappa}_{\beta,t},
\end{aligned}$$

recalling that $\mathbf{v}_t = \mathbf{v}_{\boldsymbol{\theta}_t,\pi_t}$. Next, we bound its second moment. We use the fact that $r \in [0,1]$ and $\|\mathbf{v}_t\|_\infty \leq \|\boldsymbol{\Phi}\boldsymbol{\theta}_t\|_\infty \leq D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}}$ to show that

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}\|_2^2 \,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] &= \mathbb{E}\left[\left\|\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t[R_t + \gamma v_t(X_t') - \langle\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle]\right\|_2^2 |\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&\leq 3(1 + (1+\gamma^2)D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2)\mathbb{E}_t\left[\boldsymbol{\varphi}_t^\mathsf{T}\boldsymbol{\Lambda}_\mu^{2(c-1)}\boldsymbol{\varphi}_t\right] \\
&= 3(1 + (1+\gamma^2)D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2)\mathbb{E}_t\left[\mathrm{Tr}(\boldsymbol{\Lambda}_\mu^{2(c-1)}\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\mathsf{T})\right] \\
&= 3(1 + (1+\gamma^2)D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2)\,\mathrm{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1}).
\end{aligned}$$

Thus, we can apply Lemma D.0.1 with the latter expression as G^2, $\mathbb{B}(D_{\boldsymbol{\beta}})$ as the domain, and ζ as the learning rate. □

Lemma C.3.3. *The sequence of policies $\pi_1, \ldots, \pi_T$ of Algorithm 3 satisfies the following regret bound:*

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{x\in\mathcal{X}}\nu^{\pi^*}(x)\sum_a(\pi^*(a|x) - \pi_t(a|x))\langle\boldsymbol{\theta}_t, \boldsymbol{\varphi}(x,a)\rangle\right] \leq \frac{\log|\mathcal{A}|}{\alpha} + \frac{\alpha T D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2}{2}.$$

Proof. We just apply mirror descent analysis, invoking Lemma D.0.2 with $q_t = \Phi\boldsymbol{\theta}_t$, noting that $\|q_t\|_\infty \leq D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}}$. The proof is concluded by trivially bounding the relative entropy as $\mathcal{H}\left(\pi^*\|\pi_1\right) = \mathbb{E}_{x\sim\nu^{\pi^*}}\left[\mathcal{D}\left(\pi(\cdot|x)\|\pi_1(\cdot|x)\right)\right] \leq \log|\mathcal{A}|$. □

Appendix D

Auxiliary Lemmas

The following is a standard result in convex optimization proved here for the sake of completeness—we refer to Nemirovski and Yudin (1983); Zinkevich (2003); Orabona (2019) for more details and comments on the history of this result.

Lemma D.0.1 (Online Stochastic Gradient Descent)**.** *Given $y_1 \in \mathbb{B}(D_y)$ and $\eta > 0$, define the sequences $y_2, \cdots, y_{n+1}$ and $h_1, \cdots, h_n$ such that for $k = 1, \cdots, n$,*

$$y_{k+1} = \Pi_{\mathbb{B}(D_y)} \left(y_k + \eta \hat{h}_k \right),$$

and $\hat{h}_k$ satisfies $\mathbb{E}\left[\hat{h}_k \,|\mathcal{F}_{k-1}\right] = h_k$ and $\mathbb{E}\left[\left\|\hat{h}_k\right\|_2^2 |\mathcal{F}_{k-1}\right] \leq G^2$. Then, for $y^* \in \mathbb{B}(D_y)$:

$$\mathbb{E}\left[\sum_{k=1}^{n} \langle y^* - y_k, h_k \rangle\right] \leq \frac{\|y_1 - y^*\|_2^2}{2\eta} + \frac{\eta n G^2}{2}.$$

Proof. We start by studying the following term:

$$\begin{aligned}\|y_{k+1} - y^*\|_2^2 &= \left\|\Pi_{\mathbb{B}(D_y)}(y_k + \eta\hat{h}_k) - y^*\right\|_2^2 \\ &\leq \left\|y_k + \eta\hat{h}_k - y^*\right\|_2^2 \\ &= \|y_k - y^*\|_2^2 - 2\eta\left\langle y^* - y_k, \hat{h}_k\right\rangle + \eta^2\left\|\hat{h}_k\right\|_2^2.\end{aligned}$$

The inequality is due to the fact that the projection operator is a non-expansion with respect to the Euclidean norm. Since $\mathbb{E}\left[\hat{h}_k \,|\mathcal{F}_{k-1}\right] = h_k$, we can rearrange the above equation and take a conditional expectation to obtain

$$\begin{aligned}\langle y^* - y_k, h_k\rangle &\leq \frac{\|y_k - y^*\|_2^2 - \mathbb{E}\left[\|y_{k+1} - y^*\|_2^2 \,|\mathcal{F}_{k-1}\right]}{2\eta} + \frac{\eta}{2}\mathbb{E}\left[\left\|\hat{h}_k\right\|_2^2 |\mathcal{F}_{k-1}\right] \\ &\leq \frac{\|y_k - y^*\|_2^2 - \mathbb{E}\left[\|y_{k+1} - y^*\|_2^2 \,|\mathcal{F}_{k-1}\right]}{2\eta} + \frac{\eta G^2}{2},\end{aligned}$$

where the last inequality is from $\mathbb{E}\left[\left\|\hat{h}_k\right\|_2^2 |\mathcal{F}_{k-1}\right] \leq G^2$. Finally, taking a sum over $k = 1, \cdots, n$, taking a marginal expectation, evaluating the resulting telescoping sum and upper-bounding negative terms by zero we obtain the desired result as

$$\begin{aligned}\mathbb{E}\left[\sum_{k=1}^{n}\left\langle y^* - y_k, \hat{h}_k\right\rangle\right] &\leq \frac{\|y_1 - y^*\|_2^2 - \mathbb{E}\left[\|y_{n+1} - y^*\|_2^2\right]}{2\eta} + \frac{\eta}{2}\sum_{k=1}^{n} G^2 \\ &\leq \frac{\|y_1 - y^*\|_2^2}{2\eta} + \frac{\eta n G^2}{2}.\end{aligned}$$

□

The next result is a similar regret analysis for mirror descent with the relative entropy as its distance generating function. Once again, this result is standard, and we refer the interested reader to Nemirovski and Yudin (1983); Cesa-Bianchi and Lugosi (2006); Orabona (2019) for more details. For the analysis, we recall that $\mathcal{D}$ denotes the relative entropy (or Kullback–Leibler divergence), defined for any $p, q \in \Delta(\mathcal{A})$ as $\mathcal{D}(p\|q) =$

$\sum_a p(a) \log \frac{p(a)}{q(a)}$, and that, for any two policies π, π', we define the conditional entropy[1] $\mathcal{H}(\pi \| \pi') \doteq \sum_{x \in \mathcal{X}} \nu^{\pi}(x) \mathcal{D}(\pi(\cdot|x) \| \pi'(\cdot|x))$.

Lemma D.0.2 (Mirror Descent)**.** *Let $q_t, \dots, q_T$ be a sequence of functions from $\mathcal{X} \times \mathcal{A}$ to $\mathbb{R}$ so that $\|q_t\|_\infty \le D_q$ for $t = 1, \dots, T$. Given an initial policy π_1 and a learning rate $\alpha > 0$, define the sequence of policies $\pi_2, \dots, \pi_{T+1}$ such that, for $t = 1, \dots, T$:*

$$\pi_{t+1}(a|x) \propto \pi_t e^{\alpha q_t(x,a)}.$$

Then, for any comparator policy π^*:

$$\sum_{t=1}^{T} \sum_{x \in \mathcal{X}} \nu^{\pi^*}(x) \langle \pi^*(\cdot|x) - \pi_t(\cdot|x), q_t(x, \cdot) \rangle \le \frac{\mathcal{H}(\pi^* \| \pi_1)}{\alpha} + \frac{\alpha T D_q^2}{2}.$$

Proof. We begin by studying the relative entropy between $\pi^*(\cdot|x)$ and iterates $\pi_t(\cdot|x), \pi_{t+1}(\cdot|x)$ for any $x \in \mathcal{X}$:

$$\begin{aligned}
&\mathcal{D}(\pi^*(\cdot|x) \| \pi_{t+1}(\cdot|x)) = \mathcal{D}(\pi^*(\cdot|x) \| \pi_t(\cdot|x)) - \sum_{a \in \mathcal{A}} \pi^*(a|x) \log \frac{\pi_{t+1}(a|x)}{\pi_t(a|x)} \\
&= \mathcal{D}(\pi^*(\cdot|x) \| \pi_t(\cdot|x)) - \sum_{a \in \mathcal{A}} \pi^*(a|x) \log \frac{e^{\alpha q_t(x,a)}}{\sum_{a' \in \mathcal{A}} \pi_t(a'|x) e^{\alpha q_t(x,a')}} \\
&= \mathcal{D}(\pi^*(\cdot|x) \| \pi_t(\cdot|x)) - \alpha \langle \pi^*(\cdot|x), q_t(x, \cdot) \rangle + \log \sum_{a \in \mathcal{A}} \pi_t(a|x) e^{\alpha q_t(x,a)} \\
&= \mathcal{D}(\pi^*(\cdot|x) \| \pi_t(\cdot|x)) - \alpha \langle \pi^*(\cdot|x) - \pi_t(\cdot|x), q_t(x, \cdot) \rangle \\
&\quad + \log \sum_{a \in \mathcal{A}} \pi_t(a|x) e^{\alpha q_t(x,a)} - \alpha \sum_{a \in \mathcal{A}} \pi_t(a|x) q_t(x, a) \\
&\le \mathcal{D}(\pi^*(\cdot|x) \| \pi_t(\cdot|x)) - \alpha \langle \pi^*(\cdot|x) - \pi_t(\cdot|x), q_t(x, \cdot) \rangle + \frac{\alpha^2 \|q_t(x, \cdot)\|_\infty^2}{2}
\end{aligned}$$

where the last inequality follows from Hoeffding's lemma (cf. Lemma A.1 in (Cesa-Bianchi and Lugosi 2006)). Next, we rearrange the above equation, sum over $t = 1, \cdots, T$, evaluate the resulting telescoping sum and

[1] Technically speaking, this quantity is the conditional entropy between the occupancy measures p^{π} and $p^{\pi'}$. We will continue to use this relatively imprecise terminology to keep our notation light, and we refer to Neu, Jonsson, and Gómez (2017) and Bas-Serrano et al. (2021) for more details.

upper-bound negative terms by zero to obtain

$$\sum_{t=1}^{T} \langle \pi^*(\cdot|x) - \pi_t(\cdot|x), q_t(x,\cdot)\rangle \leq \frac{\mathcal{D}\left(\pi^*(\cdot|x)\|\pi_1(\cdot|x)\right)}{\alpha} + \frac{\alpha \left\|q_t(x,\cdot)\right\|_\infty^2}{2}.$$

Finally, using that $\|q_t\|_\infty \leq D_q$ and taking an expectation with respect to $x \sim \nu^{\pi^*}$ concludes the proof. □

Appendix E

Details for the Average-Reward MDP Setting

This section provides the detailed adaptation of our contributions to the average-reward MDPs (AMDPs). In the offline reinforcement learning setting that we consider, we assume access to a sequence of data points (X_t, A_t, R_t, X'_t) in round t generated by a behaviour policy π_B whose occupancy measure is denoted as $\mathbf{p}_B$. Specifically, we will now draw i.i.d. samples from the *undiscounted* occupancy measure as $X_t, A_t \sim \mathbf{p}^\mu$, sample $X'_t \sim P(\cdot|X_t, A_t)$, and compute immediate rewards as $R_t = r(X_t, A_t)$. For simplicity, we use the shorthand notation $\boldsymbol{\varphi}_t = \varphi(X_t, A_t)$ to denote the feature vector drawn in round t.

Before describing our contributions, some definitions are in order. An important central concept in the theory of AMDPs is that of the *relative*

value functions of policy π defined as

$$v^\pi(x) = \lim_{T\to\infty} \mathbb{E}_\pi \left[\sum_{t=0}^{T} r(X_t, A_t) - \rho^\pi \middle| X_0 = x \right],$$
$$q^\pi(x,a) = \lim_{T\to\infty} \mathbb{E}_\pi \left[\sum_{t=0}^{T} r(X_t, A_t) - \rho^\pi \middle| X_0 = x, A_0 = a \right],$$

where we recalled the notation ρ^π denoting the average reward of policy π from the main text. These functions are sometimes also called the *bias functions*, and their intuitive role is to measure the total amount of reward gathered by policy π before it hits its stationary distribution. For simplicity, we will refer to these functions as value functions and action-value functions below.

By their recursive nature, these value functions are also characterized by the corresponding Bellman equations recalled below for completeness

$$\mathbf{q}^\pi = \mathbf{r} - \rho^\pi \mathbf{1} + \mathbf{P}\mathbf{v}^\pi,$$

where $\mathbf{v}^\pi$ is related to the action-value function as $v^\pi(x) = \sum_a \pi(a|x) q^\pi(x,a)$. We note that the Bellman equations only characterize the value functions up to a constant offset. That is, for any policy π, and constant $c \in \mathbb{R}$, $\mathbf{v}^\pi + c\mathbf{1}$ and $\mathbf{q}^\pi + c\mathbf{1}$ also satisfy the Bellman equations. A key quantity to measure the size of the value functions is the *span seminorm* defined for $\mathbf{q} \in \mathbb{R}^{\mathcal{X}\times\mathcal{A}}$ as $\|\mathbf{q}\|_{\text{sp}} = \sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} q(x,a) - \inf_{(x,a)\in\mathcal{X}\times\mathcal{A}} q(x,a)$. Using this notation, the condition of Assumption 7.4.1 can be simply stated as requiring $\|\mathbf{q}^\pi\|_{\text{sp}} \le D_q$ for all π.

Now, let π^* denote an optimal policy with maximum average reward and introduce the shorthand notations $\rho^* = \rho^{\pi^*}, \mathbf{p}^* = \mathbf{p}^{\pi^*}, \boldsymbol{\nu}^* = \boldsymbol{\nu}^{\pi^*}, \mathbf{v}^* = \mathbf{v}^{\pi^*}$ and $\mathbf{q}^* = \mathbf{q}^{\pi^*}$. Under mild assumptions on the MDP that we will clarify shortly, the following Bellman optimality equations are known to characterize bias vectors corresponding to the optimal policy

$$\mathbf{q}^* = \mathbf{r} - \rho^* \mathbf{1} + \mathbf{P}\mathbf{v}^*,$$

where $\mathbf{v}^*$ satisfies $v^*(x) = \max_a q^*(x,a)$. Once again, shifting the solutions by a constant preserves the optimality conditions. It is easy to see that

such constant offsets do not influence greedy or softmax policies extracted from the action value functions. Importantly, by a calculation analogous to Equation (7.2), the action-value functions are exactly realizable under the linear MDP condition (see Definition 7.1.1) and Assumption 7.4.2.

Besides the Bellman optimality equations stated above, optimal policies can be equivalently characterized via the following linear program:

$$\begin{aligned} \text{maximize} \quad & \langle \mathbf{p}, \mathbf{r} \rangle \\ \text{subject to} \quad & \mathbf{E}^\intercal \mathbf{p} = \mathbf{P}^\intercal \mathbf{p} \\ & \langle \mathbf{p}, \mathbf{1} \rangle = 1 \\ & \mathbf{p} \geq 0. \end{aligned} \tag{E.1}$$

This can be seen as the generalization of the LP stated for discounted MDPs in the main text, with the added complication that we need to make sure that the occupancy measures are normalized[1] to 1. By following the same steps as in the main text to relax the constraints and reparametrize the LP, one can show that solutions of the LP under the linear MDP assumption can be constructed by finding the saddle point of the following Lagrangian:

$$\begin{aligned} \mathfrak{L}(\boldsymbol{\lambda}, \mathbf{p}; \rho, \mathbf{v}, \boldsymbol{\theta}) &= \rho + \langle \boldsymbol{\lambda}\,, \boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v} - \boldsymbol{\theta} - \rho\boldsymbol{\varrho} \rangle + \langle \mathbf{u}\,, \boldsymbol{\Phi}\boldsymbol{\theta} - \mathbf{E}\mathbf{v} \rangle \\ &= \rho[1 - \langle \boldsymbol{\lambda}, \boldsymbol{\varrho} \rangle] + \langle \boldsymbol{\theta}, \boldsymbol{\Phi}^\intercal \mathbf{p} - \boldsymbol{\lambda} \rangle + \langle \mathbf{v}, \boldsymbol{\Psi}^\intercal \boldsymbol{\lambda} - \mathbf{E}^\intercal \mathbf{p} \rangle\,. \end{aligned}$$

As before, the optimal value functions $\mathbf{q}^*$ and $\mathbf{v}^*$ are optimal primal variables for the saddle-point problem, as are all of their constant shifts. Thus, the existence of a solution with small span seminorm implies the existence of a solution with small supremum norm.

Finally, applying the same reparametrization $\boldsymbol{\beta} = \boldsymbol{\Lambda}_\mu^{-c} \boldsymbol{\lambda}$ as in the discounted setting, we arrive to the following Lagrangian that forms the basis of our algorithm:

$$\mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}; \rho, \mathbf{v}, \boldsymbol{\theta}) = \rho + \langle \boldsymbol{\beta}\,, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v} - \boldsymbol{\theta} - \rho\boldsymbol{\varrho}] \rangle + \langle \mathbf{p}\,, \boldsymbol{\Phi}\boldsymbol{\theta} - \mathbf{E}\mathbf{v} \rangle.$$

We will aim to find the saddle point of this function via primal-dual methods. As we have some prior knowledge of the optimal solutions, we will

[1] This is necessary because of the absence of ν_0 in the LP, which would otherwise fix the scale of the solutions.

restrict the search space of each optimization variable to nicely chosen compact sets. For the $\boldsymbol{\beta}$ iterates, we consider the Euclidean ball domain $\mathbb{B}(D_{\boldsymbol{\beta}}) = \{\boldsymbol{\beta} \in \mathbb{R}^d \mid \|\boldsymbol{\beta}\|_2 \leq D_{\boldsymbol{\beta}}\}$ with the bound $D_{\boldsymbol{\beta}} > \|\boldsymbol{\Phi}^\intercal \mathbf{p}^*\|_{\boldsymbol{\Lambda}_\mu^{-2c}}$. Since the average reward of any policy is bounded in $[0, 1]$, we naturally restrict the ρ iterates to this domain. Finally, keeping in mind that Assumption 7.4.1 guarantees that $\|\mathbf{q}^\pi\|_{\text{sp}} \leq D_q$, we will also constrain the $\boldsymbol{\theta}$ iterates to an appropriate domain: $\mathbb{B}(D_{\boldsymbol{\theta}}) = \{\boldsymbol{\theta} \in \mathbb{R}^d \mid \|\boldsymbol{\theta}\|_2 \leq D_{\boldsymbol{\theta}}\}$. We will assume that this domain is large enough to represent all action-value functions, which implies that $D_{\boldsymbol{\theta}}$ should scale at least linearly with D_q. Indeed, we will suppose that the features are bounded as $\|\boldsymbol{\varphi}(x, a)\|_2 \leq D_{\boldsymbol{\varphi}}$ for all $(x, a) \in \mathcal{X} \times \mathcal{A}$ so that our optimization algorithm only admits parametric $\mathbf{q}$ functions satisfying $\|\mathbf{q}\|_\infty \leq D_{\boldsymbol{\varphi}} D_{\boldsymbol{\theta}}$. Obviously, $D_{\boldsymbol{\theta}}$ needs to be set large enough to ensure that it is possible at all to represent $\mathbf{q}$-functions with span D_q.

Thus, we aim to solve the following constrained optimization problem:

$$\min_{\rho \in [0,1], \mathbf{v} \in \mathbb{R}^{\mathcal{X}}, \boldsymbol{\theta} \in \mathbb{B}(D_{\boldsymbol{\theta}})} \max_{\boldsymbol{\beta} \in \mathbb{B}(D_{\boldsymbol{\beta}}), \mathbf{p} \in \mathbb{R}_+^{\mathcal{X} \times \mathcal{A}}} \mathfrak{L}(\boldsymbol{\beta}, \mathbf{p}; \rho, \mathbf{v}, \boldsymbol{\theta}).$$

As done in the main text, we eliminate the high-dimensional variables $\mathbf{v}$ and $\mathbf{p}$ by committing to the choices $\mathbf{v} = \mathbf{v}_{\boldsymbol{\theta},\pi}$ and $\mathbf{p} = \mathbf{p}_{\boldsymbol{\beta},\pi}$ defined as

$$\begin{aligned} v_{\boldsymbol{\theta},\pi}(x) &= \sum_a \pi(a|x) \left\langle \boldsymbol{\theta}, \boldsymbol{\varphi}(x, a) \right\rangle, \\ p_{\boldsymbol{\beta},\pi}(x, a) &= \pi(a|x) \langle \boldsymbol{\psi}(x), \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta} \rangle. \end{aligned}$$

This makes it possible to express the Lagrangian in terms of only $\boldsymbol{\beta}, \pi, \rho$ and $\boldsymbol{\theta}$:

$$\begin{aligned} f(\boldsymbol{\beta}, \pi; \rho, \boldsymbol{\theta}) &= \rho + \langle \boldsymbol{\beta}, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_{\boldsymbol{\theta},\pi} - \boldsymbol{\theta} - \rho \boldsymbol{\varrho}] \rangle + \langle \mathbf{p}_{\boldsymbol{\beta},\pi}, \boldsymbol{\Phi} \boldsymbol{\theta} - \mathbf{E} \mathbf{v}_{\boldsymbol{\theta},\pi} \rangle \\ &= \rho + \langle \boldsymbol{\beta}, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_{\boldsymbol{\theta},\pi} - \boldsymbol{\theta} - \rho \boldsymbol{\varrho}] \rangle \end{aligned}$$

The remaining low-dimensional variables $\boldsymbol{\beta}, \rho, \boldsymbol{\theta}$ are then updated using stochastic gradient descent/ascent. For this purpose it is useful to express the partial derivatives of the Lagrangian with respect to said vari-

ables:

$$\begin{aligned}\boldsymbol{\kappa}_{\boldsymbol{\beta}} &= \boldsymbol{\Lambda}_{\mu}^{c}[\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v}_{\boldsymbol{\theta},\pi} - \boldsymbol{\theta} - \rho\boldsymbol{\varrho}] \\ \kappa_{\rho} &= 1 - \left\langle \boldsymbol{\beta}, \boldsymbol{\Lambda}_{\mu}^{c}\boldsymbol{\varrho} \right\rangle \\ \boldsymbol{\kappa}_{\boldsymbol{\theta}} &= \boldsymbol{\Phi}^{\intercal}\mathbf{p}_{\boldsymbol{\beta},\pi} - \boldsymbol{\Lambda}_{\mu}^{c}\boldsymbol{\beta}\end{aligned}$$

E.1 Algorithm for average-reward MDPs

Our algorithm for the AMDP setting has the same double-loop structure as the one for the discounted setting. In particular, the algorithm performs a sequence of outer updates $t = 1, 2, \ldots, T$ on the policies π_t and the occupancy ratiositerates $\boldsymbol{\beta}_t$, and then performs a sequence of updates $i = 1, 2, \ldots, K$ in the inner loop to evaluate the policies and produce $\boldsymbol{\theta}_t$, ρ_t and $\mathbf{v}_t$. Thanks to the reparametrization $\boldsymbol{\beta} = \boldsymbol{\Lambda}_{\mu}^{-c}\boldsymbol{\lambda}$, fixing $\pi_t = \text{softmax}(\sum_{k=1}^{t-1} \boldsymbol{\Phi}\boldsymbol{\theta}_k)$, $\mathbf{v}_t(x) = \sum_{a\in\mathcal{A}} \pi_t(a|x) \langle \varphi(x,a), \boldsymbol{\theta}_t \rangle$ for $x \in \mathcal{X}$, and $p_t(x,a) = \pi_t(a|x) \left\langle \boldsymbol{\psi}(x), \boldsymbol{\Lambda}_{\mu}^{c}\boldsymbol{\beta}_t \right\rangle$ in round t we can obtain unbiased estimates of the gradients of f with respect to $\boldsymbol{\theta}$, $\boldsymbol{\beta}$, and ρ. For each primal update t, the algorithm uses a single sample transition (X_t, A_t, R_t, X_t') generated by the behavior policy π_B to compute an unbiased estimator of the first gradient $\kappa_{\boldsymbol{\beta}}$ for that round as $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} = \boldsymbol{\Lambda}_{\mu}^{c-1}\boldsymbol{\varphi}_t[R_t + v_t(X_t') - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t \rangle - \rho_t]$. Then, in iteration $i = 1, \cdots, K$ of the inner loop within round t, we sample transitions $(X_{t,i}, A_{t,i}, R_{t,i}, X_{t,i}')$ to compute gradient estimators with respect to ρ and $\boldsymbol{\theta}$ as:

$$\begin{aligned}\tilde{g}_{\rho,t,i} &= 1 - \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_{\mu}^{c-1}\boldsymbol{\beta}_t \right\rangle \\ \tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i} &= \boldsymbol{\varphi}_{t,i}' \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_{\mu}^{c-1}\boldsymbol{\beta}_t \right\rangle - \boldsymbol{\varphi}_{t,i} \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_{\mu}^{c-1}\boldsymbol{\beta}_t \right\rangle .\end{aligned}$$

We have used the shorthand notation $\boldsymbol{\varphi}_{t,i} = \varphi(X_{t,i}, A_{t,i})$, $\boldsymbol{\varphi}_{t,i}' = \varphi(X_{t,i}', A_{t,i}')$. The update steps are detailed in the pseudocode presented as Algorithm 4.

We now state the general form of our main result for this setting in Theorem E.1.1 below.

Theorem E.1.1. *Consider a linear MDP (Definition 7.1.1) such that* $\boldsymbol{\theta}^{\pi} \in \mathbb{B}(D_{\boldsymbol{\theta}})$ *for all* $\pi \in \Pi$. *Further, suppose that* $C_{\varphi,c}(\pi^*) \leq D_{\boldsymbol{\beta}}$. *Then, for*

any comparator policy $\pi^* \in \Pi$*, the policy output by Algorithm 4 satisfies:*

$$\mathbb{E}\left[\langle \mathbf{p}^{\pi^*} - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r}\rangle\right] \leq \frac{2D_{\boldsymbol{\beta}}^2}{\zeta T} + \frac{\log|\mathcal{A}|}{\alpha T} + \frac{1}{2\xi K} + \frac{2D_{\boldsymbol{\theta}}^2}{\eta K} + \frac{\zeta G_{\boldsymbol{\beta},c}^2}{2} + \frac{\alpha D_{\boldsymbol{\theta}}^2 D_{\boldsymbol{\varphi}}^2}{2} + \frac{\xi G_{\rho,c}^2}{2} + \frac{\eta G_{\boldsymbol{\theta},c}^2}{2},$$

where

$$G_{\boldsymbol{\beta},c}^2 = \mathrm{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1 + 2D_{\boldsymbol{\theta}} D_{\boldsymbol{\varphi}})^2, \tag{E.2}$$

$$G_{\rho,c}^2 = 2\left(1 + D_{\boldsymbol{\beta}}^2 \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}\right), \tag{E.3}$$

$$G_{\boldsymbol{\theta},c}^2 = 4D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\beta}}^2 \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}. \tag{E.4}$$

In particular, using learning rates $\zeta = \frac{2D_{\boldsymbol{\beta}}}{G_{\boldsymbol{\beta},c}\sqrt{T}}$, $\alpha = \frac{\sqrt{2\log|\mathcal{A}|}}{D_{\boldsymbol{\theta}} D_{\boldsymbol{\varphi}}\sqrt{T}}$, $\xi = \frac{1}{G_{\rho,c}\sqrt{K}}$, *and* $\eta = \frac{2D_{\boldsymbol{\theta}}}{G_{\boldsymbol{\theta},c}\sqrt{K}}$*, and setting* $K = T \cdot \frac{4D_{\boldsymbol{\beta}^2} G_{\boldsymbol{\beta},c}^2 + 2D_{\boldsymbol{\theta}}^2 D_{\boldsymbol{\varphi}}^2 \log|\mathcal{A}|}{G_{\rho,c}^2 + 4D_{\boldsymbol{\theta}}^2 G_{\boldsymbol{\theta},c}^2}$*, we achieve* $\mathbb{E}\left[\langle \mathbf{p}^{\pi^*} - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r}\rangle\right] \leq \epsilon$ *with a number of samples* n_ϵ *that is*

$$O\left(\epsilon^{-4} D_{\boldsymbol{\theta}}^4 D_{\boldsymbol{\varphi}}^4 D_{\boldsymbol{\beta}}^4 \,\mathrm{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1}) \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2(2c-1)} \log|\mathcal{A}|\right).$$

By Remark 7.2.2, we have that n_ϵ is of order

$$O\left(\varepsilon^{-4} D_{\boldsymbol{\theta}}^4 D_{\boldsymbol{\varphi}}^{12c-2} D_{\boldsymbol{\beta}}^4 d^{2-2c} \log|\mathcal{A}|\right).$$

Corollary E.1.2. *Assume that the bound of the feature vectors* $D_{\boldsymbol{\varphi}}$ *is of order* $O(1)$*, that* $D_{\boldsymbol{\theta}_r} = D_{\boldsymbol{\psi}} = \sqrt{d}$ *which together imply* $D_{\boldsymbol{\theta}} \leq \sqrt{d} + 1 + \sqrt{d} D_q = O(\sqrt{d} D_q)$ *and that* $D_{\boldsymbol{\beta}} = c \cdot C_{\varphi,c}(\pi^*)$ *for some positive universal constant* c*. Then, under the same assumptions of Theorem 7.2.1,* n_ε *is of order* $O\left(\varepsilon^{-4} D_q^4 C_{\varphi,c}(\pi^*)^2 d^{4-2c} \log|\mathcal{A}|\right)$.

Recall that $C_{\varphi,1/2}$ is always smaller than $C_{\varphi,1}$, but using $c = 1/2$ in the algorithm requires knowledge of the covariance matrix $\boldsymbol{\Lambda}_\mu$, and results in a slightly worse dependence on the dimension.

The proof of Theorem E.1.1 mainly follows the same steps as in the discounted case, with some added difficulty that is inherent in the more challenging average-reward setup. Some key challenges include treating the additional optimization variable ρ and coping with the fact that the optimal parameters $\boldsymbol{\theta}^*$ and $\boldsymbol{\beta}^*$ are not necessarily unique any more.

E.2 Analysis

We now prove our main result regarding the AMDP setting in Theorem E.1.1. Following the derivations in the main text, we study the dynamic duality gap defined as

$$\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho^*_{1:T}, \boldsymbol{\theta}^*_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}(f(\boldsymbol{\beta}^*, \pi^*; \rho_t, \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \rho^*_t, \boldsymbol{\theta}^*_t)). \quad \text{(E.5)}$$

First we show in Lemma E.2.1 below that, for appropriately chosen comparator points, the expected suboptimality of the policy returned by Algorithm 4 can be upper bounded in terms of the expected dynamic duality gap.

Lemma E.2.1. *Let $\boldsymbol{\theta}^*_t$ such that*

$$\langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}^*_t \rangle = \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}^{\pi_t} \rangle - \inf_{(x,a)\in\mathcal{X}\times\mathcal{A}} \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}^{\pi_t} \rangle$$

holds for all $(x,a) \in \mathcal{X}\times\mathcal{A}$, and let $\mathbf{v}^*_t$ be defined as

$$\mathbf{v}^*_t(x) = \sum_{a\in\mathcal{A}} \pi_t(a|x)\, \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}^*_t \rangle$$

for all x. Also, let $\rho^*_t = \rho^{\pi_t}$, π^* be an optimal policy, and $\boldsymbol{\beta}^* = \boldsymbol{\Lambda}_\mu^{-c}\boldsymbol{\Phi}^\top \mathbf{p}^*$ where $\mathbf{p}^*$ is the occupancy measure of π^*. Then, the suboptimality gap of the policy output by Algorithm 4 satisfies

$$\mathbb{E}_T\left[\langle \mathbf{p}^* - \mathbf{p}^{\boldsymbol{\pi}_{out}}, \mathbf{r}\rangle\right] = \mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho^*_{1:T}, \boldsymbol{\theta}^*_{1:T}).$$

Proof. Substituting $(\boldsymbol{\beta}^*, \pi^*) = (\boldsymbol{\Lambda}_\mu^{-c}\boldsymbol{\Phi}^\top \mathbf{p}^*, \pi^*)$ in the first term of the dynamic duality gap we have

$$\begin{aligned} f(\boldsymbol{\beta}^*, \pi^*; \rho_t, \boldsymbol{\theta}_t) &= \rho_t + \langle \boldsymbol{\Lambda}_\mu^{-c}\boldsymbol{\Phi}^\top \mathbf{p}^*, \boldsymbol{\Lambda}_\mu^{c}[\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v}_{\boldsymbol{\theta}_t,\pi^*} - \boldsymbol{\theta}_t - \rho_t \boldsymbol{\varrho}]\rangle \\ &= \rho_t + \langle \mathbf{p}^*, r + \mathbf{P}\mathbf{v}_{\boldsymbol{\theta}_t,\pi^*} - \boldsymbol{\Phi}\boldsymbol{\theta}_t - \rho_t \mathbf{1}\rangle \\ &= \langle \mathbf{p}^*, r\rangle + \langle \mathbf{p}^*, \mathbf{E}\mathbf{v}_{\boldsymbol{\theta}_t,\pi^*} - \boldsymbol{\Phi}\boldsymbol{\theta}_t\rangle + \rho_t[1 - \langle \mathbf{p}^*, \mathbf{1}\rangle] \\ &= \langle \mathbf{p}^*, r\rangle. \end{aligned}$$

Here, we have used the fact that $\mathbf{p}^*$ is a valid occupancy measure, so it satisfies the flow constraint $\mathbf{E}^\top \mathbf{p}^* = \mathbf{P}^\top \mathbf{p}^*$ and the normalization constraint

$\langle \mathbf{p}^*, \mathbf{1} \rangle = 1$. Also, in the last step we have used the definition of $\mathbf{v}_{\theta_t, \pi^*}$ that guarantees that the following equality holds:

$$\langle \mathbf{p}^*, \mathbf{\Phi} \boldsymbol{\theta}_t \rangle = \sum_{x \in \mathcal{X}} \nu^*(x) \sum_{a \in \mathcal{A}} \pi^*(a|x) \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}(x,a) \rangle = \sum_{x \in \mathcal{X}} \nu^*(x) v_{\boldsymbol{\theta}_t, \pi^*}(x)$$
$$= \langle \mathbf{p}^* , \mathbf{E} \mathbf{v}_{\boldsymbol{\theta}_t, \pi^*} \rangle.$$

For the second term in the dynamic duality gap, using that π_t is $\mathcal{F}_{t-1}$-measurable we write

$$\begin{aligned}
&f(\boldsymbol{\beta}_t, \pi_t; \rho_t^*, \boldsymbol{\theta}_t^*) = \rho_t^* + \langle \boldsymbol{\beta}_t \, , \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \mathbf{\Psi} \mathbf{v}_{\boldsymbol{\theta}_t^*, \pi_t} - \boldsymbol{\theta}_t^* - \rho_t^* \boldsymbol{\varrho}] \rangle \\
&= \rho_t^* + \langle \boldsymbol{\beta}_t \, , \boldsymbol{\Lambda}_\mu^{c-1} \mathbb{E}_t \Big[\boldsymbol{\varphi}_t \boldsymbol{\varphi}_t^\intercal [\boldsymbol{\theta}_r + \mathbf{\Psi} \mathbf{v}_{\boldsymbol{\theta}_t^*, \pi_t} - \boldsymbol{\theta}_t^* - \rho_t^* \boldsymbol{\varrho}] \Big] \rangle \\
&= \rho_t^* + \langle \boldsymbol{\beta}_t \, , \mathbb{E}_t \Big[\boldsymbol{\Lambda}_\mu^{c-1} \boldsymbol{\varphi}_t [R_t \\
&\quad + \sum_{x,a} P(x|X_t, A_t) \pi_t(a|x) \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}_t^* \rangle - \langle \boldsymbol{\varphi}(X_t, A_t), \boldsymbol{\theta}_t^* \rangle - \rho_t^*] \Big] \rangle \\
&= \rho^{\pi_t} + \langle \boldsymbol{\beta}_t \, , \mathbb{E}_t \Big[\boldsymbol{\Lambda}_\mu^{c-1} \boldsymbol{\varphi}_t [R_t \\
&\quad + \sum_{x,a} P(x|X_t, A_t) \pi_t(a|x) \langle \boldsymbol{\varphi}(x,a), \boldsymbol{\theta}^{\pi_t} \rangle - \langle \boldsymbol{\varphi}(X_t, A_t), \boldsymbol{\theta}^{\pi_t} \rangle - \rho^{\pi_t}] \Big] \rangle \\
&= \rho^{\pi_t} + \langle \boldsymbol{\beta}_t \, , \mathbb{E}_t \big[\boldsymbol{\Lambda}_\mu^{c-1} \boldsymbol{\varphi}_t [r(X_t, A_t) + \langle P(\cdot|X_t, A_t), v^{\pi_t} \rangle - q^{\pi_t}(X_t, A_t) - \rho^{\pi_t}] \big] \rangle \\
&= \rho^{\pi_t} = \langle \mathbf{p}^{\pi_t}, r \rangle \, ,
\end{aligned}$$

where in the fourth equality we used that

$$\langle \boldsymbol{\varphi}(x,a) - \boldsymbol{\varphi}(x',a'), \theta_t^* \rangle = \langle \boldsymbol{\varphi}(x,a) - \boldsymbol{\varphi}(x',a'), \theta^{\pi_t} \rangle$$

holds for all x, a, x', a' by definition of θ_t^*. Then, the last equality follows from the fact that the Bellman equations for π_t imply $q^{\pi_t}(x,a) + \rho^{\pi_t} = r(x,a) + \langle P(\cdot|x,a), \mathbf{v}^{\pi_t} \rangle$.

Combining both expressions for $f(\boldsymbol{\beta}^*, \pi^*; \rho_t, \boldsymbol{\theta}_t)$ and $f(\boldsymbol{\beta}_t, \pi_t; \rho_t^*, \boldsymbol{\theta}_t^*)$ in the dynamic duality gap we have:

$$\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho_{1:T}^*, \boldsymbol{\theta}_{1:T}^*) = \frac{1}{T} \sum_{t=1}^{T} (\langle \mathbf{p}^* - \mathbf{p}^{\pi_t}, r \rangle - \rho(\pi_t) [\langle \boldsymbol{\beta}_t \, , \boldsymbol{\Lambda}_\mu \boldsymbol{\varrho} \rangle - 1])$$
$$= \mathbb{E}_T \left[\langle \mathbf{p}^* - \mathbf{p}^{\pi_{\text{out}}}, r \rangle \right].$$

The second equality follows from noticing that, since $\boldsymbol{\pi}_{\text{out}}$ is sampled uniformly from $\{\pi_t\}_{t=1}^T$, $\mathbb{E}\left[\langle \mathbf{p}^{\boldsymbol{\pi}_{\text{out}}}, \mathbf{r}\rangle\right] = \frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\langle \mathbf{p}^{\pi_t}, \mathbf{r}\rangle\right]$. This completes the proof. □

Having shown that for well-chosen comparator points the dynamic duality gap equals the expected suboptimality of the output policy of Algorithm 4, it remains to relate the gap to the optimization error of the primal-dual procedure. This is achieved in the following lemma.

Lemma E.2.2. *For the same choice of comparators* $(\boldsymbol{\beta}^*, \pi^*; \rho^*_{1:T}, \boldsymbol{\theta}^*_{1:T})$ *as in Lemma E.2.1 the dynamic duality gap associated with the iterates produced by Algorithm 4 satisfies*

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho^*_{1:T}, \boldsymbol{\theta}^*_{1:T})\right] \\
&\quad\leq \frac{2D_{\boldsymbol{\beta}}^2}{\zeta T} + \frac{\mathcal{H}\left(\pi^* \| \pi_1\right)}{\alpha T} + \frac{1}{2\xi K} + \frac{2D_{\boldsymbol{\theta}}^2}{\eta K} \\
&\qquad + \frac{\zeta \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1 + 2D_{\boldsymbol{\varphi}} D_{\boldsymbol{\theta}})^2}{2} + \frac{\alpha D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2}{2} + \xi\left(1 + D_{\boldsymbol{\beta}}^2 \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}\right) \\
&\qquad + 2\eta D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\beta}}^2 \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}.
\end{aligned}$$

Proof. The first part of the proof follows from recognising that the dynamic duality gap can be rewritten in terms of the total regret of the primal and dual players in the algorithm. Formally, we write

$$\begin{aligned}
&\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho^*_{1:T}, \boldsymbol{\theta}^*_{1:T}) \\
&\quad= \frac{1}{T}\sum_{t=1}^T \left(f(\boldsymbol{\beta}^*, \pi^*; \rho_t, \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \rho_t, \boldsymbol{\theta}_t)\right) \\
&\quad\quad + \frac{1}{T}\sum_{t=1}^T \left(f(\boldsymbol{\beta}_t, \pi_t; \rho_t, \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \rho^*_t, \boldsymbol{\theta}^*_t)\right).
\end{aligned}$$

Using that $\boldsymbol{\beta}^* = \boldsymbol{\Lambda}_\mu^{-c}\boldsymbol{\Phi}^\top \mathbf{p}^*$, $\mathbf{q}_t = \langle \boldsymbol{\varphi}(x, a), \boldsymbol{\theta}_t\rangle$, $\mathbf{v}_t = \mathbf{v}_{\boldsymbol{\theta}_t, \pi_t}$ and that $\mathbf{g}_{\boldsymbol{\beta}, t} = \boldsymbol{\Lambda}_\mu^c[\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v}_t - \boldsymbol{\theta}_t - \rho_t \boldsymbol{\varrho}]$, we see that term in the first sum can be simply

rewritten as

$$
\begin{aligned}
&f(\boldsymbol{\beta}^*, \pi^*; \rho_t, \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \rho_t, \boldsymbol{\theta}_t) \\
&\quad = \langle \boldsymbol{\beta}^* \,, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_{\boldsymbol{\theta}_t, \pi^*} - \boldsymbol{\theta}_t - \rho_t \boldsymbol{\varrho}] \rangle \\
&\qquad - \langle \boldsymbol{\beta}_t \,, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_{\boldsymbol{\theta}_t, \pi_t} - \boldsymbol{\theta}_t - \rho_t \boldsymbol{\varrho}] \rangle \\
&\quad = \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t \,, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_t - \boldsymbol{\theta}_t - \rho_t \boldsymbol{\varrho}] \rangle + \langle \boldsymbol{\Psi}^\intercal \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}^* \,, \mathbf{v}_{\boldsymbol{\theta}_t, \pi^*} - \mathbf{v}_{\boldsymbol{\theta}_t, \pi_t} \rangle \\
&\quad = \langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t \,, \mathbf{g}_{\boldsymbol{\beta}, t} \rangle + \sum_{x \in \mathcal{X}} \nu^*(x) \, \langle \pi^*(\cdot|x) - \pi_t(\cdot|x), \mathbf{q}_t(x, \cdot) \rangle \,.
\end{aligned}
$$

In a similar way, using that $\mathbf{E}^\intercal \mathbf{p}_t = \boldsymbol{\Psi}^\intercal \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t$ and the definitions of the gradients $\kappa_{\rho,t}$ and $\mathbf{g}_{\boldsymbol{\theta},t}$, the term in the second sum can be rewritten as

$$
\begin{aligned}
&f(\boldsymbol{\beta}_t, \pi_t; \rho_t, \boldsymbol{\theta}_t) - f(\boldsymbol{\beta}_t, \pi_t; \rho_t^*, \boldsymbol{\theta}_t^*) \\
&\quad = \rho_t + \langle \boldsymbol{\beta}_t \,, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_{\boldsymbol{\theta}_t, \pi_t} - \boldsymbol{\theta}_t - \rho_t \boldsymbol{\varrho}] \rangle - \rho_t^* \\
&\qquad - \langle \boldsymbol{\beta}_t \,, \boldsymbol{\Lambda}_\mu^c [\boldsymbol{\theta}_r + \boldsymbol{\Psi} \mathbf{v}_{\boldsymbol{\theta}_t^*, \pi_t} - \boldsymbol{\theta}_t^* - \rho_t^* \boldsymbol{\varrho}] \rangle \\
&\quad = (\rho_t - \rho_t^*)[1 - \left\langle \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\varrho} \right\rangle] - \left\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t \right\rangle + \left\langle \mathbf{E}^\intercal \mathbf{p}_t, \mathbf{v}_{\boldsymbol{\theta}_t, \pi_t} - \mathbf{v}_{\boldsymbol{\theta}_t^*, \pi_t} \right\rangle \\
&\quad = (\rho_t - \rho_t^*)[1 - \left\langle \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\varrho} \right\rangle] - \left\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t \right\rangle + \langle \boldsymbol{\Phi}^\intercal \mathbf{p}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^* \rangle \\
&\quad = (\rho_t - \rho_t^*)[1 - \left\langle \boldsymbol{\beta}_t, \boldsymbol{\Lambda}_\mu^c \boldsymbol{\varrho} \right\rangle] + \left\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \boldsymbol{\Phi}^\intercal \mathbf{p}_t - \boldsymbol{\Lambda}_\mu^c \boldsymbol{\beta}_t \right\rangle \\
&\quad = (\rho_t - \rho_t^*) \kappa_{\rho,t} + \left\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*, \mathbf{g}_{\boldsymbol{\theta},t} \right\rangle \\
&\quad = \frac{1}{K} \sum_{i=1}^{K} \left((\rho_t^{(i)} - \rho_t^*) \kappa_{\rho,t} + \left\langle \boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^*, \mathbf{g}_{\boldsymbol{\theta},t} \right\rangle \right) .
\end{aligned}
$$

Combining both terms in the duality gap concludes the first part of the proof. As shown below the dynamic duality gap is written as the error between iterates of the algorithm from respective comparator points in the direction of the exact gradients. Formally, we have

$$
\begin{aligned}
&\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho_{1:T}^*, \boldsymbol{\theta}_{1:T}^*) = \\
&\quad \frac{1}{T} \sum_{t=1}^{T} \left(\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t \,, \mathbf{g}_{\boldsymbol{\beta},t} \rangle + \sum_{x \in \mathcal{X}} \nu^*(x) \, \langle \pi^*(\cdot|x) - \pi_t(\cdot|x), \mathbf{q}_t(x, \cdot) \rangle \right) \\
&\quad + \frac{1}{TK} \sum_{t=1}^{T} \sum_{i=1}^{K} \left((\rho_t^{(i)} - \rho_t^*) \kappa_{\rho,t} + \left\langle \boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^*, \mathbf{g}_{\boldsymbol{\theta},t} \right\rangle \right) .
\end{aligned}
$$

Then, implementing techniques from stochastic gradient descent analysis in the proof of Lemmas E.3.1 to E.3.3 and mirror descent analysis in Lemma C.3.3, the expected dynamic duality gap can be upper bounded as follows:

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{G}_T(\boldsymbol{\beta}^*, \pi^*; \rho^*_{1:T}, \boldsymbol{\theta}^*_{1:T})\right] \\
&\quad \leq \frac{2D_{\boldsymbol{\beta}}^2}{\zeta T} + \frac{\mathcal{H}\left(\pi^* \| \pi_1\right)}{\alpha T} + \frac{1}{2\xi K} + \frac{2D_{\boldsymbol{\theta}}^2}{\eta K} \\
&\qquad + \frac{\zeta \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1 + 2D_{\boldsymbol{\varphi}} D_{\boldsymbol{\theta}})^2}{2} + \frac{\alpha D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\theta}}^2}{2} \\
&\qquad + \xi\left(1 + D_{\boldsymbol{\beta}}^2 \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}\right) + 2\eta D_{\boldsymbol{\varphi}}^2 D_{\boldsymbol{\beta}}^2 \left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}.
\end{aligned}$$

This completes the proof □

Proof of Theorem E.1.1 First, we bound the expected suboptimality gap by combining Lemmas E.2.1 and E.2.2. Next, bearing in mind that the algorithm only needs $T(K+1)$ total samples from the behavior policy we optimize the learning rates to obtain a bound on the sample complexity, thus completing the proof. □

E.3 Missing proofs for Lemma E.2.2

In this section we prove Lemmas E.3.1 to E.3.3 used in the proof of Lemma E.2.2. It is important to recall that sample transitions (X_k, A_k, R_t, X'_k) in any iteration k are generated in the following way: we draw i.i.d state-action pairs (X_k, A_k) from $\mathbf{p}_B$, and for each state-action pair, the next X'_k is sampled from $P(\cdot|X_k, A_k)$ and immediate reward computed as $R_t = r(X_k, A_k)$. Precisely in iteration i of round t where $k = (t, i)$, since $(X_{t,i}, A_{t,i})$ are sampled i.i.d from $\mathbf{p}_B$ at this time step, $\mathbb{E}_{t,i}\left[\boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\right] = \mathbb{E}_{(x,a)\sim\mathbf{p}_B}\left[\boldsymbol{\varphi}(x,a)\boldsymbol{\varphi}(x,a)^\intercal\right] = \boldsymbol{\Lambda}_\mu$.

Lemma E.3.1. *The gradient estimator* $\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}$ *satisfies* $\mathbb{E}\left[\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} \,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] = \mathbf{g}_{\boldsymbol{\beta},t}$ *and*

$$\mathbb{E}\left[\|\widehat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}\|_2^2\right] \leq \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1 + 2D_{\boldsymbol{\varphi}} D_{\boldsymbol{\theta}})^2.$$

Furthermore, for any $\boldsymbol{\beta}^*$ with $\boldsymbol{\beta}^* \in \mathbb{B}(D_{\boldsymbol{\beta}})$, the iterates $\boldsymbol{\beta}_t$ satisfy

$$\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t\,, \mathbf{g}_{\boldsymbol{\beta},t}\rangle\right] \leq \frac{2D_{\boldsymbol{\beta}}^2}{\zeta} + \frac{\zeta T \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1+2D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}})^2}{2}. \tag{E.6}$$

Proof. For the first part, we remind that π_t is $\mathcal{F}_{t-1}$-measurable and $\mathbf{v}_t$ is determined given π_t and $\boldsymbol{\theta}_t$. Then, we write

$$\begin{aligned}
\mathbb{E}\left[\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] &= \mathbb{E}\left[\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t[R_t + v_t(X_t') - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle - \rho_t]\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= \mathbb{E}\left[\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t[R_t + \mathbb{E}_{x'\sim P(\cdot|X_t,A_t)}\left[v_t(x')\right] - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle - \rho_t]\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= \mathbb{E}\left[\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t[R_t + \langle P(\cdot|X_t, A_t), \mathbf{v}_t\rangle - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle - \rho_t]\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= \mathbb{E}\left[\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\intercal[\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v}_t - \boldsymbol{\theta}_t - \rho_t\boldsymbol{\varrho}]\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= \boldsymbol{\Lambda}_\mu^{c-1}\mathbb{E}\left[\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\intercal\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right][\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v}_t - \boldsymbol{\theta}_t - \rho_t\boldsymbol{\varrho}] \\
&= \boldsymbol{\Lambda}_\mu^{c}[\boldsymbol{\theta}_r + \boldsymbol{\Psi}\mathbf{v}_t - \boldsymbol{\theta}_t - \rho_t\boldsymbol{\varrho}] = \mathbf{g}_{\boldsymbol{\beta},t}.
\end{aligned}$$

Next, we use the facts that $r \in [0,1]$ and $\|\mathbf{v}_t\|_\infty \leq \|\boldsymbol{\Phi}\boldsymbol{\theta}_t\|_\infty \leq D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}}$ to show the following bound:

$$\begin{aligned}
\mathbb{E}\left[\|\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t}\|_2^2\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] &= \mathbb{E}\left[\left\|\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t[R_t + v_t(X_t') - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle]\right\|_2^2\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= \mathbb{E}\left[|R_t + v_t(X_t') - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle|\left\|\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t\right\|_2^2\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&\leq \mathbb{E}\left[(1+2D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}})^2\left\|\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t\right\|_2^2\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= (1+2D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}})^2\mathbb{E}\left[\boldsymbol{\varphi}_t^\intercal\boldsymbol{\Lambda}_\mu^{2(c-1)}\boldsymbol{\varphi}_t\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&= (1+2D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}})^2\mathbb{E}\left[\operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2(c-1)}\boldsymbol{\varphi}_t\boldsymbol{\varphi}_t^\intercal)\,|\mathcal{F}_{t-1}, \boldsymbol{\theta}_t\right] \\
&\leq \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1+2D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}})^2.
\end{aligned}$$

The last step follows from the fact that $\boldsymbol{\Lambda}_\mu$, hence also $\boldsymbol{\Lambda}_\mu^{2c-1}$, is positive semi-definite, so $\operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1}) \geq 0$. Having shown these properties, we appeal to the standard analysis of online gradient descent stated as Lemma D.0.1 to obtain the following bound

$$\mathbb{E}\left[\sum_{t=1}^{T}\langle \boldsymbol{\beta}^* - \boldsymbol{\beta}_t\,, \mathbf{g}_{\boldsymbol{\beta},t}\rangle\right] \leq \frac{\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}^*\|_2^2}{2\zeta} + \frac{\zeta T \operatorname{Tr}(\boldsymbol{\Lambda}_\mu^{2c-1})(1+2D_{\boldsymbol{\varphi}}D_{\boldsymbol{\theta}})^2}{2}.$$

Using that $\|\boldsymbol{\beta}^*\|_2 \leq D_{\boldsymbol{\beta}}$ concludes the proof. □

Lemma E.3.2. *The gradient estimator $\tilde{g}_{\rho,t,i}$ satisfies $\mathbb{E}_{t,i}\left[\tilde{g}_{\rho,t,i}\right] = \kappa_{\rho,t}$ and $\mathbb{E}_{t,i}\left[\tilde{g}^2_{\rho,t,i}\right] \leq 2 + 2D^2_{\boldsymbol{\beta}}\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}$. Furthermore, for any $\rho_t^* \in [0,1]$, the iterates $\rho_t^{(i)}$ satisfy*

$$\mathbb{E}\left[\sum_{i=1}^{K}(\rho_t^{(i)} - \rho_t^*)\kappa_{\rho,t}\right] \leq \frac{1}{2\xi} + \xi K\left(1 + \|\boldsymbol{\beta}_t\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}}\right).$$

Proof. For the first part of the proof, we use that $\boldsymbol{\beta}_t$ is $\mathcal{F}_{t,i-1}$-measurable, to obtain

$$\begin{aligned}
\mathbb{E}_{t,i}\left[\tilde{g}_{\rho,t,i}\right] &= \mathbb{E}_{t,i}\left[1 - \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right] \\
&= \mathbb{E}_{t,i}\left[1 - \left\langle \boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\boldsymbol{\varrho}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right] \\
&= 1 - \left\langle \boldsymbol{\Lambda}_\mu^{c}\boldsymbol{\varrho}, \boldsymbol{\beta}_t\right\rangle = \kappa_{\rho,t}.
\end{aligned}$$

In addition, using Young's inequality and $\|\boldsymbol{\beta}_t\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}} \leq D^2_{\boldsymbol{\beta}}\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}$ we show that

$$\begin{aligned}
\mathbb{E}_{t,i}\left[\tilde{g}^2_{\rho,t,i}\right] &= \mathbb{E}_{t,i}\left[\left(1 - \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right)^2\right] \\
&\leq 2 + 2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] \\
&= 2 + 2\|\boldsymbol{\beta}_t\|^2_{\boldsymbol{\Lambda}_\mu^{2c-1}} \leq 2 + 2D^2_{\boldsymbol{\beta}}\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}.
\end{aligned}$$

For the second part, we appeal to the standard online gradient descent analysis of Lemma D.0.1 to bound on the total error of the iterates:

$$\mathbb{E}\left[\sum_{i=1}^{K}(\rho_t^{(i)} - \rho_t^*)\kappa_{\rho,t}\right] \leq \frac{\left(\rho_t^{(1)} - \rho_t^*\right)^2}{2\xi} + \xi K\left(1 + D^2_{\boldsymbol{\beta}}\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}\right).$$

Using that $\left(\rho_t^{(1)} - \rho_t^*\right)^2 \leq 1$ concludes the proof. □

Lemma E.3.3. *The gradient estimator $\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i}$ satisfies $\mathbb{E}_{t,i}\left[\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i}\right] = \mathbf{g}_{\boldsymbol{\theta},t,i}$ and $\mathbb{E}_{t,i}\left[\|\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i}\|_2^2\right] \leq 4D^2_{\boldsymbol{\varphi}}D^2_{\boldsymbol{\beta}}\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}$. Furthermore, for any $\boldsymbol{\theta}_t^*$ with $\|\boldsymbol{\theta}_t^*\|_2 \leq D_{\boldsymbol{\theta}}$, the iterates $\boldsymbol{\theta}_t^{(i)}$ satisfy*

$$\mathbb{E}\left[\sum_{i=1}^{K}\left\langle \boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^*, \mathbf{g}_{\boldsymbol{\theta},t,i}\right\rangle\right] \leq \frac{2D^2_{\boldsymbol{\theta}}}{\eta} + 2\eta K D^2_{\boldsymbol{\varphi}}D^2_{\boldsymbol{\beta}}\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}. \tag{E.7}$$

Proof. Since $\boldsymbol{\beta}_t, \pi_t, \rho_t^i$ and $\boldsymbol{\theta}_t^i$ are $\mathcal{F}_{t,i-1}$-measurable, we obtain

$$\begin{aligned}
\mathbb{E}_{t,i}\left[\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i}\right] &= \mathbb{E}_{t,i}\left[\boldsymbol{\varphi}'_{t,i}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle - \boldsymbol{\varphi}_{t,i}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right] \\
&= \boldsymbol{\Phi}^\intercal \mathbb{E}_{t,i}\left[\mathbf{e}_{X'_{t,i},A'_{t,i}}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right] - \mathbb{E}_{t,i}\left[\boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\right]\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t \\
&= \boldsymbol{\Phi}^\intercal \mathbb{E}_{t,i}\left[[\pi_t \circ P(\cdot|X_t, A_t)]\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right] - \boldsymbol{\Lambda}_\mu^c\boldsymbol{\beta}_t \\
&= \boldsymbol{\Phi}[\pi_t \circ \boldsymbol{\Psi}^\intercal \mathbb{E}_{t,i}\left[\boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\right]\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t] - \boldsymbol{\Lambda}_\mu^c\boldsymbol{\beta}_t \\
&= \boldsymbol{\Phi}[\pi_t \circ \boldsymbol{\Psi}^\intercal\boldsymbol{\Lambda}_\mu^c\boldsymbol{\beta}_t] - \boldsymbol{\Lambda}_\mu^c\boldsymbol{\beta}_t \\
&= \boldsymbol{\Phi}^\intercal\mathbf{p}_t - \boldsymbol{\Lambda}_\mu^c\boldsymbol{\beta}_t = \mathbf{g}_{\boldsymbol{\theta},t}.
\end{aligned}$$

Next, we consider the squared gradient norm and bound it via elementary manipulations as follows:

$$\begin{aligned}
\mathbb{E}_{t,i}\left[\left\|\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i}\right\|_2^2\right] &= \mathbb{E}_{t,i}\left[\left\|\boldsymbol{\varphi}'_{t,i}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle - \boldsymbol{\varphi}_{t,i}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right\|_2^2\right] \\
&\leq 2\mathbb{E}_{t,i}\left[\left\|\boldsymbol{\varphi}'_{t,i}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right\|_2^2\right] + 2\mathbb{E}_{t,i}\left[\left\|\boldsymbol{\varphi}_{t,i}\left\langle\boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right\rangle\right\|_2^2\right] \\
&= 2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_{t,i}\left\|\boldsymbol{\varphi}'_{t,i}\right\|_2^2\boldsymbol{\varphi}_{t,i}^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] \\
&\quad + 2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_{t,i}\left\|\boldsymbol{\varphi}_{t,i}\right\|_2^2\boldsymbol{\varphi}_{t,i}^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] \\
&\leq 2D_{\boldsymbol{\varphi}}^2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] + 2D_{\boldsymbol{\varphi}}^2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_{t,i}\boldsymbol{\varphi}_{t,i}^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] \\
&= 2D_{\boldsymbol{\varphi}}^2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\Lambda}_\mu\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] + 2D_{\boldsymbol{\varphi}}^2\mathbb{E}_{t,i}\left[\boldsymbol{\beta}_t^\intercal\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\Lambda}_\mu\boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t\right] \\
&\leq 4D_{\boldsymbol{\varphi}}^2\|\boldsymbol{\beta}_t\|_{\boldsymbol{\Lambda}_\mu^{2c-1}}^2 \leq 4D_{\boldsymbol{\varphi}}^2D_{\boldsymbol{\beta}}^2\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}.
\end{aligned}$$

Having verified these conditions, we appeal to the online gradient descent analysis of Lemma D.0.1 to show the bound

$$\mathbb{E}\left[\sum_{i=1}^K\left\langle\boldsymbol{\theta}_t^{(i)} - \boldsymbol{\theta}_t^*, \mathbf{g}_{\boldsymbol{\theta},t}\right\rangle\right] \leq \frac{\left\|\boldsymbol{\theta}_t^{(1)} - \boldsymbol{\theta}_t^*\right\|_2^2}{2\eta} + 2\eta K D_{\boldsymbol{\varphi}}^2D_{\boldsymbol{\beta}}^2\left\|\boldsymbol{\Lambda}_\mu\right\|_2^{2c-1}.$$

We then use that $\left\|\boldsymbol{\theta}_t^* - \boldsymbol{\theta}_t^{(1)}\right\|_2 \leq 2D_{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}_t^*, \boldsymbol{\theta}_t^{(1)} \in \mathbb{B}(D_{\boldsymbol{\theta}})$, thus concluding the proof. □

Input: Learning rates ζ, α,ξ,η, initial iterates $\boldsymbol{\beta}_1 \in \mathbb{B}(D_{\boldsymbol{\beta}})$, $\rho_0 \in [0,1]$, $\boldsymbol{\theta}_0 \in \mathbb{B}(D_{\boldsymbol{\theta}})$, $\pi_1 \in \Pi$

for $t = 1$ ***to*** T **do**

; `// Stochastic gradient descent`

Initialize: $\boldsymbol{\theta}_t^{(1)} = \boldsymbol{\theta}_{t-1}$; **for** $i = 1$ ***to*** K **do**

Obtain sample $W_{t,i} = (X_{t,i}, A_{t,i}, R_{t,i}, X'_{t,i})$;

Sample $A'_{t,i} \sim \pi_t(\cdot|X'_{t,i})$;

Compute $\tilde{g}_{\rho,t,i} = 1 - \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t \right\rangle$;

$\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i} = \boldsymbol{\varphi}'_{t,i} \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t \right\rangle - \boldsymbol{\varphi}_{t,i} \left\langle \boldsymbol{\varphi}_{t,i}, \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\beta}_t \right\rangle$;

Update $\rho_t^{(i+1)} = \Pi_{[0,1]}(\rho_t^{(i)} - \xi\tilde{g}_{\rho,t,i})$;

$\boldsymbol{\theta}_t^{(i+1)} = \Pi_{\mathbb{B}(D_{\boldsymbol{\theta}})}(\boldsymbol{\theta}_t^{(i)} - \eta\tilde{\mathbf{g}}_{\boldsymbol{\theta},t,i})$.

end

Compute $\rho_t = \dfrac{1}{K}\sum_{i=1}^K \rho_t^{(i)}$;

$\boldsymbol{\theta}_t = \dfrac{1}{K}\sum_{i=1}^K \boldsymbol{\theta}_t^{(i)}$;

; `// Stochastic gradient ascent`

Obtain sample $W_t = (X_t, A_t, R_t, X'_t)$;

Compute $v_t(X'_t) = \sum_a \pi_t(a|X'_t)\left\langle \boldsymbol{\varphi}(X'_t, a), \boldsymbol{\theta}_t \right\rangle$;

Compute $\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t} = \boldsymbol{\Lambda}_\mu^{c-1}\boldsymbol{\varphi}_t[R_t + v_t(X'_t) - \langle \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t\rangle - \rho_t]$;

Update $\boldsymbol{\beta}_{t+1} = \Pi_{\mathbb{B}(D_{\boldsymbol{\beta}})}(\boldsymbol{\beta}_t + \zeta\hat{\boldsymbol{\kappa}}_{\boldsymbol{\beta},t})$;

; `// Policy update`

Compute $\pi_{t+1} = \sigma\left(\alpha\sum_{k=1}^t \boldsymbol{\Phi}\boldsymbol{\theta}_k\right)$.

end

Output: π_J with $J \sim \mathcal{U}(T)$

Algorithm 4: Offline primal-dual method for Average-reward MDPs