Imperial College London

Department of Mathematics

Gradient-Based Methods for Optimal Sequential Decision-Making in High-Dimension

by Emmeran Johnson

Submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

February 2026

Statement of Originality

The work presented in this thesis is my own and all work not produced by myself has been appropriately referenced. In particular, my contributions to all work produced from collaborations has also been appropriately referenced. *Emmeran Johnson*

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Abstract

Sequential decision-making encompasses fundamental problems in statistics, optimisation, and reinforcement learning where a learner repeatedly selects actions and receives feedback to improve future decisions. The success of modern large-scale instances of these problems stems largely from gradient-based methods applied in high-dimensional settings, yet the theoretical understanding of these approaches remains limited. This thesis explores gradient-based approaches for high-dimensional sequential decision-making problems across three complementary directions.

First, we establish fundamental separations between low and high dimensional problems, showing that high dimensional problems are intrinsically harder and require different methods. We focus here on online convex optimisation and reinforcement learning.

Second, we show that prohibitively high dimensionality does not always preclude strong guarantees in stochastic shortest path problems. When additional low-dimensional structure, such as sparsity is present, gradient-based methods can be designed to exploit it optimally, replacing dependence on the ambient dimension with dependence on a smaller effective dimension.

Third, we analyse gradient-based reinforcement learning algorithms in simple yet revealing settings. We identify sharp thresholds for the regret of policy gradient methods for bandits and establish optimal, dimension-independent convergence rates for exact policy mirror descent in discounted Markov decision processes.

Across these settings, the thesis provides upper and lower bounds that characterise the limits of performance. Collectively, these results clarify the interplay between dimensionality and gradient-based optimisation, advancing the theoretical foundations of modern sequential decision-making algorithms.

Acknowledgements

A PhD is a degree awarded to an individual but it is ultimately a collective achievement, shaped and sustained by the many people who make the journey unique and successful. Indeed, the journey to completing this PhD has been shaped by a series of serendipitous events. Among the many paths that might have unfolded, I feel fortunate to have followed one that has brought me much satisfaction and happiness, and for that I am deeply grateful to all those involved. A brief story seems appropriate.

In the autumn of 2020, I was studying for a Master's degree in a Covid-stricken Oxford, largely unsure of my plans for the following year. A PhD had been in the back of my mind for some time, but more as a romantic notion than a concrete ambition. One day, rather unexpectedly, Axel, who had supervised a summer research project during my undergraduate studies, emailed me about the StatML PhD programme and suggested that I apply. It sounded interesting, so I did.

At the time, I had limited research experience and no sharply defined interests. I therefore joined the programme under Axel's supervision, working on applied problems in Sequential Monte Carlo. While I enjoyed the work, it gradually became clear that it was not quite the right fit.

Around the same time, Livia, who was completing a Master's in bioengineering at Imperial, was (somewhat randomly) taking a reinforcement learning course and struggling to keep a pole balanced on a cart. Watching this struggle sparked my curiosity. I looked into whether anyone in the mathematics department was working on reinforcement learning, which led me to Ciara. I reached out to her, and she was looking for students to work on a project with Patrick. The rest, as they say, is almost history.

My relationship with Livia also sparked an interest in Italy. When the opportunity arose to do an exchange, this interest led me to Nicolò and the LAILA lab in Milan. My time there was immensely enjoyable and influential. It shifted my research interests away from reinforcement learning and towards online learning, shaped how I experienced and enjoyed conferences for the remainder of my PhD, and led to a collaboration with Alberto that ultimately resulted in Chapter 6 of this thesis.

Turning now to more specific thanks.

I would like to express my deepest gratitude to my supervisors, Ciara and Patrick. From the very beginning, they placed an extraordinary amount of trust in me, giving me both the freedom to explore my interests and the guidance to do so productively. Their generosity with their time, their confidence in me, and their constant support made this PhD not only possible but deeply enjoyable.

I am very grateful to Nicolò and all members of the LAILA lab for making me feel so welcome during my time in Milan. Those months were both professionally formative and personally memorable.

I would like to thank Axel for encouraging me to apply to the StatML programme and for supervising the first project of my PhD.

I am thankful to my collaborators Alberto, Carlo, David, and Dorian for many stimulating discussions, their support, and the pleasure of working together on these projects.

I owe an enormous debt of gratitude to my family, and especially to my parents, Brigitte and Adrian, for their unwavering love, care, and support throughout my life. They have given me a privileged education and far more than one could ever reasonably ask for.

I would like to thank my friends, whose companionship and happiness are an essential ingredient to successful and sustainable research.

I would like to thank Liverpool FC for the many emotions that somehow made this journey easier.

Finally, I would like to thank Livia: for providing an environment in which I could flourish, for her relentless support, and for shaping my PhD in unexpected ways.

Contents

Abstract **2**

Acknowledgements **3**

1 Introduction **13**

1.1 Sequential Decision-Making 14

1.2 Thesis Outline 15

1.3 Notation 24

2 Preliminaries on Gradient-Based Methods for Sequential Decision Making **26**

2.1 Gradient-Based Methods for Sequential Decision-Making 26

2.2 Online Gradient Descent 27

2.3 Online Mirror Descent 29

2.3.1 Gradient Descent and the Euclidean Geometry 29

2.3.2 Regularisers and Bregman Divergences 30

2.3.3 Online Mirror Descent - The Algorithm 31

2.3.4 The Benefits of Online Mirror Descent 32

2.3.5 Additional proofs 35

2.4 Follow-the-Regularised-Leader 39

2.4.1 Online Mirror Descent as Online Gradient Descent in Dual Space 39

2.4.2 Follow-the-Regularised-Leader - The Algorithm 40

2.5 Roadmap for the Rest of the Thesis 42

3 Preliminaries on Reinforcement Learning **43**

3.1 Markov Decision Processes 43

3.2 Planning in Markov Decision Processes 48

3.2.1 Policy Iteration 48
3.2.2 Policy Gradient Methods 52
3.2.3 Beyond Exact Algorithms 56

I High Dimensional Problems are Fundamentally Different 57

4 Optimal Anytime Online Learning on Lp-Balls 58

4.1 Preliminaries 63
4.1.1 Follow-the-Regularised-Leader 63
4.1.2 Universal optimality 66
4.1.3 Uniform convexity 68
4.1.4 FTRL analysis and regret guarantees 71
4.2 Optimal OCO on ℓ_p-balls 73
4.2.1 Low-dimensional regime 74
4.2.2 High-dimensional regime 75
4.3 Anytime optimality through adaptive regularisation 76
4.4 Necessity of adaptive regularisation 79
4.4.1 Failure of specific regularisers 79
4.4.2 Failure of fixed sign-invariant regularisation 85
4.5 Proof of Theorem 4.4.4 86
4.5.1 Proof of Proposition 4.5.1 88
4.5.2 Lower bound on eta 91
4.5.3 Upper bound on p-star-norm of G 92
4.5.4 Proof of Proposition 4.5.2 94
4.6 Online Mirror Descent 96
4.7 Bandit feedback 97
4.8 Related works 98
4.9 Conclusion 100
4.10 Missing Details - Proofs of Section 4.7 100
4.10.1 Case $p > 4/3$ 101
4.10.2 Case $1 < p \leqslant 4/3$ 107
4.10.3 Case $p \to 1$ 111

5 Dimension-Dependent Adaptivity for Sample-Efficiency in Multi-Batch RL 114

5.1 Problem Setting . . . 118

5.1.1 Markov Decision Processes . . . 118

5.1.2 Best Policy Identification (BPI) . . . 119

5.1.3 Multi-Batch Learning Model . . . 119

5.1.4 Linear Function Approximation . . . 121

5.1.5 Sample Efficiency . . . 121

5.2 Main Result . . . 121

5.3 Analysis . . . 125

5.3.1 Proof Sketch . . . 125

5.3.2 Preliminaries on Subspace Packings . . . 128

5.3.3 Roadmap for the Rest of the Proof . . . 133

5.3.4 MDP Class . . . 133

5.3.5 Instance of the Class . . . 134

5.3.6 Realizability . . . 136

5.3.7 Proof of Theorem 5.2.1 . . . 138

5.3.8 MDP Construction Illustration . . . 142

5.4 Related Works . . . 145

5.5 Conclusion . . . 146

5.6 Missing Details . . . 147

5.6.1 Proof of Lemma 5.3.6 . . . 147

II Exploiting Sparsity 151

6 Stochastic Shortest Path with Sparse Adversarial Costs 152

6.1 Problem Setting . . . 155

6.2 SSP as Online Linear Optimisation and Online Mirror Descent . . . 157

6.3 Failure of Negative Entropy Regularization . . . 159

6.4 The Benefits of Sparsity . . . 164

6.4.1 Lower bound . . . 169

6.4.2 Comparison to Kwon et al. (2016) . . . 173

6.5 Related Works . . . 175

6.6 Conclusion . . . 176

6.7 Missing Details . . . 177
6.7.1 Efficient Implementation of OMD using our Regulariser . . . 177
6.7.2 Additional Computations for the Proof of Theorem 6.3.1 . . . 179
6.7.3 Lower Bound on the Maximum of Asymmetric Zero-Mean Random Walks 184

III Gradient-Based Methods in Reinforcement Learning 190

7 Does Stochastic Gradient Really Succeed for Bandits? 191
7.1 Problem Setting . . . 192
7.2 The Stochastic Gradient Bandit policy (SGB) . . . 193
7.3 The Step-Size Boundary of Logarithmic Regret . . . 195
7.4 Proof of Theorem 7.3.3 . . . 197
7.4.1 Technical Lemmas . . . 200
7.4.2 Maximizing Reward Sequence and Probability Under $\mathcal{S}_0 \cap \mathcal{S}_1$ . . . 203
7.5 Conclusion . . . 215

8 Optimal Convergence Rate for Exact PMD in Discounted MDPs 217
8.1 Problem Setting . . . 221
8.2 Exact Policy Mirror Descent . . . 221
8.3 Properties of PMD . . . 223
8.4 Linear Convergence of Exact PMD with Gamma-Rate . . . 225
8.5 Optimality of PMD . . . 230
8.6 Adaptive Step-Size Necessity . . . 236
8.7 Sample Complexity of Inexact PMD under a Generative Model . . . 237
8.8 Related Works . . . 242
8.8.1 Convergence Rates for Exact PMD . . . 242
8.8.2 Sample Complexity of Inexact PMD . . . 244
8.8.3 Recent Works . . . 244
8.9 Conclusion . . . 245
8.10 Missing Details . . . 245
8.10.1 Simulations . . . 245
8.10.2 Proof of Theorem 8.6.1 . . . 246
8.10.3 MDP on which Distribution-Mismatch Coefficient Scales with Size of State Space . . . 254

8.10.4 Family of MDPs on which Sub-Optimality Gaps can be Made Arbitrarily Small 256

9 Conclusion and Open Questions 258

9.1 Summary of Contributions 258

9.1.1 Part I - High Dimensional Problems are Fundamentally Different 258

9.1.2 Part II - Exploiting Sparsity 259

9.1.3 Part III - Gradient-Based Methods in Reinforcement Learning 259

9.2 Open Questions and Future Directions 259

Bibliography 262

List of Figures

2.1 Illustration ($d = 1$) of Bregman Divergence $D_\psi(x_1, x_0)$ for the negative entropy (left, KL-divergence) and squared ℓ_2-norm (right, squared Euclidean distance). $x_0 = 0.2$, $x_1 = 0.7$. 31

4.1 Optimal rates of regret for OCO on ℓ_p-balls with $p > 2$. 60

4.2 Illustration in $\mathbb{R}^2$ of a uniformly convex function (red, w.r.t. $\|\cdot\|_5$) vs a strongly convex function (blue w.r.t. $\|\cdot\|_2$ so also $\|\cdot\|_5$). For the ℓ_p-ball $\mathcal{B}_5$ that gets closer to a hyper-cube (the boundary can be seen as the top outline of the red surface or in gray on the bottom surface), the strongly convex function takes large values near the corners, while the uniformly convex function does not. 69

5.1 Left: **Information can be erased in multiple directions:** Consider the setting where information is being erased along the pink plane $\mathcal{N}$: the learner's queries $\phi(s_i, a_i)$ are shown in blue and the environment's responses $-\gamma\phi(s_i^+, a)$ are shown in black. The rows of X, $\phi(s_i, a_i) - \gamma\phi(s_i^+, a)$ (blue + black vectors) all lie on the yellow line so the learner acquires no information in the directions of the pink subspace $\mathcal{N}$, the null-space of X. Right: **Information cannot be erased in all directions:** Consider the opposite setting where information is being erased along the pink line $\mathcal{N}$. Because of the constraint $\|\phi(s, a)\|_2 \leqslant 1$ and $\gamma < 1$, a query ϕ_1 (in blue) in the pink cap cannot have $\phi_1 - \gamma\phi_1^+$ (blue + black vector) projected back onto the yellow plane (unless $\|\gamma\phi_1\|_2 > \gamma \implies \|\phi_1\|_2 > 1$). Despite the environment not being able to erase information in certain directions, if the number of queries is "small", it can find directions to erase. 127

5.2 Illustration of a hyperspherical sector of a 1-dimensional subspace (left) and a 2-dimensional subspace (right - all vectors whose direction is within the two pink bands). In both cases, the subspace H is in yellow. 130
5.3 Round 1. 143
5.4 Round 2. 144

6.1 Diagram illustrating MDP construction for the proof of Theorem 6.3.1. When an action is not specified for an edge, then both actions give the same transition and cost. If an edge has a number in black, it is a transition probability; if it does not then the transition is deterministic. The costs are given in red. The formal description of the MDP is given below. 160
6.2 Bregman Divergence between a deterministic distribution $x = [0, 1]$ and the uniform distribution $y = [1/2, 1/2]$ for our regulariser ψ_p, squared Euclidean norm ψ_E and negative entropy H for increasing values of p. 165
6.3 Diagram illustrating MDP construction for the proof of Theorem 6.4.2. 170

8.1 MDP used in the proof of Theorem 8.5.1 231
8.2 $\alpha = 10^{-1}$. Green curve: $y = \gamma^x$. 247
8.3 $\alpha = 10^{-10}$. Green curve: $y = \gamma^x$. 247
8.4 $\alpha = 10^{-50}$. Green curve: $y = \gamma^x$. 247
8.5 $\alpha = 10^{-10}$. $\eta_k = \max(\text{Adaptive}, \eta_0/\gamma^k)$ means the step-size is chosen as the largest between our "Adaptive" and the geometrically increasing of (Xiao, 2022). Green curve: $y = \gamma^x$. 247

List of Tables

2.1 Overview of how gradient-based methods are instantiated across the thesis. . . . 42

4.1 Optimal regret rates for OCO on ℓ_p-balls ($p > 2$). 61
4.2 Chapter overview. 62
4.3 Chapter overview. 63

5.1 Chapter overview. 118

6.1 Chapter overview. 155

7.1 Chapter overview. 192

8.1 Comparison of contributions with prior work that study PMD. Khodadadian et al. (2021) focus on NPG, an instance of PMD for a specific mirror map (see Section 8.1). Their analysis is fundamentally different to ours as it exploits the closed-form update of NPG. Their step-size is similar to ours, although it has a dependence on a sub-optimality gap (see Section 8.4). Linear γ-rate holds if the linear convergence is with the γ-rate, not only linear convergence. The ℓ_∞-bound is satisfied if it holds for $\|V^{\pi^\star} - V^{\pi^k}\|_\infty$. Dimension independence is satisfied when there is no instance for which the bound can scale with the size of the state or action space. We compare these works in more detail in Section 8.4. 220
8.2 Chapter overview. 221

Chapter 1

Introduction

Sequential decision-making is a fundamental framework that captures a wide range of problems in statistics, online learning, optimisation, and reinforcement learning (Robbins, 1952; Bellman, 1957; Puterman, 1994; Littlestone and Warmuth, 1994; Sutton and Barto, 2018; Gordon, 1999; Zinkevich, 2003; Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2012; Bubeck and Cesa-Bianchi, 2012; Hazan, 2016; Orabona, 2019; Lattimore and Szepesvári, 2020) covering many applications including healthcare (Thompson, 1933; Gittins, 1979; Villar et al., 2015), finance (Cover, 1991; Blum and Kalai, 1997; Helmbold et al., 1998; Nevmyvaka et al., 2006; Li and Hoi, 2014) and games (Tesauro, 1995; Mnih et al., 2015; Silver et al., 2016; Berner et al., 2019). It models a learner or agent that repeatedly selects actions and receives feedback with the aim of improving future decisions.

Recent successes in machine learning and sequential decision-making have been driven by large-scale models, such as neural networks with trillions of parameters. Training these models relies on efficient optimisation techniques, among which *gradient-based methods* play a central role, due to their scalability and ability to handle *high-dimensional* problems.

As algorithmic decision-making becomes increasingly prevalent, it is crucial to understand when and why these models and optimisation techniques work. Despite impressive empirical successes, the theoretical understanding of many widely used approaches remains limited. This gap is especially evident in reinforcement learning, where scalable gradient-based algorithms such as policy gradient methods are poorly understood, even in simple settings like the multi-armed bandit. More broadly, much of existing theory is established under simplified or idealised assumptions

that often fail to hold in practice. This abstraction is a necessary step, but progress depends on systematically relaxing these assumptions to move towards more realistic settings, such as non-i.i.d. data or high-dimensional environments.

In this thesis, we explore gradient-based approaches to optimally solve high-dimensional problems across a variety of sequential decision-making settings. Specifically, the thesis is separated into three directions:

1. High-dimensional problems are fundamentally different from low-dimensional problems and require distinct optimal gradient-based methods.

2. Low-dimensional structures, such as sparsity, can be effectively exploited within high-dimensional problems using gradient-based methods.

3. Analysis of the performance of gradient-based algorithms in simple reinforcement learning settings, highlighting both their strengths and limitations.

Together, these results deepen our understanding of the interplay between dimensionality, gradient-based optimisation, and sequential decision-making. More precisely, the thesis combines direct analyses of gradient-based methods, information-theoretic lower bounds that explain the need for adaptivity in high dimension, and algorithm-specific lower bounds for policy-gradient-like updates.

1.1 Sequential Decision-Making

In this introduction, we formalise sequential decision-making via a general framework of a game played by a learner over T rounds where in each round $t = 1, 2, ..., T$:

- the learner picks an action x_t in a set V.
- the learner suffers a (differentiable) loss $\ell_t(x_t)$ where ℓ_t is a loss function $\ell_t : V \to \mathbb{R}$.
- the learner receives feedback $\ell_t(x_t)$ (bandit feedback) or ℓ_t (full-information feedback).

This framework covers a variety of settings that we will consider: online convex optimisation, reinforcement learning, bandit learning and stochastic shortest path. While reinforcement learning and stochastic shortest path involve an internal sequential decision-making process, in which a policy selects actions across states in a Markov decision process, we study these problems from a

higher-level perspective in which policies are deployed sequentially over rounds, treating policies as the learner's actions and framing decision-making over the policy space.

The unified notion of *dimension* that we consider in this framework is that of the action set V, which captures the effective size or complexity of the decision space. Depending on the setting, this dimension may correspond to the ambient Euclidean dimension d or the size of a state–action space SA.

The Regret. The goal of the learner is to make decisions that minimise the total loss $\sum_{t=1}^{T} \ell_t(x_t)$. However, the total loss is not meaningful on its own, since even the best possible strategy may suffer large loss. Instead, performance is measured relative to a fixed benchmark or competitor u, where often $u = \arg\min_{x \in V} \sum_{t=1}^{T} \ell_t(x)$. This leads to the notion of regret:

$$R_T \;=\; \sum_{t=1}^{T} \ell_t(x_t) \;-\; \sum_{t=1}^{T} \ell_t(u).$$

Regret quantifies how much worse the learner performs compared to the competitor. Achieving *sublinear regret*, that is $R_T = o(T)$, ensures that the average regret R_T/T tends to zero, meaning the learner eventually performs as well as the competitor on a per-round basis.

We will also sometimes be interested in producing a single final decision x_T at time T that minimises a terminal loss, rather than minimising cumulative loss over all rounds. This setting can be incorporated into the regret framework by defining the losses to be zero for all rounds prior to time T, so that the regret reduces to a comparison based solely on the final decision.

1.2 Thesis Outline

In this thesis, we are interested in establishing upper and lower bounds on the regret that match up to constant factors. Upper bounds demonstrate that a given algorithm can achieve a certain level of performance. Lower bounds, on the other hand, play a complementary role: they certify that no algorithm can perform substantially better under the same assumptions, allowing us to distinguish fundamental limitations from algorithmic shortcomings. The optimal regret corresponds to the regime where the lower and upper bound meet. It depends on problem parameters such as the geometry of the set V, regularity assumptions on the losses ℓ_t, the time horizon T, and often crucially on the dimension of the problem.

The central focus of this thesis is to understand the optimal regret in high-dimensional settings and whether gradient-based methods can achieve it.

In what follows, we outline the specific settings and contributions of the three parts of the thesis.

Part I - High Dimensional Problems are Fundamentally Different

The first part of the thesis studies separations in optimality between low and high dimensional settings. It contains two chapters (Chapter 4, Chapter 5).

Chapter 4 - On the necessity of adaptive regularisation: Optimal anytime online learning on ℓ_p-balls

We study regret minimisation in Online Convex Optimisation (OCO) (Gordon, 1999; Zinkevich, 2003) where the losses are convex and adversarial and the learner competes against the best fixed point in hindsight. We consider the case where the action set $V = \left\{x \in \mathbb{R}^d : \|x\|_p \leqslant 1\right\}$ is an ℓ_p-ball in d-dimensional space with $p > 2$, which has been the focus of many works in optimisation (d'Aspremont et al., 2018; Levy and Duchi, 2019; Guzmán and Nemirovski, 2015; Kerdreux et al., 2021b,c) as it interpolates between strongly curved and nearly flat geometries, and serves as a canonical example for understanding optimisation beyond Euclidean space.

For ℓ_p-balls with $p > 2$, the optimal regret exhibits a shift from the high-dimensional setting ($d > T$) where it scales with $T^{1-1/p}$ (Sridharan and Tewari, 2010; Nemirovskij and Yudin, 1983; Guzmán and Nemirovski, 2015; Guzmán, 2015) to the low-dimensional setting ($d \leqslant T$) where it scales with $T^{1/2}d^{1/2-1/p}$ (Levy and Duchi, 2019). The shift in the optimal regret already highlights the fundamental difference of the high-dimensional setting. The focus of this chapter is then to understand how this impacts the ability of gradient-based methods to achieve the optimal regret in both settings.

We study the behaviour of the gradient-based Follow-The-Regularised-Leader (FTRL) family of algorithms that is parametrised by a regularisation function (see Section 2.4). FTRL achieves the optimal regret in both settings but with different regularisers: strongly-convex in low-dimensions and uniformly-convex regulariser of degree p (see Definition 4.1.6) in high-dimensions (Orabona, 2019; Bhaskara et al., 2020). In practice, however, the learner typically does not know the

time horizon T in advance and therefore cannot determine at the outset whether the problem will fall into the low- or high-dimensional regime. We show that *anytime* optimality where the optimal regret is obtained simultaneously across all dimension regimes can be achieved by FTRL through adaptive regularisation, whereby the algorithm initially employs the uniformly convex regulariser suitable for the high-dimensional setting, and later switches to the strongly convex regulariser. While this approach achieves the desired anytime regret optimality, it leaves open a deeper structural question about FTRL and the differences between low and high dimensional problems: *whether such adaptivity is fundamentally necessary for anytime optimality, or whether a fixed regulariser could suffice.*

Contributions: To answer this question, it is natural to first consider the regularisers that are optimal in one of the dimension regimes. However, we give algorithm-specific lower bounds that show that these are provably sub-optimal in the complementary regime. More generally, the main result and contribution of this chapter is to show that adaptive regularisation for FTRL is necessary to achieve anytime optimality for the general class of sign-invariant regularisers. This result rules out the existence of a single regulariser with which FTRL is optimal both in low and high dimensional settings. *It establishes fundamental differences in the gradient-based methods that achieve optimality in both of these settings.* In particular, optimal low-dimensional methods fail in high-dimensions.

Bibliographic remark: This chapter is based on Johnson et al. (2025a) with the main result being extended to more general regulariser. It is my own work. It is licensed under a Creative Commons Attribution 4.0 International License.

Chapter 5 - Sample-Efficiency in Multi-Batch Reinforcement learning: The Need for Dimension-Dependent Adaptivity

Data collection in Reinforcement Learning (RL) usually falls into two main paradigms: online and offline. An online learner interacts with the environment, immediately receiving feedback and adapting its decisions in real-time. In contrast in offline RL, the dataset is collected in a single batch prior to observing any feedback. Many practical applications consider RL algorithms with limited adaptivity, which fall between online and offline RL. For example, in clinical trials, groups of patients undergo multiple treatments simultaneously and the treatment allocations are only be updated once the outcomes of the all the previous group have been observed (Yu et al., 2021). Similar settings with parallelized data-collection include marketing, advertising

and numerical simulations (Esfandiari et al., 2021). There are also applications where the data collection needs to be validated prior to deployment by a human due to concerns of safety (Dann et al., 2019) and fairness (Koenecke et al., 2020), which with large scale data collection, is not feasible to have at every data point.

This motivates considering the multi-batch learning model (Perchet et al., 2015) where a data-set of size n is collected in K batches, and feedback from within a batch is only observed at the end of a batch. The multi-batch setting covers different levels of adaptivity to feedback. At one extreme, all the data is collected in a single batch ($K = 1$, offline-RL, no adaptivity). At the other extreme, each batch contains a single data-point ($K = n$, full adaptivity). We say a learner is *adaptive* when $K > 1$. In some of the applications that parallelize data-collection mentioned above, algorithms with low-adaptivity where K is as small as possible are of interest since there may be a cost associated to unifying and processing the parallelized data.

We study adaptivity in infinite-horizon discounted Markov Decision Processes (MDPs) under linear function approximation. As discussed earlier, reinforcement learning involves an internal sequential decision-making process in which a policy selects actions across states in an MDP. In this chapter, however, we adopt a higher-level perspective in which the learner sequentially determines how data are collected across interactions with the environment. Furthermore, unlike the regret-minimisation setting considered elsewhere in the thesis, our focus here is on producing a single final output once data collection is complete. Specifically, we consider the best policy identification (BPI) problem, where the learner is tasked with finding a near-optimal policy. A desirable property of the learner is that it is **sample-efficient**, i.e. it only needs a dataset size that is polynomial in the dimension d of the linear function approximation. We aim to understand the minimum level of adaptivity necessary for sample efficient learning.

It is known there is a sample-efficiency separation between offline RL ($K = 1$) and fully-adaptive RL ($K = n$) under linear function approximation. Algorithms have been shown to be sample-efficient in the fully-adaptive setting (Lattimore et al., 2020) and under various assumptions in the offline setting (Duan et al., 2020; Xie and Jiang, 2020). Without assumptions in the offline setting, it has been shown information-theoretically that no sample-efficient algorithms can solve BPI problems (Zanette, 2021) even under the best possible offline dataset, showing the separation. However, the MDP constructions of Zanette (2021) are easily solved with algorithms

using $K = 2$ batches, suggesting that the boundary of this separation may be between offline RL ($K = 1$) and RL with adaptivity ($K > 1$). This observation naturally raises the question of where the true boundary of sample efficiency lies: does it coincide with the transition from offline to adaptive learning, or does it depend more subtly on the dimension of the problem and the level of adaptivity K?

Contributions: In this chapter, we establish $\Omega(\log \log d)$ lower-bounds on the number of adaptive updates, K, required to solve BPI problems sample-efficiently. The results show that adaptivity ($K > 1$) does not guarantee sample-efficiency. The level of adaptivity, or number of batches K, needed to guarantee sample-efficiency scales with the dimension d of the linear representation. In particular, the boundary of sample-efficiency does not lie between offline RL ($K = 1$) and adaptive RL ($K > 1$). Instead this boundary must lie within a regime of adaptivity scaling with dimension: $\Omega(\log \log d) \leqslant K \leqslant n$. *This establishes that high-dimensional problems in this setting are fundamentally harder than low-dimensional ones because they requires more batches of data as the dimension grows* and optimal methods for the latter are not for the former.

Bibliographic remark: This chapter is based on Johnson et al. (2024) and is my own work. It is licensed under a Creative Commons Attribution 4.0 International License.

Part II - Exploiting Sparsity

The second part of the thesis studies to what effect sparsity can be exploited to improve dimension dependence in the regret guarantees of adversarial stochastic shortest path problems. It contains one chapter (Chapter 6).

Chapter 6 - Stochastic Shortest Path with Sparse Adversarial Costs

The Stochastic Shortest Path (SSP) problem is a fundamental model in reinforcement learning (Bertsekas and Tsitsiklis, 1991; Tarbouriech et al., 2020), which describes tasks where an agent interacts with an environment over episodes and must reach a designated goal state within each episode while minimizing accumulated costs. This covers problems such as car navigation while trying to avoid traffic jams, or internet routing. This classical setting has recently been extended to the adversarial regime, where costs may vary arbitrarily between episodes (Rosenberg and Mansour, 2021; Chen et al., 2021c; Chen and Luo, 2021; Zhao et al., 2022) and the goal is to obtain theoretical guarantees robust to any cost-generation mechanism.

Under *full-information* feedback where the full cost vector is observed after each episode and known transitions, current algorithms (Rosenberg and Mansour, 2021; Chen et al., 2021c) achieve regret bounds which scale with the size of the state-action space SA (the dimension of the problem) as $\mathcal{O}\left(\sqrt{\log SA}\right)$. These bounds are independent of any cost structure and we show this unimprovable in the worst-case. However, many real-world high-dimensional problems have costs with structural properties that may be leveraged for improved regret. A common property often considered in the statistics and machine learning literature (Wainwright, 2019b) is sparsity, which can naturally arise for SSP problems. For instance, in the car navigation example, the number of traffic jams is usually much smaller than the number of roads. Motivated by this, we consider *sparsity* in high-dimensional SSP problems where M, the maximum number of state-action pairs with non-zero cost in an episode, can be much smaller than the dimension SA.

In such scenarios, the regret bounds should capture some dependence on M, reflecting an improvement in performance on easier sparse problems as $M \to 1$ and recovering the standard bounds on worst-case problems as $M \to SA$. In fact, for the sparse experts setting[1] ($S = 1$), polynomial improvements in the dimension A can be obtained (Kwon and Perchet, 2016) by the gradient-based Online Mirror Descent (OMD) algorithm with negative entropy regularisation (see Chapter 2). Motivated by this, the focus of this chapter is to understand the extent to which sparsity can be exploited in the more general SSP problem and the ability of gradient-based methods to do so optimally.

Contributions: We show that sparsity cannot be exploited by OMD with negative entropy regularization, which is used in the experts setting and in prior work on SSP to achieve the worst-case optimal $\sqrt{\log SA}$ dependence (Chen et al., 2021c). We construct a *sparse* SSP problem where this algorithm suffers $\sqrt{\log S}$ regret, providing no improvement in terms of S compared to the minimax regret for the non-sparse worst-case $M = SA$ problem. Instead, we consider OMD with a family of regularizers based on ℓ_r-norms for $r \in (1, 2)$. We show the regret scales with $\sqrt{\log M}$ instead of $\sqrt{\log SA}$. We also show that the $\sqrt{\log M}$ dependence on M is unimprovable by constructing a matching lower bound. This establishes that the benefit of sparsity in SSP is logarithmic in SA instead of polynomial, as in the simpler experts problem. It also highlights that M plays the role of effective dimension, replacing the general dimension SA in controlling

[1]The experts setting is the single-state full-information feedback A-action online learning problem (Cesa-Bianchi et al., 1997b; Freund and Schapire, 1997).

the scaling of the minimax regret.

Our results demonstrate that sparsity can effectively and optimally be exploited by gradient-based methods in high-dimensional full-information feedback SSP problems to obtain dimension-independent regret bounds that only depend on the sparsity level.

Bibliographic remark: This chapter is based on Johnson et al. (2025b). It was done jointly with Alberto Rumi; I worked on all aspects of the work except the unknown transition lower bound. It is licensed under a Creative Commons Attribution 4.0 International License.

Part III - Gradient-Based Methods in Reinforcement Learning

The third and final part of the thesis studies the performance of gradient-based methods in RL, based on optimising a parameterised policy with respect to an objective. We consider these methods in simple settings, the single-state two-armed bandit setting and the exact gradient setting, where they have not yet been fully understood. Gradient-based methods and in particular the policy gradient algorithm (see Chapter 3; (Sutton et al., 1999; Konda and Tsitsiklis, 1999; Kakade, 2001)) are ubiquitous in RL due to their scalability. Studying their behaviour in these simple settings provides foundational insight that is essential for understanding their performance in more complex, high-dimensional problems, which are the central focus of this thesis. This part contains two chapters (Chapter 7, Chapter 8).

Chapter 7 - Does Stochastic Gradient Really Succeed for Bandits?

We study the regret properties of the *Stochastic Gradient Bandit* (SGB) algorithm (Sutton and Barto, 2018, Chapter 2.8) for the multi-armed bandit problem. SGB is an instance of the softmax policy gradient algorithm in the bandit setting and can be viewed as a canonical building block of policy gradient methods. Understanding its behaviour in the bandit setting provides insight into the strengths and limitations of gradient-based reinforcement learning algorithms more broadly and the results we provide can serve as a stepping stone towards understanding policy gradient methods in more general setting.

Even in the simple single-state bandit setting, the theoretical guarantees of policy gradient methods in terms of regret remain weak. In this stochastic multi-armed bandit setting, it is

known that regret scaling with $\log T$ is achievable using classical algorithms such as UCB (Auer et al., 2002) or Thompson Sampling (Agrawal and Goyal, 2012). Unlike these classical bandit algorithms, SGB does not rely on explicit exploration mechanisms or confidence bounds. Instead, its behaviour is governed by a single learning-rate parameter η, whose role is crucial in balancing exploration and exploitation. Recently, Mei et al. (2023) established a regret upper bound of $\mathcal{O}(\log(T))$ for SGB with a *small* constant learning rate η satisfying $\eta \lesssim \Delta^2/K^{3/2}$ where K is the number of arms and Δ is the smallest sub-optimality gap between the expected reward of the arms. In a follow-up work, Mei et al. (2024) further proved that SGB asymptotically converges to a globally optimal policy for *any* constant learning rate. While asymptotic convergence is a desirable property, it does not provide meaningful regret guarantees. This motivates further investigation of the regret properties of SGB without the small rate constraint. For the $K = 2$ armed problem, Baudry et al. (2025) show that logarithmic regret is achievable for a larger range of step-size $\eta \lesssim \Delta$. What remains is to understand the regret of SGB when $\eta > \Delta$, which is the focus of this chapter.

Contributions: The main theoretical contribution is the identification of a sharp learning-rate threshold separating logarithmic and polynomial regret regimes for the $K = 2$ case. We prove that the regret of SGB can be **polynomial** on some instances if $\eta \gtrsim \Delta$. Hence, we characterize the regret of SGB in two-armed bandits, except for a critical regime $\eta \approx \Delta$. *These results establish Δ as the critical threshold for η between logarithmic and polynomial results and imply that knowledge of the gap Δ is both necessary and sufficient to tune the learning rate so as to guarantee logarithmic regret and optimal performance.*

Bibliographic remark: This chapter is based on Section 2.2 of Baudry et al. (2025). The work was done in collaboration with Dorian Baudry; Dorian Baudry is the first author and my contributions involve the lower bounds, in particular Theorem 2 in Section 2.2. It is licensed under a Creative Commons Attribution 4.0 International License.

Chapter 8 - Optimal Convergence Rate for Exact Policy Mirror Descent in Discounted MDPs

The problem of finding an optimal policy in tabular discounted Markov Decision Processes (MDPs) was classically solved using dynamic programming approaches such as policy iteration (PI, see Chapter 3) and value iteration (VI) (Puterman, 1994; Sutton and Barto, 2018). These methods are well understood theoretically and are guaranteed to converge *linearly* (geo-

metrically) to the optimal policy in the tabular setting with a rate equal to the discount factor γ of the MDP (Bellman, 1957). In this chapter, continuing our effort to better understand the performance of gradient-based methods, we consider this problem through the lens of policy gradient methods and aim to determine whether their guarantees can match those of classical algorithms in tabular MDPs. In particular, we view the sequential decision-making process in *policy space*, where the learner selects a policy at each iteration and observes the resulting value function, rather than acting directly in the environment. The goal is not to minimise regret over rounds, but rather to produce a *final* policy that is near-optimal.

Among the recent works focused on understanding policy-gradient methods in the tabular setting, Xiao (2022) study a general family of algorithms based on the gradient-based mirror descent method (see Chapter 2) known as Policy Mirror Descent (PMD). PMD can be interpreted as algorithmically regularising the policy improvement step of policy iteration (PI), and can therefore be seen as a form of regularised PI, without actually modifying the original objective. A notable instance of PMD is the Natural Policy Gradient (NPG) algorithm (Kakade, 2001), which underlies many widely used methods in modern reinforcement learning (Schulman et al., 2015, 2021). This makes the study of PMD particularly interesting because it can allow us to not only better understand NPG but also discover closely related algorithms which may share similar properties while having additional desirable features as well.

Under exact gradients, linear convergence of PMD was established by Xiao (2022), though their rate depends on an instance-dependent factor that can scale poorly with the dimension of the problem, such as the size of the state space. The γ-rate of classical approaches (PI and VI) can be achieved by PMD when the MDP objective is regularised (Cen et al., 2021; Lan, 2021; Zhan et al., 2023). However, the classical approaches achieve the γ-rate without MDP-regularisation, revealing that regularisation is, in general, not necessary for algorithms to reach the γ-rate. In fact, Khodadadian et al. (2021) showed that NPG can achieve the γ-rate, suggesting that such rates are attainable by instances of PMD without regularisation. Therefore, in this chapter, we aim to understand if the classical dimension-independent linear γ-rate can be matched by general PMD in unregularised MDPs, and more generally what the best achievable rate is.

Contributions: *Our results establish that in the exact gradient setting, PMD can converge with the dimension-independent linear γ-rate,* under an adaptive step-size. We also show that

the γ-rate is in fact the best rate achievable for PMD as well as for a more general family of algorithms. Finally, we extend our results to the inexact setting where gradients can be estimated using a generative model and improve the sample complexity of PMD to the optimal dependence on the size of the state-action space.

Bibliographic remark: This chapter is based on Johnson et al. (2023) and is my own work. It is licensed under a Creative Commons Attribution 4.0 International License.

1.3 Notation

We summarise here some notation used throughout the thesis.

Basic operations. For real numbers A, B, we write

$$A \wedge B := \min\{A, B\}, \qquad A \vee B := \max\{A, B\}, \qquad (A)_+ := A \vee 0.$$

Norms and duality. For $r \geqslant 1$, we denote by $r_\star$ the dual exponent satisfying $1/r + 1/r_\star = 1$. For $x \in \mathbb{R}^d$, the ℓ_r norm is defined as

$$\|x\|_r := \Big(\sum_{i=1}^d |x_i|^r\Big)^{1/r}.$$

Given a norm $\|\cdot\|$, its dual norm $\|\cdot\|_\star$ is defined by

$$\|y\|_\star := \sup_{\|x\| \leqslant 1} \langle y, x\rangle.$$

We denote by $\mathcal{B}_p := \{x \in \mathbb{R}^d : \|x\|_p \leqslant 1\}$ the unit ℓ_p ball.

Sets and vectors. For a set $V \subset \mathbb{R}^d$, $\operatorname{int} V$ denotes its interior, $\operatorname{bdry} V$ its boundary. We write e_i for the i-th canonical basis vector of $\mathbb{R}^d$.

Time indexing. We use the double index $x_{t,i}$ to denote the i-th coordinate of the vector x_t at time t. When referring to an entire vector at time t, or to a coordinate of a vector without a time index, we use a single index; this should be clear from context.

Asymptotic notation. For functions $f, g : \mathbb{R} \to \mathbb{R}$, we write $f(x) = \mathcal{O}(g(x))$ (respectively $f(x) = \Omega(g(x))$) if there exist constants $c > 0$ and $N > 0$ such that for all $x \geqslant N$,

$$f(x) \leqslant cg(x) \qquad \text{(respectively } f(x) \geqslant cg(x)\text{)}.$$

We use $\widetilde{\mathcal{O}}(\cdot)$ and $\widetilde{\Omega}(\cdot)$ to suppress logarithmic factors.

Chapter 2

Preliminaries on Gradient-Based Methods for Sequential Decision Making

Many of the gradient-based methods studied in this thesis share a common foundation: they can be interpreted within the framework of the *online mirror descent* (OMD) algorithm. This chapter is devoted to introducing this framework while we leave the details of some specific instantiations and analyses to the relevant chapters. Useful references on the topic are Orabona (2019); Beck and Teboulle (2003); Bubeck (2011); Bubeck and Cesa-Bianchi (2012); Hazan (2016). Throughout this chapter, we assume basic familiarity with convex analysis.

2.1 Gradient-Based Methods for Sequential Decision-Making

We consider the sequential decision-making framework described in Section 1.1. Recall that at each round t, the learner chooses an action $x_t \in V_t$ and suffers a loss $\ell_t(x_t)$. For ease of exposition in this chapter, we consider a simplified setting with a few additional assumptions:

- **the losses are convex:** In Chapter 8, we consider an offline problem where the objective is not convex but the general ideas from this chapter remain relevant.
- **the losses are differentiable** and we are able to compute the gradient $g_t = \nabla \ell_t(x_t)$ at the current point x_t.

We formally define what we mean by a gradient-based method:

Definition 2.1.1. *We use the term* gradient-based method *to refer to any sequential decision-making algorithm whose update rule for x_{t+1} depends only on past iterates $x_1, \ldots, x_t$ and first-order gradient information $g_1, \ldots, g_t$. The learner computes x_{t+1} through an update of the form*

$$x_{t+1} = \mathcal{A}(x_1, g_1, x_2, g_2, \ldots, x_t, g_t),$$

where $\mathcal{A}$ is an algorithm-specific update map.

Procedure 2.1: Gradient-Based Method for Sequential Decision-Making

Learner picks initial point $x_1 \in V$

For $t \in \{1, 2, \cdots, n\}$:

1. Learner plays $x_t \in V$
2. Learner suffers loss $\ell_t(x_t)$
3. Learner computes the gradient $g_t = \nabla \ell_t(x_t)$
4. Learner computes x_{t+1} based on $(x_1, \cdots x_t)$ and $(g_1, \cdots, g_t)$

This definition of gradient-based methods summarised in Procedure 2.1 is general. Next, we give a specific example that will serve as a starting point to understand the Online Mirror Descent algorithm.

2.2 Online Gradient Descent

We begin with a brief presentation of *Online Gradient Descent* (OGD), a canonical instance of online gradient-based methods and a special case of Online Mirror Descent. It provides a simple and intuitive baseline that will be useful for understanding the more general framework introduced later.

Given a step-size η_t, the previous iterate x_t and gradient $g_t = \nabla \ell_t(x_t)$, OGD computes the

update as a projected gradient step:

$$x_{t+1} \;=\; \arg\min_{x \in V} \left\| x - (x_t - \eta_t g_t) \right\|_2^2. \qquad \text{(OGD update)}$$

If the unconstrained update $x_t - \eta_t g_t$ lies in V, OGD reduces to a standard step of gradient descent $x_{t+1} = x_t - \eta_t g_t$. If not, OGD performs the gradient step followed by a Euclidean projection back into V.

Intuition for Online Gradient Descent. It is not immediately obvious why OGD should perform well, especially when the losses are adversarial. Nevertheless, in important settings, such as Euclidean spaces with Lipschitz losses, OGD achieves sublinear regret in the time horizon T. While we should be cautious about intuition in adversarial environments, we offer the following perspective. The regret is defined with respect to the best *fixed* decision in hindsight,

$$u = \arg\min_{x \in V} \sum_{t=1}^{T} \ell_t(x),$$

which induces an inherent notion of stability: although individual losses may be adversarial and uninformative about the future, their aggregate constrains where the optimal comparator can lie. In particular, for a learner to incur large regret, the adversary must either repeatedly lure the learner toward a region that appears favourable in the short term but is later revealed to have high cumulative loss, or consistently ensure that one region of the space has significantly lower cumulative loss than others. The latter effect cannot be created by a single loss function, but only through persistent loss differences over many rounds, which a learning algorithm can exploit.

OGD exactly leverages this structure by moving gradually in response to past gradients. While any individual gradient may be misleading, their cumulative effect points toward the location of the best fixed comparator in hindsight. The slow, controlled updates of OGD prevent overreaction to individual misleading losses, while still ensuring steady progress toward the stable hindsight optimum.

2.3 Online Mirror Descent

2.3.1 Gradient Descent and the Euclidean Geometry

Let's for a moment consider the classical offline (projected) gradient descent algorithm to minimise an objective function f:

$$x_{t+1} = \arg\min_{x \in V} \left\{ \left\| x - \Big(x_t - \eta_t \cdot \nabla f(x_t) \Big) \right\|_2^2 \right\}.$$

It is an optimisation method that leverages a local first-order linear approximation of f at each iteration. A naïve approach of leveraging such an approximation would be to directly use it instead of f:

$$\min_{x \in V} f(x) \approx \min_{x \in V} \Big\{ f(x_t) + \langle \nabla f(x_t), x - x_t \rangle \Big\} = \min_{x \in V} \Big\{ \langle \nabla f(x_t), x \rangle \Big\},$$

but this drives the iterate to an extreme point of V, typically far from x_t, where the linear approximation is inaccurate. To effectively leverage the linear approximation, the iterate should be encouraged to remain in the region where the approximation is accurate: close to x_t. Algorithmically, this is achieved by penalising points that are "far" from x_t. For classical gradient descent, this penalisation is enforced through the Euclidean ℓ_2-norm. While this is not reflected in the typical way of writing the (projected) gradient descent step, it can be rewritten as:

$$\begin{aligned}
x_{t+1} &= \arg\min_{x \in V} \left\{ \left\| x - \Big(x_t - \eta_t \cdot \nabla f(x_t) \Big) \right\|_2^2 \right\} \\
&= \arg\min_{x \in V} \Big\{ \|x\|_2^2 + 2\langle x, \eta_t \cdot \nabla f(x_t) - x_t \rangle \Big\} \\
&= \arg\min_{x \in V} \Big\{ 2\eta_t \cdot \langle x, \nabla f(x_t) \rangle + \|x\|_2^2 - 2\langle x, x_t \rangle \Big\} \\
&= \arg\min_{x \in V} \Big\{ 2\eta_t \cdot \langle x, \nabla f(x_t) \rangle + \|x - x_t\|_2^2 \Big\} \\
&= \arg\min_{x \in V} \left\{ \eta_t \cdot \Big(f(x_t) + \langle \nabla f(x_t), x - x_t \rangle \Big) + \frac{1}{2} \|x - x_t\|_2^2 \right\}.
\end{aligned}$$

The step-size can now be interpreted as a parameter that balances the trade-off between minimising the first order approximation and the penalisation of points far from the previous iterate.

This alternative formulation also applies to OGD, the online analogue of gradient descent dis-

cussed earlier in the previous section:

$$\begin{aligned} x_{t+1} &= \arg\min_{x \in V} \left\{ \eta_t \cdot \Big(\ell_t(x_t) + \langle g_t, x - x_t \rangle \Big) + \frac{1}{2} \|x - x_t\|_2^2 \right\} \\ &= \arg\min_{x \in V} \left\{ \eta_t \cdot \langle g_t, x \rangle + \frac{1}{2} \|x - x_t\|_2^2 \right\}. \end{aligned} \tag{OGD update}$$

This expression makes explicit that OGD is fundamentally adapted to the Euclidean geometry: the proximity term uses the squared ℓ_2-norm as its notion of distance. However, this Euclidean notion of distance is not appropriate for all action sets (e.g. the simplex). To design algorithms adapted to other geometries, we can use a different notion of distance. Online Mirror Descent does exactly this with a specific measure of distance that we introduce in the following section.

2.3.2 Regularisers and Bregman Divergences

The notion of distance used by Online Mirror Descent is induced by a strictly convex function ψ, which we refer to as a *regulariser* since we can view the penalisation as regularisation.

Definition 2.3.1 (Regulariser)**.** *A* regulariser *is a strictly convex function $\psi : V \to \mathbb{R}$ satisfying one of the following:*

1. *ψ is differentiable on V; or*
2. *ψ is differentiable on $\operatorname{int} V$ and, for all $x \in \operatorname{bdry} V$ and $y \in \operatorname{int} V$,*

$$\lim_{\lambda \to 0^+} \langle \nabla \psi(x + \lambda(y - x)), y - x \rangle = -\infty.$$

The regulariser naturally defines a notion of distance between two points captured by the *Bregman divergence.*

Definition 2.3.2 (Bregman Divergence)**.** *Given a regulariser ψ, the* Bregman divergence *$D_\psi : V \times \operatorname{int} V \to \mathbb{R}$ is defined as*

$$D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla \psi(y), x - y \rangle.$$

Geometrically, $D_\psi(x, y)$ measures how much $\psi(x)$ exceeds its first-order (linear) approximation at y. This is illustrated in Figure 2.1. Because ψ is strictly convex, this discrepancy is always non-negative and vanishes if and only if $x = y$.

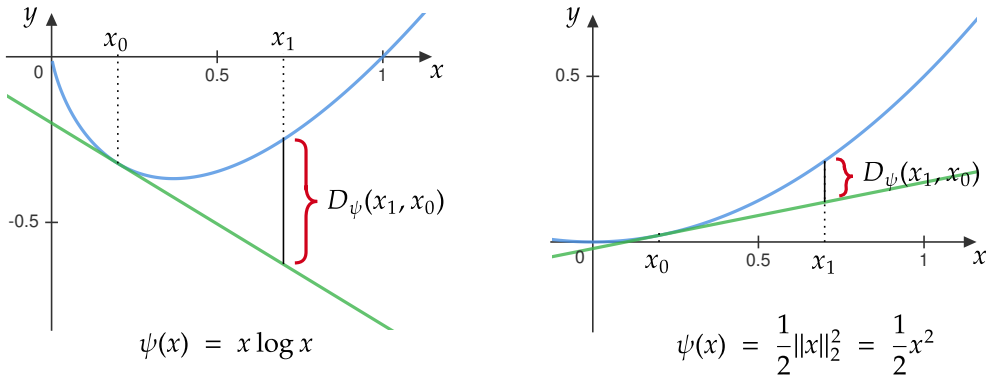

Figure 2.1: Illustration ($d = 1$) of Bregman Divergence $D_\psi(x_1, x_0)$ for the negative entropy (left, KL-divergence) and squared ℓ_2-norm (right, squared Euclidean distance). $x_0 = 0.2$, $x_1 = 0.7$.

There are many possible choice for ψ. Two of the most common ones are:

- **Euclidean regulariser:** $\psi(x) = \frac{1}{2}\|x\|_2^2$, from which we will recover OSD. The corresponding divergence is the squared Euclidean distance:

$$D_\psi(x, y) = \tfrac{1}{2}\|x - y\|_2^2.$$

- **Negative entropy regulariser:** $\psi(x) = \sum_{i=1}^d x_i \log x_i$ (for $V \subseteq \mathbb{R}^d_{\geqslant 0}$). The corresponding divergence is the Kullback–Leibler (KL) divergence:

$$D_\psi(x, y) = \sum_i x_i \log \frac{x_i}{y_i} - x_i + y_i.$$

2.3.3 Online Mirror Descent - The Algorithm

Equipped with the concept of regularisers and Bregman divergences, we can now finally present the *Online Mirror Descent* (OMD) update. As previously discussed, OMD is a generalisation of OGD where the penalisation or proximity term in the update is replaced by the Bregman divergence of a regulariser ψ:

$$x_{t+1} = \operatorname*{arg\,min}_{x \in V} \Big\{ \eta_t \cdot \langle g_t, x \rangle + D_\psi(x, x_t) \Big\}. \qquad \text{(OMD update)}$$

The update is well-defined since if ψ is differentiable on V then D_ψ can be defined on $V \times V$ and if the second condition in Definition 2.3.1 is satisfied then successive iterates are guaranteed to

stay in $\operatorname{int} V$ (e.g. see Theorem 6.8 in Orabona (2019)).

The update can be divided into an unconstrained optimisation and a Bregman projection (see Section 6.4.3 in Orabona (2019)):

$$y_{t+1} = \arg\min_{x \in \mathbb{R}^d} \left\{ \eta_t \cdot \langle g_t, x \rangle + D_\psi(x, x_t) \right\}$$
$$x_{t+1} = \arg\min_{x \in V} D_\psi(x, y_{t+1})$$

With the Euclidean regulariser $\psi(x) = \frac{1}{2}\|x\|_2^2$, we directly recover OGD. With the negative entropy regulariser $\psi(x) = \sum_{i=1}^d x_i \log x_i$, the unconstrained update is $y_{t+1} = x_t \exp(-\eta_t g_t)$ and when the action set is the probability simplex $V = \Delta_{d-1} = \{x \in \mathbb{R}^d_{\geqslant 0} : \|x\|_1 = 1\}$, the Bregman projection is a normalisation step giving the OMD update for $i = 1, ..., d$:

$$x_{t+1,i} = \frac{x_{t,i} e^{-\eta_t g_{t,i}}}{\sum_{j=1}^d x_{t,j} e^{-\eta_t g_{t,j}}},$$

which is a popular algorithm known under many different names (e.g. hedge (Freund and Schapire, 1997), weighted majority (Littlestone and Warmuth, 1994), exponentiated gradient (Kivinen and Warmuth, 1997), ... etc). We will refer to it as *Exponentiated Gradient* (EG).

2.3.4 The Benefits of Online Mirror Descent

We have motivated OMD as a generalisation of OGD that can be better adapted to non-Euclidean geometries. We now show that this intuition is more than conceptual: there exists a real separation in the performance of different instances of OMD for the simplex.

Consider the action set to be the probability simplex $V = \Delta_{d-1} = \{x \in \mathbb{R}^d_{\geqslant 0} : \|x\|_1 = 1\}$ and assume $\|g_t\|_\infty \leqslant 1$. The standard analysis of OMD with a strongly convex regulariser (e.g. Theorem 6.11 in Orabona (2019)) implies that, with appropriately chosen constant step-sizes, Exponentiated Gradient (EG) achieves a regret bound

$$R_T \leqslant \sqrt{2T \log d},$$

while Online Gradient Descent (OGD) achieves a regret bound

$$R_T \leqslant 2\sqrt{Td}.$$

These regret bounds follow from Section 6.6 in Orabona (2019). The $\sqrt{T \log d}$ dependence in the regret of EG is optimal both asymptotically (Cesa-Bianchi and Lugosi, 2006) and non-asymptotically (Orabona and Pal, 2015).

The improvement factor $\sqrt{\frac{\log d}{d}}$ appears to confirm the benefits of generalising OMD beyond OGD. However, upper bounds alone do not rule out the possibility that OGD might itself achieve a $\sqrt{T \log d}$ regret with an improved analysis. To establish a true separation we show in the following proposition that the $\sqrt{Td}$ dependence in the regret of OGD is tight.

Proposition 2.3.3. *Fix $\frac{1}{2}T > d \geqslant 4$. Let $x_1 = [1/d, \ldots, 1/d] \in \Delta_{d-1}$, and suppose the losses are convex and L-Lipschitz with respect to the ℓ_1-norm. Then, for any constant step-size $\eta > 0$, there exists a sequence of losses for which OGD incurs regret*

$$R_T \geqslant \frac{L}{8} \min\{T, \sqrt{Td}\}.$$

Proof. The claim follows by combining two complementary lower bounds, proved in Section 2.3.5, which apply in different regimes of the step size.

When the step size is *large*, OGD becomes overly reactive. The adversary alternates the sign of a fixed linear loss, causing the gradient direction to flip at every round. As a result, the iterates oscillate between distant regions of the simplex, which leads to linear regret in T once the step size is sufficiently large giving the following proposition.

Proposition 2.3.4. *Fix $d \geqslant 4$. Let $x_1 = [1/d, \ldots, 1/d] \in \Delta_{d-1}$, and suppose the losses are convex and L-Lipschitz with respect to the ℓ_1-norm. Then, for any constant step-size η, there exists a sequence of losses for which OGD satisfies*

$$R_T \geqslant \frac{TL}{2} \min\{1, \eta d L\}.$$

When the step size is *small*, OGD moves too slowly to reach an optimal action fast enough. The adversary exploits this by using a fixed linear loss that consistently favours a single vertex giving the next proposition.

Proposition 2.3.5. *Fix $d \geqslant 4$. Let $x_1 = [1/d, \dots, 1/d] \in \Delta_{d-1}$, and suppose the losses are convex and L-Lipschitz with respect to the ℓ_1-norm. Then, for any constant step-size $\eta \leqslant dL$, there exists a sequence of losses for which OGD satisfies*

$$R_T \geqslant \frac{1}{4\eta}.$$

When the step-size is moderate, trading off the two lower bounds above gives us the optimal regret of OGD, which depends polynomially on d.

Formally, if $\eta \geqslant \frac{1}{dL}$, the claim follows directly from Proposition 2.3.4. If $\eta \leqslant \frac{2}{TL}$, it follows from Proposition 2.3.5.

Finally, when $\frac{2}{TL} < \eta < \frac{1}{dL}$, combining Propositions 2.3.4 and 2.3.5 yields

$$R_T \geqslant \max\left\{\frac{Td\eta L^2}{2}, \frac{1}{4\eta}\right\} \geqslant \frac{Td\eta L^2}{4} + \frac{1}{8\eta} \geqslant 2\sqrt{\frac{TdL^2}{32}} = \frac{L\sqrt{Td}}{2\sqrt{2}}.$$

This completes the proof. □

Thus, while it is common to argue for the advantage of OMD by contrasting the respective upper bounds of EG and OGD, Proposition 2.3.3 goes one step further and provides a matching algorithmic-specific lower bound for OGD, establishing that *OGD is provably sub-optimal* on the simplex. A similar result was shown for OGD on the ℓ_1-ball in Levy and Duchi (2019) (Theorem 4) but to the best of our knowledge, this result does not appear in the literature for the simplex and strengthens the usual motivations of OMD.

Universality of OMD: Beyond this canonical example that demonstrates the ability of OMD to achieve optimality on the simplex where OGD does not, OMD is a powerful framework to achieve optimality in general online convex optimisation problems. It was shown by Sridharan (2012) that for a fixed $r \in [2, \infty)$ and constant $C > 0$, for any problem for which a regret upper bound of $CT^{1-1/r}$ can be guaranteed for all T, then for any $T > e^{r-1}$, there exists a regulariser with which OMD that can guarantee a regret bound of $\widetilde{O}(T^{1-1/r})$. This result was recently improved by (Gatmiry et al., 2025) to omit any log-factors in T for the case of online linear optimisation and $r = 2$. The result establishes the universality and optimality of the OMD framework: any achievable regret bound with constant rate across all time horizons can be matched up to logarithmic factors by OMD with some regulariser (see Section 4.1.2).

2.3.5 Additional proofs

Proof of Proposition 2.3.4. Assume without loss of generality that d is even (otherwise replace d by $d-1$). Consider the linear losses

$$\ell_t(x) = (-1)^t L\langle g, x\rangle,$$

where

$$g = \big[\underbrace{-1,\dots,-1}_{d/2 \text{ entries}}, \underbrace{+1,\dots,+1}_{d/2 \text{ entries}}\big].$$

Since $\|g\|_\infty = 1$, each loss ℓ_t is L-Lipschitz with respect to the ℓ_1 norm. Let $x_1 = (1/d,\dots,1/d)$. Define

$$y := x_1 + \eta L g, \qquad y_i = \begin{cases} \frac{1}{d} - \eta L, & i \leqslant \frac{d}{2}, \\ \\ \frac{1}{d} + \eta L, & i > \frac{d}{2}. \end{cases}$$

Case A: $\eta \leqslant \frac{1}{dL}$. In this regime, $y \in \Delta_{d-1}$.

Claim. For all $t \geqslant 1$,

$$x_t = \begin{cases} x_1, & t \text{ odd}, \\ \\ y, & t \text{ even}. \end{cases}$$

Proof of the claim. The base case $t = 1$ is immediate. If t is odd and $x_t = x_1$, then

$$x_{t+1} = x_t - \eta \nabla \ell_t(x_t) = x_1 + \eta L g = y \in \Delta_{d-1}.$$

If t is even and $x_t = y$, then

$$x_{t+1} = y - \eta(-Lg) = x_1 \in \Delta_{d-1}.$$

Thus the iterates alternate between x_1 and y, completing the proof of the claim.

Regret lower bound (Case A). When t is odd,

$$\ell_t(x_t) = -L\langle g, x_1\rangle = 0,$$

while when t is even,

$$\ell_t(x_t) = L\langle g, y\rangle = L \cdot \frac{d}{2}\Big(-\Big(\frac{1}{d} - \eta L\Big) + \Big(\frac{1}{d} + \eta L\Big)\Big) = \eta d L^2.$$

Hence,

$$\sum_{t=1}^{T} \ell_t(x_t) = \frac{T}{2}\,\eta d L^2.$$

Since $\sum_{t=1}^{T} \ell_t(x_1) = 0$, we obtain

$$R_T \geqslant \frac{T}{2}\,\eta d L^2.$$

Case B: $\eta > \frac{1}{dL}$. In this case, $y \notin \Delta_{d-1}$ and the projection becomes active. It can be shown that

$$x_2 = \Pi_{\Delta_{d-1}}(y) = \big[\underbrace{0, \ldots, 0}_{d/2 \text{ entries}}, \underbrace{2/d, \ldots, 2/d}_{d/2 \text{ entries}}\big].$$

We now distinguish two subcases.

Case B.1: $\eta > \frac{2}{dL}$. In this regime, a second update yields

$$x_3 = \big[\underbrace{2/d, \ldots, 2/d}_{d/2 \text{ entries}}, \underbrace{0, \ldots, 0}_{d/2 \text{ entries}}\big].$$

An induction argument shows that for all $t \geqslant 2$,

$$x_t = \begin{cases} x_2, & t \text{ even}, \\ x_3, & t \text{ odd}. \end{cases}$$

Moreover, $\ell_t(x_t) = L$ for $t \geqslant 2$. Therefore,

$$\sum_{t=1}^{T} \ell_t(x_t) \geqslant (T-1)L \geqslant \frac{TL}{2}.$$

Since $\sum_{t=1}^{T} \ell_t(x_1) = 0$, this gives $R_T \geqslant \frac{TL}{2}$.

Case B.2: $\frac{1}{dL} < \eta \leqslant \frac{2}{dL}$. Here, the iterates again alternate between x_2 and a reflected point,

$$x_3 = \big[\underbrace{\eta L, \ldots, \eta L}_{d/2 \text{ entries}}, \underbrace{2/d - \eta L, \ldots, 2/d - \eta L}_{d/2 \text{ entries}}\big].$$

and one checks that

$$\ell_t(x_t) \geqslant \begin{cases} 0, & t \text{ odd}, \\ L, & t \text{ even}. \end{cases}$$

Hence,

$$\sum_{t=1}^{T} \ell_t(x_t) \geqslant \frac{T}{2} L,$$

and again $R_T \geqslant \frac{TL}{2}$.

Conclusion. Combining all cases, we conclude that for any η,

$$R_T \geqslant \frac{TL}{2} \min\{1, \eta d L\},$$

which completes the proof. □

Proof of Proposition 2.3.5. Consider the linear loss

$$\ell_t(x) = L\langle g, x\rangle, \qquad g = -e_1 + \frac{1}{d-1} \sum_{i=2}^{d} e_i.$$

Note that $\|g\|_\infty \leqslant 1$, so ℓ_t is L-Lipschitz with respect to the ℓ_1 norm.

Claim. For all

$$t \leqslant T_0 := \Big\lfloor \frac{d-1}{d} \cdot \frac{1}{L\eta} \Big\rfloor,$$

the OGD iterates satisfy

$$x_{t+1} = \Big[\frac{1}{d} + tL\eta,\ \frac{1}{d} - \frac{tL\eta}{d-1},\ \ldots,\ \frac{1}{d} - \frac{tL\eta}{d-1}\Big].$$

Proof of the claim. For $t \leqslant T_0$, we have

$$\eta \leqslant \frac{d-1}{d} \cdot \frac{1}{Lt}.$$

Base case ($t = 1$). Starting from $x_1 = (1/d, \ldots, 1/d)$, we compute

$$x_1 - L\eta g = \Big[\frac{1}{d} + L\eta,\ \frac{1}{d} - \frac{L\eta}{d-1},\ \ldots,\ \frac{1}{d} - \frac{L\eta}{d-1}\Big].$$

All coordinates are nonnegative since

$$\frac{1}{d} - \frac{L\eta}{d-1} \geqslant 0,$$

and the sum of the coordinates equals

$$\frac{1}{d} + L\eta + (d-1)\Big(\frac{1}{d} - \frac{L\eta}{d-1}\Big) = 1.$$

Hence $x_1 - L\eta g \in \Delta_{d-1}$, and no projection is needed:

$$x_2 = x_1 - L\eta g.$$

Inductive step. Assume that for some $t \leqslant T_0$,

$$x_t = \Big[\frac{1}{d} + (t-1)L\eta,\ \frac{1}{d} - \frac{(t-1)L\eta}{d-1},\ \ldots,\ \frac{1}{d} - \frac{(t-1)L\eta}{d-1}\Big].$$

Then

$$x_t - L\eta g = \Big[\frac{1}{d} + tL\eta,\ \frac{1}{d} - \frac{tL\eta}{d-1},\ \ldots,\ \frac{1}{d} - \frac{tL\eta}{d-1}\Big].$$

For $t \leqslant T_0$, we have

$$\frac{1}{d} - \frac{tL\eta}{d-1} \geqslant \frac{1}{d} - \frac{1}{d} = 0,$$

and the sum of the coordinates is

$$\frac{1}{d} + tL\eta + (d-1)\Big(\frac{1}{d} - \frac{tL\eta}{d-1}\Big) = 1.$$

Thus $x_t - L\eta g \in \Delta_{d-1}$ and

$$x_{t+1} = x_t - L\eta g,$$

which completes the induction.

Regret lower bound. Since $\ell_t(e_1) = -L$, we have

$$\begin{aligned} R_T &= \sum_{t=1}^{T} \big(\ell_t(x_t) - \ell_t(e_1)\big) \\ &\geqslant LT_0 + L\sum_{t=1}^{T_0} \langle g, x_t\rangle. \end{aligned}$$

Using the explicit form of x_t, we compute

$$\langle g, x_t \rangle = -\Big(\frac{1}{d} + (t-1)L\eta\Big) + \Big(\frac{1}{d} - \frac{(t-1)L\eta}{d-1}\Big) = -\frac{d}{d-1}L\eta(t-1).$$

Therefore,

$$\begin{aligned} R_T &\geqslant LT_0 - \frac{d}{d-1}L^2\eta\sum_{t=1}^{T_0}(t-1) \\ &= LT_0 - \frac{1}{2}\frac{d}{d-1}L^2\eta T_0(T_0-1). \end{aligned}$$

Using $T_0 \leqslant \frac{d-1}{d\,\eta L}$, we conclude

$$R_T \geqslant LT_0 - \frac{1}{2}LT_0 = \frac{1}{2}LT_0 = \frac{L}{2}\Big\lfloor\frac{d-1}{d}\cdot\frac{1}{\eta L}\Big\rfloor \geqslant \frac{1}{4\eta},$$

where the final inequality holds for $\eta \leqslant dL$ and $d \geqslant 4$. □

2.4 Follow-the-Regularised-Leader

2.4.1 Online Mirror Descent as Online Gradient Descent in Dual Space

In the previous section, we introduced OMD as a proximal update which minimises a first-order linear approximation of the current loss with a penalisation for points that are far from the current iterate. There is an alternative formulation for OMD, which is the one originally introduced for offline optimisation by Nemirovskij and Yudin (1983).

Consider the regulariser ψ used by an instance of OMD and its restriction to the action set V: $\psi_V(x) = \psi(x) + i_V$ where i_V is the indicator function of the set V:

$$i_V(x) = \begin{cases} 0, & x \in V \\ +\infty, & x \notin V \end{cases}$$

Its Fenchel conjugate is defined as

$$\psi_V^\star(y) = \sup_{x\in\mathbb{R}^d}\Big\{\langle y, x\rangle - \psi_V(x)\Big\}.$$

It can be shown (see Section 6.4.2 in Orabona (2019) for details) that when the Fenchel conjugate

is differentiable, the OMD update is equivalent to

$$x_{t+1} = \nabla\psi_V^\star\Big(\nabla\psi(x_t) - \eta_t g_t\Big). \qquad \text{(OMD update)}$$

Under this formulation, we can interpret OMD as performing a gradient step in the dual space to which gradients and linear maps usually belong (in our case the distinction between primal and dual space is only conceptual since all points belong to $\mathbb{R}^d$). The gradient of the regulariser $\nabla\psi$ acts as a mapping from the primal to the dual space and $\nabla\psi_V^\star$ acts as the inverse mapping back into the primal space.

This formulation highlights OGD as a canonical instance of OMD since when the regulariser $\psi(x) = \frac{1}{2}\|x\|_2^2$ is the squared Euclidean norm, the mapping into the dual space $\nabla\psi(x) = x$ is the identity.

2.4.2 Follow-the-Regularised-Leader - The Algorithm

The dual-space formulation of Online Mirror Descent provides a natural motivation for a closely related algorithm known as *Follow-the-Regularised-Leader* (FTRL). While OMD and FTRL are deeply connected and often enjoy comparable theoretical guarantees, they differ in how past gradient information is aggregated and used to generate future decisions.

OMD performs a gradient step in the dual space at each round and then maps the resulting point back to the primal space via the mirror map, producing the next iterate x_{t+1}. Importantly, this projected primal point is then used as the starting point for the next dual update. In contrast, FTRL accumulates gradient information directly in the dual space across rounds and only applies the mirror map when producing the action to be played. In other words, FTRL continues the descent in the dual space without resetting to the projected primal iterate after each round. The FTRL update can be written as

$$x_{t+1} = \nabla\psi_V^\star\Big(\nabla\psi(x_1) - \eta_t \sum_{s=1}^{t} g_s\Big).$$

More generally, FTRL allows the use of time-varying regularisers ψ_{t+1}, with the step sizes η_t

absorbed into the regularisation. This yields the update

$$x_{t+1} = \nabla\psi^{\star}_{V,t+1}\Big(-\sum_{s=1}^{t} g_s\Big). \qquad \text{(FTRL update)}$$

Using Fenchel duality, this update can equivalently be expressed in a primal, proximal-formulation as

$$x_{t+1} = \operatorname*{arg\,min}_{x\in V}\left\{\left\langle \sum_{s=1}^{t} g_s, x \right\rangle + \psi_{t+1}(x)\right\}.$$

We emphasise that this corresponds to an instance of FTRL applied to the *linearised losses* $\langle g_t, x\rangle$ rather than the original losses $\ell_t(x)$. For convex losses, optimising over linearised losses yields the same worst-case performance guarantees as applying FTRL directly to the original losses. This linearised version is the version we adopt in this thesis.

Intuition behind the name. A natural strategy in online optimisation is to play the point that minimises the losses observed so far,

$$\operatorname*{arg\,min}_{x\in V} \sum_{s=1}^{t-1} \ell_s(x),$$

the so-called *Follow-the-Leader* (FTL) algorithm. However, as already suggested by the intuition for OGD, this strategy can be highly unstable: it may react too strongly to recent losses, causing the iterates to oscillate widely across the action space. An adversary can exploit this sensitivity by making a region appear favourable in the short term before reversing its losses, leading to large regret. FTRL addresses this issue by explicitly incorporating stability through regularisation of the FTL algorithm. Instead of exactly following the minimiser of past losses, it follows the *regularised* leader, balancing cumulative loss minimisation with a preference for stable, well-behaved decisions.

Comparison to Online Mirror Descent. When the iterates remain in the interior of the action space, constant step-size OMD and FTRL with fixed regulariser coincide. In particular, the lower bound of Proposition 2.3.3 extends to FTRL almost immediately. Beyond this, FTRL is more robust than OMD because its regret bound does not depend on the maximum Bregman divergence between the evolving iterates and the comparator, which it does for OMD. This dependence can be a genuine weakness: there exist settings in which OMD suffers linear regret,

while FTRL continues to achieve sublinear regret (Orabona and Pál, 2018). An example is the simplex with negative entropy regularisation, where the associated Bregman divergence (the Kullback–Leibler divergence) becomes unbounded as points approach the boundary. In such cases, the reliance of OMD on the maximum Bregman divergence is prohibitive for meaningful regret bounds.

2.5 Roadmap for the Rest of the Thesis

The material introduced in this chapter provides the common language and algorithmic primitives used throughout much of the remainder of the thesis. Online Mirror Descent and Follow-the-Regularised-Leader serve as the foundational templates for most gradient-based algorithms we analyse. While the presentation here is deliberately general, later chapters specialise this framework to a particular setting (see Table 2.1).

Chapter	**Setting**	**Algorithm**	**Regulariser**
Chapter 4	Online Convex Optimisation	FTRL	$\frac{1}{r}\|x\|_r^r$, $r \geqslant 2$
Chapter 6	Stochastic Shortest Path	OMD	Negative entropy & $\|x\|_r^r$, $1 \leqslant r \leqslant 2$
Chapter 8	Reinforcement Learning	OMD variant	General

Table 2.1: Overview of how gradient-based methods are instantiated across the thesis.

Chapter 5 is devoted exclusively to lower bounds, while Chapter 7 studies a gradient-based algorithm that does not directly fall within the OMD framework, though it remains closely related (see Section 3.2.2).

Chapter 3

Preliminaries on Reinforcement Learning

This chapter introduces the reinforcement learning framework considered in Chapters 5, 6, 7, and 8. We discuss the connection between gradient-based methods in reinforcement learning and the mirror-descent framework introduced in the previous chapter in Section 3.2.2. However, we begin by presenting the foundational model of reinforcement learning, namely Markov Decision Processes, and associated concepts that will play a central role in subsequent chapters.

3.1 Markov Decision Processes

A discounted Markov Decision Process (MDP) (Puterman, 1994) is a discrete-time stochastic process, comprised of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a discount factor $\gamma \in [0, 1)$ and for each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ a next-state transition function $p(\cdot|s, a) \in \Delta(\mathcal{S})$ and a (assumed here deterministic) reward function $r(s, a) \in [0, 1]$. $\Delta(\mathcal{X})$ denotes the probability simplex over a set $\mathcal{X}$. The tabular setting corresponds to the case when both $\mathcal{S}$ and $\mathcal{A}$ are finite, which we assume throughout this chapter to present results in a canonical setting. In Chapter 5, we consider non-tabular MDPs.

Policies. In a state $s \in \mathcal{S}$, an agent chooses an action $a \in \mathcal{A}$, which gives them a reward $r(s, a)$ and transitions them to a new state according to the transition function $p(\cdot|s, a)$. Once they are in a new state, the process continues. The actions chosen by an agent are formalised through policies. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a mapping from a state to a distribution over actions. We

will often write it as an element in $\Pi = \Delta(\mathcal{A})^{|\mathcal{S}|}$. In each state $s \in \mathcal{S}$, an agent following policy π chooses an action $a \in \mathcal{A}$ according to $\pi_s = \pi(\cdot|s) \in \Delta(\mathcal{A})$. If a policy π is deterministic, we sometimes write $\pi(s) \in \mathcal{A}$ to denote the single action that the policy selects in state s.

Value and Action-Value Functions. The performance of a policy in an MDP is measured with respect to the value function $V^\pi : \mathcal{S} \to \mathbb{R}$,

$$V^\pi(s) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \pi, s_0 = s\Big],$$

where s_t, a_t are the state and action in time-step t and the expectation is with respect to both the randomness in the transitions and the choice of actions under policy π. This is a notion of long-term reward that describes the discounted rewards accumulated over future time-steps when following policy π and starting in state s. For a distribution over starting states $\rho \in \Delta(\mathcal{S})$, we write $V^\pi(\rho) = \sum_{s\in\mathcal{S}} \rho(s) V^\pi(s)$ for the expected value when starting in a state distributed according to ρ. It is also useful to work with the state-action value $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$Q^\pi(s, a) = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | \pi, s_0 = s, a_0 = a\Big],$$

which is similar to V^π, with the additional constraint of taking action a in the first time-step. We will often write $V^\pi \in \mathbb{R}^{|\mathcal{S}|}$ (resp. $Q^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$) to refer to the vector form, where each entry represents the value (resp. action-value) in that state (resp. state-action pair). Similarly, we write $Q^\pi_s \in \mathbb{R}^{|\mathcal{A}|}$ for the vector of action-values in state s. Note that since the rewards are bounded in $[0, 1]$, the value and action-value of a policy are bounded in $[0, 1/(1-\gamma)]$.

Optimal Policy. One of the main aims of reinforcement learning is to find a policy π that maximises V^π. Under certain conditions on the state and action space, it is known that there exists a deterministic policy that simultaneously maximises V^π and Q^π for all states and actions [Puterman (1994), Theorem 6.2.12]. We call such a policy an optimal policy and denote it by $\pi^\star$. We will also denote $V^{\pi^\star} = V^\star$ and $Q^{\pi^\star} = Q^\star$.

Bellman Equations. The following expressions relate Q^π and V^π in terms of each other, and when combined give the Bellman equations (Bellman, 1957):

$$V^\pi(s) = \langle Q^\pi_s, \pi_s \rangle = \sum_{a\in\mathcal{A}} \pi(a \mid s) Q^\pi(s, a),$$

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^\pi(s').$$

Substituting V^π in terms of Q^π or vice versa gives the fully expanded forms:

$$\begin{aligned} V^\pi(s) &= \sum_{a \in \mathcal{A}} \pi(a \mid s) r(s,a) + \gamma \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^\pi(s'), \\ Q^\pi(s,a) &= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \sum_{a' \in \mathcal{A}} \pi(a' \mid s') Q^\pi(s',a'). \end{aligned} \tag{3.1}$$

Bellman Operators. For a policy π, the *Bellman evaluation operator* $\mathcal{T}^\pi$ acts on action-value functions as

$$(\mathcal{T}^\pi Q)(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \sum_{a' \in \mathcal{A}} \pi(a' \mid s') Q(s',a').$$

The *Bellman optimality operator* $\mathcal{T}^\star$ is defined as

$$(\mathcal{T}^\star Q)(s,a) := r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \max_{a' \in \mathcal{A}} Q(s',a').$$

Both Q^π and $Q^\star$ are the unique fixed points of $\mathcal{T}^\pi$ and $\mathcal{T}^\star$, respectively (Bellman, 1957; Agarwal et al., 2022).

Contraction Property. Both $\mathcal{T}^\pi$ and $\mathcal{T}^\star$ are *contraction operators* in the sup-norm $\|\cdot\|_\infty$ with contraction factor $\gamma \in [0,1)$ (Bellman, 1957; Agarwal et al., 2022). Specifically, for any two action-value functions $Q_1, Q_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$:

$$\begin{aligned} \|\mathcal{T}^\pi Q_1 - \mathcal{T}^\pi Q_2\|_\infty &\leqslant \gamma \|Q_1 - Q_2\|_\infty, \\ \|\mathcal{T}^\star Q_1 - \mathcal{T}^\star Q_2\|_\infty &\leqslant \gamma \|Q_1 - Q_2\|_\infty. \end{aligned} \tag{3.2}$$

Proof. For the evaluation operator $\mathcal{T}^\pi$, consider any (s,a):

$$\begin{aligned} |(\mathcal{T}^\pi Q_1)(s,a) - (\mathcal{T}^\pi Q_2)(s,a)| &= \left| \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') \big(Q_1(s',a') - Q_2(s',a')\big) \right| \\ &\leqslant \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') |Q_1(s',a') - Q_2(s',a')| \\ &\leqslant \gamma \|Q_1 - Q_2\|_\infty \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s') \\ &= \gamma \|Q_1 - Q_2\|_\infty, \end{aligned}$$

where the last equality holds because p and π are distributions. Taking the supremum over (s, a) gives the contraction property for $\mathcal{T}^\pi$.

For the optimality operator $\mathcal{T}^\star$:

$$\begin{aligned}|(\mathcal{T}^\star Q_1)(s,a) - (\mathcal{T}^\star Q_2)(s,a)| &= \gamma \left| \sum_{s'} p(s'|s,a) \big(\max_{a'} Q_1(s',a') - \max_{a'} Q_2(s',a') \big) \right| \\ &\leqslant \gamma \sum_{s'} p(s'|s,a) \max_{a'} |Q_1(s',a') - Q_2(s',a')| \\ &\leqslant \gamma \|Q_1 - Q_2\|_\infty,\end{aligned}$$

and taking the supremum over (s, a) again establishes the contraction property. □

This contraction property gives an efficient procedure to compute the action-value function of any policy or the optimal action-value function: starting from any initial Q_0, iteratively applying $\mathcal{T}^\pi$ or $\mathcal{T}^\star$ converges geometrically fast to the unique fixed points Q^π or $Q^\star$, respectively. The latter case gives rise to the value iteration algorithm (see Section 1.3.1 in Agarwal et al. (2022)).

Computing Value and Action-Value Functions. We have just seen that the Bellman evaluation operator gives rise to an efficient procedure to compute the value function of any fixed policy π. However, it also admits a closed-form characterization as the solution to a system of linear equations. Let $P_\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ denote the transition matrix induced by policy π, with entries

$$(P_\pi)_{s,s'} = \sum_{a \in \mathcal{A}} \pi(a \mid s)\, p(s' \mid s, a),$$

and let $r_\pi \in \mathbb{R}^{|\mathcal{S}|}$ be the expected reward vector defined by

$$r_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a \mid s)\, r(s, a).$$

Using the Bellman equations in (3.1), the value function V^π satisfies

$$V^\pi = r_\pi + \gamma P_\pi V^\pi.$$

Since $\gamma \in [0, 1)$, the matrix $I - \gamma P_\pi$ is invertible, and hence

$$V^\pi = (I - \gamma P_\pi)^{-1} r_\pi. \tag{3.3}$$

The corresponding action-value function Q^π can then be recovered from V^π via

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V^\pi(s').$$

Computing the values of policies exactly will play a central role in the planning algorithms introduced in Section 3.2 and studied in Chapter 8.

Visitation Distribution. We now define the discounted visitation-distribution for starting state s' and policy π,

$$d^\pi_{s'}(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^\pi(s_t = s | s_0 = s'), \tag{3.4}$$

which plays a central role in the analysis of policy gradient methods and, in particular, in understanding distribution mismatch phenomena that are a key focus of Chapter 8. Note that $\mathbb{P}^\pi(s_t = s|s_0 = s')$ is the probability of being in state s at time t when starting in state s' and following policy π. We also write $d^\pi_\rho(s) = \sum_{s' \in \mathcal{S}} \rho(s') d^\pi_{s'}(s)$.

Stochastic Shortest Path. In Chapter 6, we study the stochastic shortest path (SSP) problem, which can be viewed as an undiscounted Markov Decision Process (i.e., $\gamma = 1$) augmented with a designated terminal (goal) state. Once the goal state is reached, the interaction with the environment terminates and no further costs or rewards are incurred. The objective in an SSP is to minimise the expected cumulative cost incurred before reaching the goal state, starting from a given initial state. Unlike discounted MDPs, the absence of discounting requires additional structural assumptions, such as proper policies that reach the goal with probability one, to ensure that value functions are well defined and finite. We defer a formal definition of the model, along with these assumptions and their consequences, to Chapter 6.

Multi-Armed Bandits and Online Learning. The classical multi-armed bandit (MAB) problem (Lai and Robbins, 1985; Auer et al., 2002; Lattimore and Szepesvári, 2020) can be viewed as a one-state Markov Decision Process, where an agent sequentially chooses among A actions (or "arms") to maximize cumulative reward. Each arm is associated with an unknown reward distribution, and the agent must balance *exploration* (trying suboptimal arms to learn their rewards) with *exploitation* (choosing the empirically best arm). In this sense, MABs are a special case of reinforcement learning with $|\mathcal{S}| = 1$ and no state transitions, yet they capture many of the core challenges of sequential decision-making under uncertainty. There also exists

an *adversarial* variant, in which the rewards are not stochastic but can change arbitrarily across rounds. In Chapter 7, we study gradient-based methods for stochastic multi-armed bandits.

Online learning (Shalev-Shwartz, 2012; Hazan, 2016; Orabona, 2019) generalizes the multi-armed bandit problem to arbitrary convex action spaces. In this framework, an agent iteratively selects actions and suffers a loss function chosen by the environment, which may be stochastic or even adversarial. Depending on the feedback model, the agent may observe only the loss associated with the action played (bandit feedback) or the entire loss function (full-information feedback). In Chapter 4, we focus on gradient-based methods for online learning under full-information feedback, i.e., online convex optimization.

3.2 Planning in Markov Decision Processes

In this section, we focus on the *planning* problem in reinforcement learning, where the objective is to compute an optimal policy $\pi^\star$ that maximises the value function. We assume the transition dynamics $p(\cdot \mid s, a)$ and reward function r of the MDP are known and that the value and action-value functions of any policy can be computed exactly via (3.3). This is also the setting considered in Chapter 8 where we are particularly interested in policy gradient algorithms, which fall under the category of *policy-based* methods. To set a point of reference, we begin by presenting Policy Iteration, a classical policy-based planning algorithm with strong theoretical guarantees.

3.2.1 Policy Iteration

Policy Iteration is a classical algorithm for solving discounted MDPs exactly in the tabular setting (Howard, 1960; Puterman, 1994; Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 2018). It alternates between two steps: policy evaluation and policy improvement.

Policy Evaluation. Given a policy π^k, compute its action-value function Q^{π^k} using (3.3).

Policy Improvement. Construct a new policy π^{k+1} by acting greedily with respect to Q^{π^k}:

$$\pi^{k+1}(s) \in \arg\max_{a \in \mathcal{A}} Q^{\pi^k}(s, a), \quad \forall s \in \mathcal{S}.$$

By construction, π^{k+1} is deterministic; in the presence of ties in the argmax, they are broken arbitrarily.

Policy Iteration Update. The Policy Iteration algorithm can be written compactly as the recursion

$$\pi^{k+1} \in \arg\max_{\pi \in \Pi} \left\langle Q_s^{\pi^k}, \pi_s \right\rangle, \quad \forall s \in \mathcal{S}, \tag{3.5}$$

where Q^{π^k} is obtained via exact policy evaluation of π^k.

Intuition. Policy Iteration works because acting greedily with respect to the action-value function of any *sub-optimal* policy produces a new policy that is guaranteed to be strictly better in at least one state. In the tabular setting, there are only finitely many policies, so repeated application of this procedure must eventually reach the optimal policy.

We now present a formal characterisation of the convergence of policy iteration. The following results follow the analysis in Agarwal et al. (2022) (Section 1.3.2).

Convergence Analysis.

Lemma 3.2.1 (Policy Improvement)**.** *Let π^k be the policy at iteration k of Policy Iteration, and let π^{k+1} be the policy greedy with respect to Q^{π^k}. Then the following component-wise inequalities hold:*

$$Q^{\pi^k} \leqslant \mathcal{T} Q^{\pi^k} \leqslant Q^{\pi^{k+1}}.$$

Proof. Let (s, a) be any state-action pair. Recall the Bellman optimality operator:

$$(\mathcal{T} Q^{\pi^k})(s, a) = r(s, a) + \gamma \sum_{s'} p(s'|s, a) \max_{a'} Q^{\pi^k}(s', a'),$$

and by definition, π^{k+1} is greedy w.r.t. Q^{π^k}:

$$\pi^{k+1}(s) \in \arg\max_{a} Q^{\pi^k}(s, a), \quad \forall s.$$

Step 1: $\mathcal{T} Q^{\pi^k} \geqslant Q^{\pi^k}$.

$$\begin{aligned}
(\mathcal{T} Q^{\pi^k})(s, a) &= r(s, a) + \gamma \sum_{s'} p(s'|s, a) \max_{a'} Q^{\pi^k}(s', a') \\
&\geqslant r(s, a) + \gamma \sum_{s'} p(s'|s, a) Q^{\pi^k}(s', \pi^k(s')) \\
&= Q^{\pi^k}(s, a) \quad \text{(Bellman equation)}.
\end{aligned}$$

Step 2: $Q^{\pi^{k+1}} \geqslant Q^{\pi^k}$.

$$\begin{aligned}
Q^{\pi^k}(s,a) &= r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \sum_{a'} \pi^k(a' \mid s') Q^{\pi^k}(s',a') \quad \text{(Bellman equation)} \\
&\leqslant r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) \sum_{a'} \pi^{k+1}(a' \mid s') Q^{\pi^k}(s',a') \quad \text{(greedy policy)} \\
&= r(s,a) + \gamma (P^{\pi^{k+1}} Q^{\pi^k})(s,a) \\
&\leqslant \left[\sum_{t=0}^{\infty} \gamma^t (P^{\pi^{k+1}})^t r \right] (s,a) = Q^{\pi^{k+1}}(s,a).
\end{aligned}$$

Step 3: $Q^{\pi^{k+1}} \geqslant \mathcal{T} Q^{\pi^k}$.

$$\begin{aligned}
Q^{\pi^{k+1}}(s,a) &= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \sum_{a' \in \mathcal{A}} \pi^{k+1}(a' \mid s') Q^{\pi^{k+1}}(s',a') \quad \text{(Bellman equation)} \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) Q^{\pi^{k+1}}(s', \pi^{k+1}(s')) \quad \text{(deterministic policy)} \\
&\geqslant r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) Q^{\pi^k}(s', \pi^{k+1}(s')) \quad \text{(step 2)} \\
&= r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) \max_{a'} Q^{\pi^k}(s',a') \quad \text{(greedy policy)} \\
&= \mathcal{T} Q^{\pi^k}(s,a).
\end{aligned}$$

□

This lemma shows that each iteration of Policy Iteration produces a policy whose action-value function is component-wise greater than the application of the Bellman optimality operator to the previous policy. Combining this with the contraction property of the Bellman operator allows us to formally quantify the geometric convergence of the sequence of policies to the optimal policy.

Proposition 3.2.2 (Geometric Convergence of Policy Iteration)**.** *Let* $\{\pi^k\}$ *be the sequence of policies generated by Policy Iteration. Then*

$$\|Q^\star - Q^{\pi^k}\|_\infty \leqslant \gamma^k \|Q^\star - Q^{\pi^0}\|_\infty, \quad \|V^\star - V^{\pi^k}\|_\infty \leqslant \gamma^{k+1} \|V^\star - V^{\pi^0}\|_\infty,$$

where π^0 *is the initial policy.*

Proof. **Step 1: Contraction of the Bellman operator.** We have seen in (3.2) that the Bellman optimality operator $\mathcal{T}$ is a γ-contraction in ℓ_∞:

$$\|\mathcal{T} Q_1 - \mathcal{T} Q_2\|_\infty \leqslant \gamma \|Q_1 - Q_2\|_\infty, \quad \forall Q_1, Q_2.$$

Since $Q^\star$ is the unique fixed point of $\mathcal{T}$,

$$\|Q^\star - \mathcal{T}Q^{\pi^k}\|_\infty = \|\mathcal{T}Q^\star - \mathcal{T}Q^{\pi^k}\|_\infty \leqslant \gamma \|Q^\star - Q^{\pi^k}\|_\infty.$$

Step 2: Policy Improvement. By Lemma 3.2.1, $Q^\star \geqslant Q^{\pi^{k+1}} \geqslant \mathcal{T}Q^{\pi^k}$ component-wise. Combining with Step 1,

$$\|Q^\star - Q^{\pi^{k+1}}\|_\infty \leqslant \|Q^\star - \mathcal{T}Q^{\pi^k}\|_\infty \leqslant \gamma \|Q^\star - Q^{\pi^k}\|_\infty.$$

Step 3: Recursion. Applying this recursively yields

$$\|Q^\star - Q^{\pi^k}\|_\infty \leqslant \gamma^k \|Q^\star - Q^{\pi^0}\|_\infty.$$

The bound on V^{π^k} follows immediately from $V^\pi(s) = \langle Q^\pi_s, \pi_s \rangle$. □

This is a strong result as it implies that to obtain a ε-optimal policy, which satisfies $V^\star - V^\pi \leqslant \varepsilon$, it suffices to perform

$$\frac{1}{1-\gamma} \log \frac{1}{(1-\gamma)\varepsilon}$$

iterations of Policy Iteration. In particular, the iteration complexity scales logarithmically with the desired accuracy.

In Chapter 8, we establish a lower bound showing that the γ-rate in Proposition 3.2.2 is optimal for iterations $k \leqslant (S-1)/2$. To the best of our knowledge, this lower bound is new. While such a result may have been previously known for Policy Iteration, we were unable to find a proof in the literature. Existing lower bounds for Policy Iteration instead focus on exact convergence to the optimal policy (Hollanders et al., 2012), or consider undiscounted MDPs (Fearnley, 2010).

An asymptotic lower-bound is not achievable since Policy Iteration converges exactly in finitely many iterations due to the strict policy improvement step and the finite number of deterministic policies. Ye (2011) showed that Policy Iteration terminates in at most $S(A-1)\left\lceil \frac{S}{1-\gamma} \log \frac{S^2}{1-\gamma} \right\rceil$ iterations. This bound was subsequently improved to $(SA+1)\left\lceil \frac{1}{1-\gamma} \log \frac{S}{1-\gamma} \right\rceil$ by Hansen et al. (2010), and finally to $S(A-1)\left\lceil \frac{1}{1-\gamma} \log \frac{1}{1-\gamma} \right\rceil$ by Scherrer (2013). Recently, Liu et al. (2025) established a dimension-free bound of $\left\lceil \frac{1}{1-\gamma} \log \frac{3}{(1-\gamma)\Delta} \right\rceil$ iterations for exact convergence, where Δ is the minimum sub-optimality gap of the optimal policy across the MDP and may be arbitrarily

small. The strong guarantees above rely critically on exact policy evaluation; when evaluation is approximate, as in large-scale or online settings, the contraction-based analysis no longer applies directly (see Section 3.2.3).

While Policy Iteration enjoys strong geometric convergence guarantees, it relies critically on exact policy evaluation (see Section 3.2.3). This motivates the study of smoother alternatives that may be more robust in inexact settings, such as policy gradient methods discussed next.

3.2.2 Policy Gradient Methods

We now turn to *policy gradient* methods (Williams, 1992; Sutton et al., 1999; Konda and Tsitsiklis, 1999; Kakade, 2001), which are of central interest in this thesis, and in particular in Chapters 7 and 8. Like Policy Iteration, policy gradient methods directly operate in the space of policies. However, rather than performing exact greedy improvement with respect to the action-value function, they cast policy optimisation as a differentiable optimisation problem and apply gradient-based updates. Although policy gradient methods are most commonly used in large-scale or model-free settings, their behaviour is not fully understood even in discounted tabular MDPs.

Policy Parameterisation. The core idea of policy gradient methods is to consider parameterised policies, define a differentiable objective with respect to these parameters, and then perform gradient ascent to improve performance. In general, there are many ways to parameterise policies, including approaches that leverage function approximation and neural networks in large-scale settings (Schulman et al., 2015, 2021). We consider two parametrisations suited to the tabular setting below (direct and softmax).

Optimisation Objective. Given an initial state distribution $\mu \in \Delta(\mathcal{S})$, the performance of a policy π_θ is still measured by its value $V^{\pi_\theta}(\mu)$ but expressed as a function of its parameters

$$J(\theta) := V^{\pi_\theta}(\mu) = \sum_{s\in\mathcal{S}} \mu(s)\mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)|\pi_\theta, s_0 = s\Big],$$

The policy optimisation problem is then

$$\max_{\theta\in\mathbb{R}^{S\times A}} J(\theta).$$

Policy Gradient Theorem. A central result underpinning policy gradient methods is the *policy gradient theorem* (Williams, 1992; Sutton et al., 1999), which provides an explicit expression for the gradient of the performance objective $J(\theta)$:

$$\nabla_\theta J(\theta) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\mu^{\pi_\theta},\, a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(a \mid s)\right]. \tag{3.6}$$

Policy Gradient Update. The policy gradient method considered in this section performs gradient ascent on the objective $J(\theta)$, with the gradient evaluated at the current parameter vector θ^k:

$$\theta_{s,a}^{k+1} \;\leftarrow\; \theta_{s,a}^{k} + \eta\,\frac{\partial J(\theta)}{\partial \theta_{s,a}}\Big|_{\theta=\theta^k}, \quad \forall s \in \mathcal{S},\ a \in \mathcal{A},$$

where $\eta > 0$ denotes a learning rate.

Convergence Analysis. The convergence properties of policy gradient methods have been the subject of significant recent interest. Even in the tabular, discounted setting with exact gradients, the analysis is non-trivial, as the objective $J(\theta)$ is generally *non-concave* in the policy parameters (Agarwal et al., 2021). As a consequence, standard tools from convex optimisation or from the analysis of Online Mirror Descent methods discussed in Chapter 2 do not apply directly.

Direct Parameterisation. In the tabular setting, a natural choice of parameterisation is the *direct parameterisation*, in which a separate parameter is associated with each state–action pair. Concretely, we write $\pi_\theta(a \mid s) = \theta_{s,a}$, so that a policy over S states and A actions is described by $S \times A$ parameters. This simple parameterisation allows us to study policy gradient methods in a fully tractable setting while preserving the essential features of gradient-based policy optimisation. For this parameterisation, the policy gradient theorem (3.6) simplifies to

$$\frac{\partial J(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma}\, d_\mu^{\pi_\theta}(s)\, Q^{\pi_\theta}(s,a).$$

Since $\pi_\theta(\cdot \mid s)$ must lie in the probability simplex $\Delta(\mathcal{A})$ for each state s, the policy gradient update must be followed by a projection step (with respect to the Euclidean norm). The resulting algorithm is therefore an instance of *projected gradient ascent*.

Global convergence guarantees for this method with exact gradients have been established for

optimisation with respect to $V^{\pi}(\mu)$, with performance evaluated under $V^{\pi}(\rho)$. Agarwal et al. (2021) first proved convergence at a $1/\sqrt{k}$ rate, which was subsequently improved to a $1/k$ rate by Zhang et al. (2020a), Xiao (2022), and Liu et al. (2025). Most recently, Liu et al. (2024) established *linear* (geometric) convergence, achieving a β^k rate for some $\beta < 1$. In addition, Liu et al. (2025) showed that the method converges *exactly* to an optimal policy in a finite number of iterations. As in the case of Policy Iteration, these strong guarantees critically rely on access to exact policy gradients.

A notable feature of all these bounds is the presence of *distribution mismatch* coefficients such as $\left\| \frac{d_{\rho}^{\pi^\star}}{\mu} \right\|_{\infty}^2$, which quantify the discrepancy between the state distribution induced by the optimal policy and the distribution μ used to define the optimisation objective. Such mismatch factors are common in the analysis of policy gradient methods. Understanding how to control or eliminate this dependence will be a key focus of Chapter 8, and also motivates the distinction between the distribution μ used to optimise the policy and the distribution ρ on which performance is ultimately evaluated (e.g. ideally we would use $\mu = d_{\rho}^{\pi^\star}$).

Softmax Parameterisation. The direct parameterisation studied above requires an explicit projection step to ensure that the policies remain in the probability simplex. An alternative approach is the *softmax parameterisation*, in which for each state-action pair (s, a), we associate a parameter $\theta_{s,a} \in \mathbb{R}$ and define the policy as

$$\pi_{\theta}(a \mid s) = \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{s,a'})}, \qquad \forall s \in \mathcal{S},\ a \in \mathcal{A}.$$

This parameterisation ensures that $\pi_{\theta}(\cdot \mid s) \in \Delta(\mathcal{A})$ automatically for every s without requiring any projection, allowing the use of tools from standard unconstrained optimisation.

For the softmax parameterisation, the policy gradient theorem (3.6) simplifies to

$$\frac{\partial J(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_{\mu}^{\pi_\theta}(s)\, \pi_{\theta}(a \mid s) \Big(Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s) \Big). \tag{3.7}$$

While the softmax parameterisation avoids the need for projection, it introduces a fundamental difficulty: the optimal policy in a discounted MDP is deterministic, whereas the softmax policy class can only represent deterministic policies in the limit of the parameters going to infinity. Consequently, global optima of $J(\theta)$ are not attained at finite parameter values, and convergence

analysis must account for diverging parameters.

The convergence behaviour of policy gradient methods under the softmax parameterisation has also been studied extensively in recent years. Agarwal et al. (2021) established asymptotic convergence. This was later to a convergence rate of $1/k$ for sufficiently small step sizes by Mei et al. (2020), and subsequently extended to all constant step sizes by Liu et al. (2024). Stronger guarantees were later obtained with *linear* (geometric) convergence, i.e. a β^k rate for some $\beta < 1$ (Mei et al., 2021; Liu et al., 2024). As in the direct parameterisation, all of these results involve distribution mismatch coefficients. Despite these positive guarantees, Li et al. (2023a) showed that the associated convergence rates depend on problem-dependent constants that can be exponentially large in the size of the MDP. Due to the nature of the softmax parameterisation, exact convergence results in finite iterations are not possible.

Connection to Online Mirror Descent. Under the direct parameterisation, the policy gradient update is equivalent to the offline version of Online Gradient Descent (see Section 2.2) applied to the policy space Π, in which the policy gradient objective $J(\theta)$ is used as the loss at every iteration. This correspondence, however, does not hold for more general policy parameterisations since in Online Mirror Descent, gradients are taken with respect to the point / policy produced by the algorithm, which only coincides with the policy gradient update in the case of direct parameterisation. The two frameworks could be unified by considering how a non-direct parameterisation affects the policy gradient. In Chapter 8, we consider an algorithm that explicitly combines the online mirror descent framework with the policy gradient method under direct parameterisation.

Beyond Vanilla Policy Gradient. A wide variety of policy gradient variants have been proposed to improve convergence, stability, and sample efficiency. One prominent example is the *Natural Policy Gradient* (NPG) method with the softmax parameterisation (Kakade, 2001) . NPG is a preconditioned policy gradient method: it scales the standard gradient by the inverse of the Fisher information matrix. This preconditioning has significant theoretical implications. For instance, Agarwal et al. (2021) showed that NPG achieves a $1/k$ convergence rate that is *independent* of any distribution mismatch coefficients. Moreover, Khodadadian et al. (2021) established that NPG can attain *linear* (geometric) convergence at a γ-rate comparable to Policy Iteration. In Chapter 8, we explore this type of preconditioned policy gradient methods in detail to achieve strong convergence guarantees without incurring dependence on distribution mismatch

coefficients.

3.2.3 Beyond Exact Algorithms

In many practical settings, exact computation of value functions or policy gradients is infeasible due to the size of the state-action space, function approximation or incomplete knowledge of the MDP. This motivates the study of *inexact* planning algorithms, which use approximate evaluations in place of exact quantities.

Inexact Policy Iteration. When either Q^{π^k} or the corresponding greedy policy π^{k+1} cannot be computed exactly, classical Policy Iteration is no longer directly applicable. In such cases, *Approximate Policy Iteration* (API) algorithms (Bertsekas and Tsitsiklis, 1996; Bertsekas, 2011) are typically employed. The standard API approach simply constructs the greedy policy with respect to an estimated action-value function. However, this leads to unstable oscillatory behaviours acting fully greedily is highly sensitive to the accuracy of the estimates. As a result, many approaches have been proposed to stabilize API (Scherrer, 2014b; Kakade and Langford, 2002b; Lagoudakis and Parr, 2003; Munos, 2005; Metelli et al., 2021).

Inexact Policy Gradient. In contrast to policy iteration, policy gradient methods may be naturally better suited to the inexact setting since they move gradually through the parameter space, providing more stable improvement even when only approximate gradient information is available. In practice, this gradient can be estimated from data using Monte Carlo rollouts or other sampling-based procedures, leading to *stochastic policy gradient* methods such as REINFORCE (Williams, 1992; Sutton et al., 1999; Konda and Tsitsiklis, 1999). Despite their popularity, the theoretical understanding of these methods remains limited, even in very simple settings such as single-state bandits (see Chapter 7 and Lattimore and Szepesvári (2020)) or tabular MDPs. In particular, even under a generative model that allows sampling from arbitrary state–action pairs, the best known sample complexity guarantees for policy gradient methods in tabular environments are still far from optimal and significantly weaker than those available for more classical planning algorithms, as we further discuss in Chapter 8.

Part I

High Dimensional Problems are Fundamentally Different

Chapter 4

Optimal Anytime Online Learning on Lp-Balls

Online Convex Optimisation (OCO) (Gordon, 1999; Zinkevich, 2003) models a sequential decision-making problem in which, at each round $t = 1, \dots, T$, a learner selects a point x_t from a convex decision set $V \subset \mathbb{R}^d$ and subsequently incurs a convex loss $\ell_t(x_t)$. After observing the full loss function ℓ_t, the learner updates its decision for the next round according to a learning algorithm. Performance is measured through the regret with respect to the best fixed decision in hindsight,

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) - \min_{u \in V} \sum_{t=1}^{T} \ell_t(u).$$

A central goal in OCO is to design algorithms that achieve optimal regret guarantees under minimal assumptions on the losses and the geometry of the decision set. It is well understood that the optimal regret rates depend critically on both the structure of V and constraints on the losses (Levy and Duchi, 2019; Kerdreux et al., 2021a,c; Tsuchiya and Ito, 2024). A classical and widely studied approach for the learning algorithm in this setting is the Follow-The-Regularised-Leader (FTRL) family of algorithms (see Chapter 2 and Abernethy et al. (2008); Hazan and Kale (2008); Abernethy et al. (2012)). FTRL is particularly appealing because its performance is tightly controlled by the choice of regulariser, allowing it to adapt to the geometry of the decision set and achieve optimal regret guarantees in a wide variety of regimes. We focus on FTRL but the results also hold for Online Mirror Descent (see Section 4.6).

In this chapter, we focus on the setting where the action set is an ℓ_p-ball,

$$V = \{x \in \mathbb{R}^d : \|x\|_p \leqslant 1\},$$

for $p > 2$, and the losses are L-Lipschitz with respect to the ℓ_p norm. This setting has attracted sustained interest in optimisation and online learning, as ℓ_p-balls with $p > 2$ interpolate between strongly curved and nearly flat geometries, and serve as canonical examples for understanding optimisation beyond Euclidean spaces (d'Aspremont et al., 2018; Levy and Duchi, 2019; Guzmán and Nemirovski, 2015; Kerdreux et al., 2021b,c; Mayekar and Tyagi, 2020; Mendelson, 2021).

Regret minimisation in this setting has predominantly been considered in a low-dimensional regime where the time horizon T is assumed to be larger than the dimension d. In this regime, the optimal regret scales with $LT^{1/2}d^{1/2-1/p}$ (Levy and Duchi, 2019) and can be achieved using FTRL with a strongly convex regulariser. In a high-dimensional regime where $d \gg T$, this bound becomes vacuous. Indeed, the Lipschitz condition on the losses ensures that $g_t \in \partial\ell_t(x_t)$ satisfies $\|g_t\|_{p_\star} \leqslant L$ where $p_\star$ is the dual of p and we immediately obtain a trivial dimension-independent bound on the regret,

$$R_T = \max_{u \in V} \sum_{t=1}^{T} \Big\{ \ell_t(x_t) - \ell_t(u) \Big\} \leqslant \max_{u \in V} \sum_{t=1}^{T} \langle g_t, u - x_t \rangle \leqslant \max_{u \in V} \sum_{t=1}^{T} \|g_t\|_{p_\star} \|u - x_t\|_p \leqslant 2LT.$$

As the dimension d increases, the bound $O(LT^{1/2}d^{1/2-1/p})$ eventually exceeds the trivial upper bound $2LT$ and ceases to be meaningful. While it is not obvious that sub-linear regret should remain achievable as the dimension increases and the problem becomes fundamentally harder, a distinctive feature of OCO on ℓ_p-balls with $p > 2$ is that sub-linear regret is always achievable. However, there exist qualitatively different regimes determined by the relationship between d and T, which determine which rates are possible. In particular, there are exactly two dimension regimes, each characterised by a different optimal rate and a different form of regularisation used by FTRL (see Table 4.1 and Figure 4). To summarize,

- In the *low-dimensional regime* where $d < T$, the optimal regret scales as $L\sqrt{Td^{1-2/p}}$ and this can be achieved by FTRL with a strongly convex regulariser.
- In contrast, in the *high-dimensional regime* where $d \geqslant T$, the optimal regret becomes dimension-independent and scales as $LT^{1-1/p}$; this rate is achieved by FTRL with a uniformly convex regulariser of degree p (Sridharan and Tewari, 2010; Nemirovskij and Yudin,

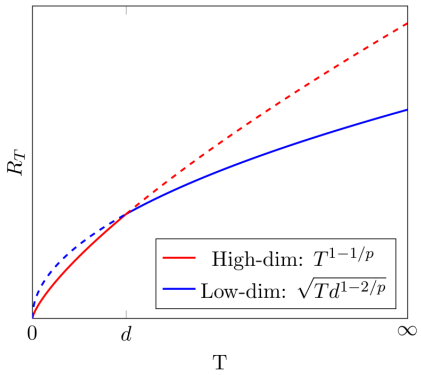

Figure 4.1: Optimal rates of regret for OCO on ℓ_p-balls with $p > 2$.

1983; Guzmán and Nemirovski, 2015; Guzmán, 2015; Sridharan and Tewari, 2010; Orabona, 2019; Bhaskara et al., 2020).

Thus, while FTRL achieves optimality across the two regimes, it is through fundamentally different types of regularisation. In practice, however, the learner typically does not know the time horizon T in advance and therefore cannot determine at the outset whether the problem will fall into the low- or high-dimensional regime. We show that anytime optimality where the optimal regret is obtained simultaneously across all dimension regimes can be achieved by FTRL through adaptive regularisation, whereby the algorithm initially employs the uniformly convex regulariser suitable for the high-dimensional setting, and later switches to the strongly convex regulariser. While this approach achieves the desired anytime regret optimality, it leaves open a deeper structural question about FTRL: whether such adaptivity is fundamentally necessary for anytime optimality, or whether a fixed regulariser could suffice. The latter would be desirable since it would provide algorithmic simplicity as well as an understanding of how to appropriately regularise ℓ_p-balls across all dimension-regimes simultaneously. Explicitly, we aim to answer the following question:

Can FTRL with a fixed regulariser be anytime optimal for OCO on ℓ_p-balls with $p > 2$?

The main objective of this chapter is to address this question. We begin by examining the regularisers that are optimal in one of the two regimes, and show that each fails to achieve anytime optimality outside its intended regime. We then establish our main result: for the broad class of sign-invariant regularisers, no fixed regulariser can achieve anytime optimal regret with FTRL. This demonstrates that adaptive regularisation is a necessary ingredient for optimality in this

setting.

Our results reveal a previously unexplored limitation of fixed regularisation in online learning and highlight a fundamental obstacle to universality for mirror descent-type algorithms (Sridharan, 2012) with fixed regularisation on non-Euclidean geometries. This impossibility result represents a paradigm shift in our understanding of ℓ_p geometries; it establishes that for $p > 2$, the learner faces a fundamental trade-off that static algorithms are powerless to resolve. In doing so, we elevate adaptive regularisation and doubling tricks from a convenient design choice to a proven requirement for optimality, laying the groundwork for a more nuanced theory of anytime learning that accounts for the inescapable interplay between geometry and the time-horizon. In particular, *our results establish a fundamental separation between low and high dimensional regimes, both in the optimal regret and in the gradient-based methods that achieve it.*

Remark 4.0.1. *We focus on ℓ_p-balls for $p > 2$ because the case $p \in [1, 2]$ is already understood (Levy and Duchi, 2019). Levy and Duchi (2019) show that when $p \in [1, 2]$, FTRL with regulariser $\psi(x) = \|x\|_a^2/2(a-1)$ and $a = \max\{1 + 1/\log(2d), p\}$ achieves a regret of $O(\sqrt{T/(a-1)})$ and this is optimal for all d except if $T < 1/(a-1)$ for which sub-linear regret is not possible.*

Finally, we consider the linear bandit problem where only the loss evaluated at x_t is observed (Section 4.7). We would like to approach this problem in the same way as the full-information problem by characterising the algorithms achieving optimal regret guarantees across all dimension regimes. However, while sub-linear regret bounds have been established in the low-dimensional setting by Kerdreux et al. (2021c), we show that the bandit setting is fundamentally different and much more difficult since sub-linear regret bounds become unachievable when the dimension is large enough (Theorem 4.7.1).

Table 4.1: Optimal regret rates for OCO on ℓ_p-balls ($p > 2$).

	$d \leqslant T$ (low-dim)	$d > T$ (high-dim)
Optimal Regret Rate	$L\sqrt{Td^{1-2/p}}$	$LT^{1-1/p}$
Regularisation (FTRL)	Strongly-convex	Uniformly-convex

Contributions We highlight our contributions for OCO on ℓ_p-balls for $p > 2$ below.

- We consider anytime bounds where the time horizon T is not known in advance and show *FTRL with adaptive regularisation achieves anytime optimality* (Theorem 4.3.1).

Setting	Performance Criterion	Main Theorem	Publication Source
Tabular RL			
Tabular RL			

Table 4.2: Chapter overview.

- We establish algorithmic-specific lower bounds for instances of FTRL that show that the fixed regulariser achieving optimality in low-dimensions ($\|x\|_2^2$) is provably sub-optimal in high-dimensions (Corollary 4.4.2), and the fixed regulariser achieving optimality in high-dimensions ($\|x\|_p^p$) is provably sub-optimal in low-dimensions (Corollary 4.4.3).

- **Our main result:** For **sign-invariant regularisers**, we show that **adaptive regularisation for OMD or FTRL is necessary to achieve anytime optimality** (Theorem 4.4.4), ruling out the existence of a single anytime optimal sign-invariant regulariser.

- In Section 4.2, we connect results from the literature to *fully characterise optimality for ℓ_p-balls with $p > 2$ across all dimension regimes* . In particular, we highlight that FTRL with a strongly-convex regulariser achieves the optimal regret in low-dimensions and FTRL with a uniformly-convex regulariser of degree p achieves the optimal regret in high-dimensions.

- Finally, for bandit feedback where only $\ell_t(x_t)$ is revealed to the learner in each round t instead of the full loss, we establish lower bounds for all convex ℓ_p-balls ($p \geqslant 1$) showing *any linear bandit learner suffers linear regret when the dimension is large enough* (Theorem 4.7.1).

Notation. In this chapter, along with the notation described in Section 1.3, we also use $r_\star$ for the dual of $r \geqslant 1$ satisfying $1/r + 1/r_\star = 1$, $\phi_r(x) = \frac{1}{r}\|x\|_r^r$ for $r \geqslant 2$ and $\partial f(x)$ for the set of gradients of a function f at x.

Roadmap. In Section 4.1, we present preliminaries on FTRL and uniform convexity. In Section 4.2, we characterise the optimal regret of OCO on ℓ_p-balls for $p > 2$ across low and high dimensional regimes. In Section 4.3, we show how to obtain anytime optimal regret with FTRL without advance knowledge of the time horizon T. In Section 4.4, we present our main results on the failure of FTRL with fixed regularisation to achieve anytime optimality. In Section 4.5, we give the proof of the main result Theorem 4.4.4. In Section 4.6, we discuss how our results extend to online mirror descent. In Section 4.7, we consider the bandit feedback setting and show that sub-linear regret is not possible in high dimensions. In Section 4.8, we discuss related

Setting	Performance Criteria	Main Theorem(s)	Publication Source
Online Convex Optimisation	Regret Minimisation	Theorem 4.4.4	Johnson et al. (2025a)

Table 4.3: Chapter overview.

works. In Section 4.9, we conclude and discuss future-work. Finally, in Section 4.10, we provide missing details and proofs omitted from the bandit setting in Section 4.7.

Bibliographic remark: This chapter is based on Johnson et al. (2025a) with the main result being extended to more general regulariser. It is my own work and is licensed under a Creative Commons Attribution 4.0 International License.

4.1 Preliminaries

In this section, we present the Follow-The-Regularised-Leader (FTRL, Section 2.4) family of algorithms and review the regret bounds it can achieve on general OCO problems with uniformly convex regularisers. We focus here on FTRL because of its advantages over Online Mirror Descent (OMD, Section 2.3). When the iterates remain in the interior of the action space, constant step-size OMD and FTRL with fixed regulariser coincide. Beyond this, FTRL is more robust than OMD because its regret bound does not depend on the maximum Bregman divergence between the evolving iterates and the comparator, which it does for OMD. This dependence can be a genuine weakness: there exist settings in which OMD suffers linear regret, while FTRL continues to achieve sublinear regret (Orabona and Pál, 2018). However, the results we discuss also hold for OMD (see Section 4.6).

4.1.1 Follow-the-Regularised-Leader

Let $V \subset \mathbb{R}^d$ be convex and consider proper differentiable[1] convex losses $(\ell_t)_{t=1}^T$. For $t \geqslant 1$, let $\psi_t : \mathbb{R}^d \to \mathbb{R}$ be a proper, closed and differentiable strictly-convex function on V. At time-step $t = 1, ..., T$, FTRL on linearised losses with time-varying regularisers $(\psi_t)_{t=1}^T$ outputs the following points with $g_t \in \partial \ell_t(x_t)$ for all t,

$$x_t = \arg\min_{x \in V} \Big\{ \psi_t(x) + \sum_{s=1}^{t-1} \langle g_s, x \rangle \Big\}. \tag{4.1}$$

[1] We work with differentiable losses for simplicity but all the results we discuss hold with sub-gradients instead of gradients.

Closed-form update of FTRL with a specific uniformly convex regulariser

Fix $r \geqslant 2$. We consider the specific regulariser $\psi(x) = \frac{1}{r}\|x\|_r^r$ and compute the update rule and closed form update for specific losses. The Bregman divergence of ψ is $D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y\rangle$ for all $x, y \in \mathbb{R}^d$. First, we present a general update rule for ψ.

Lemma 4.1.1. *Fix $r \geqslant 2$. Let $\psi(x) = \frac{1}{r}\|x\|_r^r$. Let $V = \mathcal{B}_p$. Let $g_t \in \partial\ell_t(x_t)$. The update rule of FTRL using $\psi_t(x) = \frac{1}{\eta_{t-1}}\psi(x)$ as regularisers is*

$$\tilde{G}_{t+1} = -\eta_t \sum_{s=1}^{t} g_s$$
$$x_{t+1} = \arg\min_{x\in\mathcal{B}_p} D_\psi\Big(x, sign(\tilde{G}_{t+1})|\tilde{G}_{t+1}|^{r_\star - 1}\Big),$$

where sign, power and absolute value functions are applied component-wise to vectors.

Proof. Given $g_t \in \partial\ell_t(x_t)$, the update of FTRL with regulariser ψ_t is (see (4.1))

$$x_{t+1} = \arg\min_{x\in V}\Big\{\eta_t\langle\sum_{s=1}^{t} g_s, x\rangle + \psi(x)\Big\}.$$

By Theorem 6.15 in Orabona (2019), this update is equivalent to

$$\tilde{x}_{t+1} = \arg\min_{x\in\mathbb{R}^d}\Big\{\eta_t\langle\sum_{s=1}^{t} g_s, x\rangle + \psi(x)\Big\},$$
$$x_{t+1} = \arg\min_{x\in\mathcal{B}_p} D_\psi\left(x, \tilde{x}_{t+1}\right).$$

Now by Theorem 6.13 of Orabona (2019), the first minimisation (over $\mathbb{R}^d$) is equivalent to

$$\tilde{x}_{t+1} = \nabla\psi^\star\Big(-\eta_t\sum_{s=1}^{t} g_s\Big),$$

where $\psi^\star$ denotes the Fenchel conjugate of ψ, defined for a proper convex function $f : \mathbb{R}^d \to \mathbb{R}\cup\{+\infty\}$ as

$$f^\star(y) \coloneqq \sup_{x\in\mathbb{R}^d}\big\{\langle x, y\rangle - f(x)\big\}.$$

For an arbitrary norm $\|\cdot\|$, the Fenchel conjugate of $f(x) = \frac{1}{r}\|x\|^r$ is $f^\star(x) = \frac{1}{r_\star}\|x\|_\star^{r_\star}$ (see Lemma 2.2 in Kerdreux et al. (2021b)). Therefore the Fenchel conjugate of $\psi(x)$ is $\psi^\star(x) = \frac{1}{r_\star}\|x\|_{r_\star}^{r_\star}$ and $\nabla\psi^\star(x) = \text{sign}(x)|x|^{r_\star-1}$. Combining everything gives the result. □

We now provide two lemmas pertaining to the Bregman projections of the FTRL update in the

above lemma for specific cases that will be of use in the proofs in Section 4.4. These will allow us to exactly compute points played by FTRL with $\psi(x) = \frac{1}{r}\|x\|_r^r$ with certain losses, and in turn compute their regret.

Bregman projection for constant vector.

Lemma 4.1.2. *Consider $z = c \cdot w$ where w is a vector with all entries equal to 1 and $c > d^{-1/p}$ so that $z \notin \mathcal{B}_p$. The Bregman projection $\operatorname{argmin}_{x \in \mathcal{B}_p} D_\psi(x, z)$ with $\psi(x) = \frac{1}{r}\|x\|_r^r$ of z is $d^{-1/p} \cdot w$, the rescaled version of w that has ℓ_p-norm equal to 1.*

Proof. We make use of Lemma 5.4 in Bubeck (2011): if f is a convex and differentiable function on $\mathcal{B}_p$ then x is a minimiser of $f(x)$ in $\mathcal{B}_p$ if and only if $\nabla f(x)^T(y-x) \geqslant 0$ for all $y \in \mathcal{B}_p$. Consider

$$\begin{aligned} f(x) = D_\psi(x, z) &= \psi(x) - \psi(z) - \nabla\psi(z)^T(x - z), \\ \nabla f(x) &= \nabla\psi(x) - \nabla\psi(z), \\ [\nabla\psi(x)]_i &= \operatorname{sign}(x_i)|x_i|^{r-1} \end{aligned}$$

Consider $x = d^{-1/p} \cdot w$. From the lemma mentioned above, it is enough to show that $\nabla f(x)^T(y - x) \geqslant 0$ for all $y \in \mathcal{B}_p$:

$$\begin{aligned} \nabla f(x)^T(y - x) &= (\nabla\psi(x) - \nabla\psi(z))^T(y - x) \\ &= (d^{-(r-1)/p} \cdot w - c^{r-1} \cdot w)^T(y - x) \\ &= (c^{r-1} - d^{-(r-1)/p})w^T(x - y) \\ &= (c^{r-1} - d^{-(r-1)/p})(d^{1-1/p} - \sum_{i=1}^{d} y_i) \\ &\geqslant (c^{r-1} - d^{-(r-1)/p})(d^{1-1/p} - \|y\|_1) \\ &\geqslant (c^{r-1} - d^{-(r-1)/p})(d^{1-1/p} - d^{1-1/p}\|y\|_p) \\ &\geqslant 0, \end{aligned}$$

where we used that $c^{r-1} - d^{-(r-1)/r} > 0$ and $\|y\|_1 \leqslant d^{1-1/p}\|y\|_p \leqslant d^{1-1/p}$ for all $y \in \mathcal{B}_p$. □

Bregman projection for basis vector.

Lemma 4.1.3. *Consider $z = c \cdot e_1$ where e_1 is the first canonical basis vector and $|c| > 1$ so that $z \notin \mathcal{B}_p$. The Bregman projection $\operatorname{argmin}_{x \in \mathcal{B}_p} D_\psi(x, z)$ with $\psi(x) = \frac{1}{r}\|x\|_r^r$ of z is $\operatorname{sign}(c) \cdot e_1$.*

Proof. As in the proof of Lemma 4.1.2, it is enough to show that $\nabla f(x)^T(y-x) \geqslant 0$ for all $y \in \mathcal{B}_p$, with $x = \text{sign}(c) \cdot e_1$, and

$$\begin{aligned}
f(x) &= D_\psi(x,z) = \psi(x) - \psi(z) - \nabla\psi(z)^T(x-z), \\
\nabla f(x) &= \nabla\psi(x) - \nabla\psi(z), \\
[\nabla\psi(x)]_i &= \text{sign}(x_i)|x_i|^{r-1}. \\
\nabla f(x)^T(y-x) &= (\nabla\psi(x) - \nabla\psi(z))^T(y-x) \\
&= (\nabla\psi(\text{sign}(c)\cdot e_1) - \nabla\psi(c\cdot e_1))^T(y-x) \\
&= \text{sign}(c)\cdot(|c|^{r-1}-1)e_1^T(x-y) \\
&= \text{sign}(c)\cdot(|c|^{r-1}-1)(\text{sign}(c)-y_1) \\
&\geqslant 0,
\end{aligned}$$

where we used that $|c| > 1$ and $y_1 \leqslant 1$ for all $y \in \mathcal{B}_p$. □

4.1.2 Universal optimality

We present an extension of the result of Sridharan (2012) on the universality of OMD and FTRL. We first review their result and refer the reader to Part II / Chapters 5 & 7 of Sridharan (2012) for the complete details.

We consider general OCO where the action set $V \subset \mathcal{B}$ is a convex and centrally symmetric set that is a subset of an arbitrary real vector space $\mathcal{B}$. The gradients of the (convex) losses belong to a set $\mathcal{X} \subset \mathcal{B}^\star$ that is a subset of the dual $\mathcal{B}^\star$ of $\mathcal{B}$ - this is assumed below throughout. We will denote the regret suffered by a learning algorithm A on the sequence of losses ℓ_t by $R_T(A)$. The minimax regret is

$$\mathcal{V}_T(V,\mathcal{X}) = \inf_A \sup_{\ell_1,\ldots,\ell_T} R_T(A).$$

The following result from Sridharan (2012) shows that FTRL is a universally optimal OCO learning algorithm, in the sense that is optimal for any OCO problem. The result was originally stated for OMD but also holds for FTRL.

Theorem 4.1.4 (Theorem 71 of Sridharan (2012))**.** *If for some constant $C > 0$ and some $q \in [2,\infty)$, $\mathcal{V}_T(V,\mathcal{X}) \leqslant CT^{1-1/q}$ for all T, then for any $T > e^{q-1}$, there exists a regulariser*

function ψ and step-size η, such that the regret of FTRL $\mathcal{A}_{FTRL}$ with regularisers $\psi_t(x) = \frac{1}{\eta}\psi(x)$ against any $\ell_1, ..., \ell_T$ chosen by the adversary is bounded as

$$R_T(\mathcal{A}_{FTRL}) \leqslant 6002 \cdot C \cdot \log^2(T) \cdot T^{1-1/q}.$$

The result states that any regret bound with constant rate that is achievable across all time horizons can be matched by FTRL up to logarithmic factors. The result was recently improved by (Gatmiry et al., 2025) to omit any log-factors in T for the case of online linear optimisation and $q = 2$.

We now provide an extension that handles the case where there may be multiple regret bounds with different constant rates that exchange ordering at different time horizon intervals.

Theorem 4.1.5. *Let $K > 0$ be an integer. If for $k = 1, ..., K$, there exists constants $C_k > 0$ (w.r.t. T) and $q_k \geqslant 2$ such that $\mathcal{V}_T(V, \mathcal{X}) \leqslant \min_{k=1,...,K}\left\{C_k T^{1-1/q_k}\right\}$ for all T, then for any $T > \max_{k=1,...,K} e^{q_k - 1}$, there exists a procedure $\mathcal{A}_{FTRL+}$ running FTRL with different regularisers over intervals of doubling lengths such that the corresponding regret against any $\ell_1, ..., \ell_T$ chosen by the adversary is bounded as*

$$R_T(\mathcal{A}_{FTRL+}) \leqslant (2 + \sqrt{2}) \cdot 6002 \cdot \log^2(T) \cdot \min_k\left\{C_k T^{1-1/q_k}\right\}.$$

The procedure $\mathcal{A}_{FTRL+}$ does not require knowledge of T.

The result highlights the power of FTRL: even with non-constant rate with respect to T, any regret bound can be matched by an FTRL-inspired procedure up to logarithmic factors. This matches up to a factor of $2 + \sqrt{2}$ the regret bound we would get by using Theorem 4.1.4 / Theorem 71 of Sridharan (2012) with advance knowledge of T, and otherwise matches the minimax regret up to constant and logarithmic factors.

We now provide the proof and procedure based on the doubling-trick.

Proof of Theorem 4.1.5. Fix T, $T_i = 2^i$, $B = \min\left\{j \in \mathbb{N} : \sum_{i=0}^{j} T_i \geqslant T\right\}$. We have:

$$2^B - 1 = \sum_{i=0}^{B-1} T_i < T \leqslant \sum_{i=0}^{B} T_i = \sum_{i=0}^{B} 2^i = 2^{B+1} - 1 \implies T \in [2^B, 2^{B+1} - 1].$$

Consider the following procedure $\mathcal{A}_{FTRL+}$. For $i = 0, 1, ..., B$:

- set $k(i) = \arg\min_k \left\{ C_k T_i^{1-1/q_k} \right\}$.
- Use Theorem 4.1.4 / Theorem 71 of Sridharan (2012) to get a regulariser for FTRL that achieves the regret upper-bound of $6002 \cdot C_{k(i)} \log^2(T_i) \cdot T_i^{1-1/q_{k(i)}}$ over T_i rounds. Use this on the T_i rounds $\left\{ \sum_{j=0}^{i-1} T_j + 1, ..., \sum_{j=0}^{i-1} T_j + T_i \right\}$.

Let $k_\star = \arg\min_k \left\{ C_k T^{1-1/q_k} \right\}$. The regret is bounded as follows:

$$
\begin{aligned}
R_T(\mathcal{A}_{FTRL+}, g_1, ..., g_T) &\leqslant \sum_{i=0}^{B} 6002 C_{k(i)} \log^2(T_i) \cdot T_i^{1-1/q_{k(i)}} \\
&\leqslant 6002 \log^2(T) \sum_{i=0}^{B} C_{k(i)} \cdot T_i^{1-1/q_{k(i)}} \qquad \text{since for all i, } T_i \leqslant T \\
&\leqslant 6002 \log^2(T) \sum_{i=0}^{B} C_{k_\star} \cdot T_i^{1-1/q_{k_\star}} \qquad \text{by definition of } k(i) \\
&= 6002 \cdot C_{k_\star} \cdot \log^2(T) \cdot \sum_{i=0}^{B} (2^{1-1/q_{k_\star}})^i \\
&= 6002 \cdot C_{k_\star} \cdot \log^2(T) \cdot \frac{(2^{1-1/q_{k_\star}})^{B+1} - 1}{2^{1-1/q_{k_\star}} - 1} \\
&\leqslant \frac{2^{1-1/q_{k_\star}}}{2^{1-1/q_{k_\star}} - 1} \cdot 6002 \cdot C_{k_\star} \cdot \log^2(T) \cdot (2^B)^{1-1/q_{k_\star}} \\
&\leqslant \frac{2^{1-1/q_{k_\star}}}{2^{1-1/q_{k_\star}} - 1} \cdot 6002 \cdot C_{k_\star} \cdot \log^2(T) \cdot T^{1-1/q_{k_\star}} \qquad \text{since } 2^B \leqslant T \\
&\leqslant \frac{\sqrt{2}}{\sqrt{2} - 1} \cdot 6002 \cdot C_{k_\star} \cdot \log^2(T) \cdot T^{1-1/q_{k_\star}} \qquad \text{since } q_{k_\star} \geqslant 2 \\
&= (2 + \sqrt{2}) \cdot 6002 \cdot \log^2(T) \cdot \min_k \left\{ C_k T^{1-1/q_k} \right\},
\end{aligned}
$$

which concludes the proof. □

4.1.3 Uniform convexity

We work with the definition of a uniformly-convex function from (Nesterov, 2018) (note there are also other standard equivalent definitions, see e.g. (Kerdreux et al., 2021b)).

Definition 4.1.6 (Nesterov (2018))**.** *A differentiable function f on a closed convex set V is μ-uniformly-convex on V of degree $p > 2$ w.r.t. a norm $\|\cdot\|$ if there exists $\mu > 0$ such that for all $x, y \in V$,*

$$
f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{p} \|y - x\|^p.
$$

Uniform convexity generalises strong convexity by weakening the condition of a quadratic lower

bound, allowing functions that are locally much flatter (see Figure 4.1.3 for an illustration). In particular, uniform convexity with $p = 2$ recovers strong convexity.

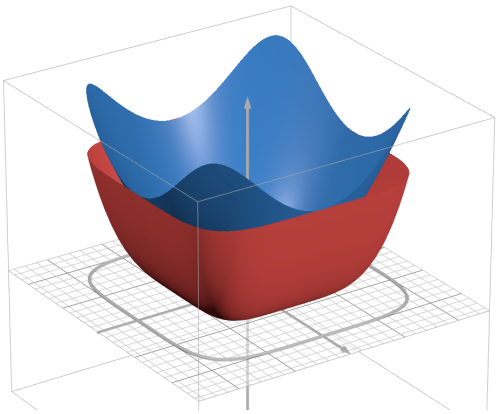

Figure 4.2: Illustration in $\mathbb{R}^2$ of a uniformly convex function (red, w.r.t. $\|\cdot\|_5$) vs a strongly convex function (blue w.r.t. $\|\cdot\|_2$ so also $\|\cdot\|_5$). For the ℓ_p-ball $\mathcal{B}_5$ that gets closer to a hyper-cube (the boundary can be seen as the top outline of the red surface or in gray on the bottom surface), the strongly convex function takes large values near the corners, while the uniformly convex function does not.

A useful property of uniformly convex functions is given in the following lemma, which is an extension of Corollary 7.7 of Orabona (2019). Recall that $r_\star$ is the dual of $r \geqslant 1$ satisfying $1/r + 1/r_\star = 1$.

Lemma 4.1.7. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be closed, proper, sub-differentiable and μ-uniformly convex of degree r w.r.t. a norm $\|\cdot\|$. Let $x^\star = \arg\min_{x \in dom f} f(x)$. Then for all $x \in$ dom f and $g \in \partial f(x)$, we have*

$$f(x) - f(x^\star) \leqslant \frac{r-1}{r\mu^{1/(r-1)}} \|g\|_\star^{r/(r-1)}.$$

Proof. By the uniform convexity of f, we have

$$\begin{aligned}
f(x^\star) &= \min_{z \in \mathrm{dom} f} f(z) \\
&\geqslant \min_{z \in \mathrm{dom} f} \left\{ f(x) + \langle g, z - x \rangle + \frac{\mu}{r} \|z - x\|^r \right\} \\
&\geqslant f(x) + \min_{z \in \mathbb{R}^d} \left\{ \langle g, z - x \rangle + \frac{\mu}{r} \|z - x\|^r \right\} \\
&= f(x) + \min_{z \in \mathbb{R}^d} \left\{ \langle g, z \rangle + \frac{\mu}{r} \|z\|^r \right\} \\
&= f(x) - \mu \max_{z \in \mathbb{R}^d} \left\{ \langle \frac{-g}{\mu}, z \rangle - \frac{1}{r} \|z\|^r \right\} \\
&= f(x) - \frac{\mu}{r_\star} \left\| \frac{-g}{\mu} \right\|_\star^{r_\star}
\end{aligned}$$

$$
\begin{aligned}
&= f(x) - \mu^{1-r_\star} \frac{\|g\|_\star^{r_\star}}{r_\star} \\
&= f(x) - \frac{r-1}{r\mu^{1/(r-1)}} \|g\|_\star^{r/(r-1)}
\end{aligned}
$$

where we used that the fenchel conjugate of $\frac{\|x\|^r}{r}$ is $\frac{\|x\|_\star^{r_\star}}{r_\star}$ Rearranging gives the result. □

Uniformy convexity of $\|x\|_p^p$: An important example of a uniformly convex function is given in the following proposition.

Proposition 4.1.8. *Fix $p > 2$. $\phi_p(x) = \frac{1}{p}\|x\|_p^p$ is 2^{1-p}-uniformly-convex of degree p w.r.t. $\|\cdot\|_p$ on $\mathcal{B}_p$.*

Proof. This is a well-known result derived from Clarkson's inequality. We include a proof to provide clarity on the constant of uniform convexity.

Consider $x, y \in \mathcal{B}_p$. Following the steps in Remark 2.1 of Zălinescu (1983), using convexity of ψ we have for $\lambda \in [0, 1/2]$,

$$
\begin{aligned}
\psi(\lambda x + (1-\lambda)y) &= \psi\Big(2\lambda\Big(\frac{x+y}{2}\Big) + (1-2\lambda)y\Big) \\
&\leqslant 2\lambda\psi\Big(\frac{x+y}{2}\Big) + (1-2\lambda)\psi(y) \\
&= \frac{2\lambda}{p}\Big\|\frac{x+y}{2}\Big\|_p^p + \frac{(1-2\lambda)}{p}\|y\|_p^p.
\end{aligned}
$$

From Clarkson's inequality (equation (2.1) in Ball et al. (1994)), we have that

$$
\Big\|\frac{x+y}{2}\Big\|_p^p + \Big\|\frac{x-y}{2}\Big\|_p^p \leqslant \frac{\|x\|_p^p}{2} + \frac{\|y\|_p^p}{2}.
$$

Using this in the above we have

$$
\begin{aligned}
\psi(\lambda x + (1-\lambda)y) &\leqslant \frac{2\lambda}{p}\frac{\|x\|_p^p}{2} + \frac{2\lambda}{p}\frac{\|y\|_p^p}{2} - \frac{2\lambda}{p}\Big\|\frac{x-y}{2}\Big\|_p^p + \frac{(1-2\lambda)}{p}\|y\|_p^p \\
&= \lambda\frac{\|x\|_p^p}{p} + (1-\lambda)\frac{\|y\|_p^p}{p} - \frac{2\lambda}{p}\Big\|\frac{x-y}{2}\Big\|_p^p \\
&\leqslant \lambda\psi(x) + (1-\lambda)\psi(y) - \frac{2\lambda(1-\lambda)}{p}\Big\|\frac{x-y}{2}\Big\|_p^p. \qquad (4.2)
\end{aligned}
$$

We now show (following steps in Definition 3.2 of Kerdreux et al. (2021b)) that this is equivalent

to uniform convexity. From the convexity and differentiability of ψ,

$$\begin{aligned}\psi(y) + \lambda\langle\nabla\psi(y), x - y\rangle &= \psi(y) + \langle\nabla\psi(y), [y + \lambda(x - y)] - y\rangle \\ &\leqslant \psi(y + \lambda(x - y)) \\ &\leqslant \lambda\psi(x) + (1 - \lambda)\psi(y) - \frac{2\lambda(1 - \lambda)}{p}\left\|\frac{x - y}{2}\right\|_p^p.\end{aligned}$$

Rearrenging,

$$\begin{aligned}&\implies \lambda\langle\nabla\psi(y), x - y\rangle \leqslant \lambda(\psi(x) - \psi(y)) - \frac{2\lambda(1 - \lambda)}{p}\left\|\frac{x - y}{2}\right\|_p^p \\ &\implies \langle\nabla\psi(y), x - y\rangle \leqslant (\psi(x) - \psi(y)) - \frac{2(1 - \lambda)}{p}\left\|\frac{x - y}{2}\right\|_p^p \\ &\implies \psi(x) \geqslant \psi(y) + \langle\nabla\psi(y), x - y\rangle + \frac{2}{p}\left\|\frac{x - y}{2}\right\|_p^p,\end{aligned}$$

as $\lambda \to 0$. So for any $x, y \in \mathcal{B}_p$ we have the condition of uniform convexity with $\mu = 2^{1-p}$. □

4.1.4 FTRL analysis and regret guarantees

Though FTRL is usually considered with a strongly-convex regulariser, its analysis can be generalised to uniformly-convex regularisers (Bhaskara et al., 2020). We follow and extend the analysis of FTRL from Orabona (2019) (Section 7). Existence and uniqueness of the update can be handled along the same lines as Theorem 6.8 in Orabona (2019) with uniform convexity.

For $t \geqslant 1$, let $\psi_t : \mathbb{R}^d \to \mathbb{R}$ be μ_t-uniformly-convex on V of degree $r_t \geqslant 2$ with respect to a norm $\|\cdot\|_{|t}$ (we use this notation to avoid confusion with the ℓ_p norm $\|\cdot\|_p$, we will denote the dual norm as $\|\cdot\|_{|t\star}$).

The analysis begins with the following expression for the regret. We refer the reader to Orabona (2019) for the proof.

Lemma 4.1.9. *(Lemma 7.1 of Orabona (2019)) Denote $F_t(x) = \psi_t(x) + \sum_{s=1}^{t-1}\ell_s(x)$ and set $x_t \in \arg\min_{x\in V} F_t(x)$. Let[2] $\psi_{T+1} := \psi_T$. Then, for any $u \in V$ we have*

$$\sum_{t=1}^{T}\ell_t(x_t) - \ell_t(u) \leqslant \psi_T(u) - \min_{x\in V}\psi_1(x) + \sum_{t=1}^{T}\Big\{F_t(x_t) - F_{t+1}(x_{t+1}) + \ell_t(x_t)\Big\}. \tag{4.3}$$

Since ψ_t is proper, closed, differentiable and μ_t-uniformly convex of degree r_t with respect to $\|\cdot\|_{|t}$

[2] ψ_{T+1} is used in the analysis but not in the algorithm over T rounds

and the losses are proper and convex, $F_t(x) + \ell_t(x) = \psi_t(x) + \sum_{s=1}^t \ell_s(x)$ is also proper, closed, sub-differentiable and μ_t-uniformly convex of degree r_t with respect to $\|\cdot\|_{|t}$. Applying Lemma 4.1.7 to $F_t + \ell_t$, we have with $x_t^\star = \arg\min_{x\in V} F_t(x) + \ell_t(x)$

$$\begin{aligned}
F_t(x_t) - F_{t+1}(x_{t+1}) + \ell_t(x_t) &= \Big(F_t(x_t) + \ell_t(x_t)\Big) - \Big(F_t(x_{t+1}) + \ell_t(x_{t+1})\Big) + \psi_t(x_{t+1}) - \psi_{t+1}(x_{t+1}) \\
&\leqslant \Big(F_t(x_t) + \ell_t(x_t)\Big) - \Big(F_t(x_t^\star) + \ell_t(x_t^\star)\Big) + \psi_t(x_{t+1}) - \psi_{t+1}(x_{t+1}) \\
&\leqslant \frac{r_t - 1}{r_t \mu_t^{1/(r_t-1)}} \|g_t\|_{|t\star}^{r_t/(r_t-1)} + \psi_t(x_{t+1}) - \psi_{t+1}(x_{t+1}),
\end{aligned}$$

where we used that $g_t \in \partial(F_t + \ell_t)(x_t)$ since $g_t \in \partial\ell_t(x_t)$ and $x_t = \arg\min_{x\in V} F_t(x)$. Plugging the above into (4.3) gives the following result:

Theorem 4.1.10. *Let $V \subset \mathbb{R}^d$ be convex and consider proper differentiable convex losses $(\ell_t)_{t=1}^T$. For $t \geqslant 1$, let $\psi_t : \mathbb{R}^d \to \mathbb{R}$ be a proper, closed and differentiable μ_t-uniformly-convex function on V of degree $r_t \geqslant 2$ with respect to a norm $\|\cdot\|_{|t}$ (whose dual norm is denoted $\|\cdot\|_{|t\star}$). Then for any $u \in V$ and $g_t \in \partial\ell_t(x_t)$, the points played by FTRL with time-varying regularisers $(\psi_t)_{t=1}^T$ satisfy the following regret bound:*

$$\sum_{t=1}^T \ell_t(x_t) - \ell_t(u) \leqslant \psi_T(u) - \min_{x\in V} \psi_1(x) + \sum_{t=1}^T \left\{ \frac{(r_t - 1)}{r_t \mu_t^{\frac{1}{r_t-1}}} \|g_t\|_{|t\star}^{\frac{r_t}{r_t-1}} + \psi_t(x_{t+1}) - \psi_{t+1}(x_{t+1}) \right\}.$$

It is common to consider FTRL with a fixed regulariser with a step-size ($\psi_t(x) = \frac{1}{\eta_{t-1}}\psi(x)$ for a fixed ψ) so that the condition of uniform convexity is fixed for all rounds, for which we get the following result.

Corollary 4.1.11. *Let $V \subset \mathbb{R}^d$ be convex and consider proper differentiable convex losses $(\ell_t)_{t=1}^T$. Let $\psi : \mathbb{R}^d \to \mathbb{R}_{\geqslant 0}$ be a proper, closed and differentiable μ-uniformly-convex function on V of degree $r \geqslant 2$ with respect to a norm $\|\cdot\|$. Assume V is bounded and let D be such that $\sup_{x\in V} \psi(x) \leqslant D$. Assume the losses are $L_{\|\cdot\|}$-Lipschitz with respect to $\|\cdot\|$. Consider using FTRL in (4.1) with regularisers $\psi_t(x) = \frac{1}{\eta_{t-1}}\psi(x)$ and $\eta_{t-1} = \frac{D^{1/r_\star}\mu^{1/r}}{L_{\|\cdot\|}(r_\star-1)^{1/r_\star} t^{1/r_\star}}$. Then for any $u \in V$,*

$$\sum_{t=1}^T \ell_t(x_t) - \ell_t(u) \leqslant \frac{r^{1/r} r_\star^{1/r_\star}}{\mu^{1/r}} L_{\|\cdot\|} D^{1/r} T^{1/r_\star}.$$

Proof. Since ψ is μ-uniformly convex function on V of degree r with respect to $\|\cdot\|$, then the regulariser used by FTRL in round t, $\psi_t = \frac{1}{\eta_{t-1}}\psi$ is $\frac{\mu}{\eta_{t-1}}$-uniformly convex function on V of degree r with respect to $\|\cdot\|$. Since $\eta_t \leqslant \eta_{t-1}$, $\psi_t(x_{t+1}) - \psi_{t+1}(x_{t+1}) = \left(\frac{1}{\eta_{t-1}} - \frac{1}{\eta_t}\right)\psi(x_{t+1}) \leqslant 0$.

By the Lipschitz condition on the losses, we have $\|g_t\|_\star \leqslant L_{\|\cdot\|}$. Applying Theorem 4.1.10 and plugging in value of η_{t-1}, we have

$$\begin{aligned}
\sum_{t=1}^{T} \ell_t(x_t) - \ell_t(u) &\leqslant \frac{\psi(u)}{\eta_{T-1}} + \frac{L_{\|\cdot\|}^{r_\star}}{r_\star \mu^{r_\star - 1}} \sum_{t=1}^{T} \eta_{t-1}^{r_\star - 1} \\
&\leqslant \frac{L_{\|\cdot\|} D^{1-1/r_\star} (r_\star - 1)^{1/r_\star} T^{1/r_\star}}{\mu^{1/r}} + \frac{L_{\|\cdot\|}^{r_\star}}{r_\star \mu^{r_\star - 1}} \sum_{t=1}^{T} \Big(\frac{D^{1/r_\star} \mu^{1/r}}{L_{\|\cdot\|} (r_\star - 1)^{1/r_\star} t^{1/r_\star}} \Big)^{r_\star - 1} \\
&\leqslant \frac{L_{\|\cdot\|} D^{1/r} (r_\star - 1)^{1/r_\star} T^{1/r_\star}}{\mu^{1/r}} + \frac{L_{\|\cdot\|} D^{1/r}}{r_\star (r_\star - 1)^{1/r} \mu^{(r_\star - 1)(1-1/r)}} \sum_{t=1}^{T} \frac{1}{t^{1/r}} \\
&= \frac{L_{\|\cdot\|} D^{1/r}}{\mu^{1/r}} \Big((r_\star - 1)^{1/r_\star} T^{1/r_\star} + \frac{1}{r_\star (r_\star - 1)^{1/r}} \sum_{t=1}^{T} \frac{1}{t^{1/r}} \Big).
\end{aligned}$$

Now note that

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{t^{1/r}} &\leqslant \int_0^T \frac{1}{x^{1/r}} dx = \Big[\frac{1}{1 - 1/r} x^{1 - 1/r} \Big]_0^T = r_\star T^{1/r_\star} \\
\implies \sum_{t=1}^{T} \ell_t(x_t) - \ell_t(u) &\leqslant \frac{L_{\|\cdot\|} D^{1/r} T^{1/r_\star}}{\mu^{1/r}} \Big((r_\star - 1)^{1/r_\star} + \frac{1}{(r_\star - 1)^{1/r}} \Big).
\end{aligned}$$

The proof is concluded by noting that $(r_\star - 1)^{1/r_\star} + \frac{1}{(r_\star - 1)^{1/r}} = r^{1/r} r_\star^{1/r_\star}$. □

The degree r of uniform convexity in the above regret bound offers a trade-off between the dependence on the horizon T and the diameter D, while leaving the Lipschitz constant unaffected. In particular, a larger r will shrink the dependence on the diameter at the cost of a worst rate w.r.t. T. This can give better regret bounds for high-dimensional problems where the dimension dependence arises through the diameter D, as is the case for ℓ_p-balls (see Section 4.2.2). With $r = 2$, we obtain the regret bound of FTRL using a strongly-convex regulariser, $R_T \leqslant 2 L_{\|\cdot\|} \sqrt{DT/\mu}$.

It is also possible to obtain bounds that are Lipschitz-adaptive (do not require knowledge of $L_{\|\cdot\|}$) and adapt to the sequence of gradients (scale with $\sum_{t=1}^{T} \|g_t\|_\star^{r_\star}$ instead of $T^{1/r_\star}$) using a gradient clipping technique (the techniques described in the blog-post by Tim van Erven (2021) and Section 4 in Cutkosky (2019) apply immediately to our setting).

4.2 Optimal OCO on ℓ_p-balls

In this section, we consider OCO on ℓ_p-balls and characterise the optimal regret across different dimension regimes. The action set is $V = \mathcal{B}_p = \{x \in \mathbb{R}^d : \|x\|_p \leqslant 1\}$, the unit ℓ_p-ball with $p > 2$.

We assume we have L-Lipschitz losses in ℓ_p norm (i.e. $\|g_t\|_{p_\star} \leqslant L$ for $g_t \in \partial \ell_t(x_t)$), ensuring the regret incurred in a single round is bounded by $2L$, and the overall regret by $2LT$.

The optimal regret exhibits a separation between the *low-dimensional* regime where the dimension d of the ℓ_p-ball is smaller than the time horizon T, and the *high-dimensional* regime where it is larger. We consider these two regimes separately in this section. For each, we apply the FTRL regret bounds from the previous section with specific regularisers to obtain regret bounds, which we show are optimal.

4.2.1 Low-dimensional regime

In this section, we consider OCO on $\mathcal{B}_p$ in the low-dimensional setting where $d \leqslant T$. Using FTRL with strongly-convex regularisation in this setting achieves the optimal regret. Specifically, we consider the squared ℓ_2-norm $\phi_2(x) = \frac{1}{2}\|x\|_2^2$ as the regulariser. This is 1-strongly-convex on V with respect to $\|\cdot\|_2$ and we can apply Corollary 4.1.11 with $r = 2$. We have that $L_{\|\cdot\|_2} \leqslant L$ since the losses are L-Lipschitz with respect to $\|\cdot\|_p$ and $p_\star \leqslant 2$ so $\|g_t\|_2 \leqslant \|g_t\|_{p_\star} \leqslant L$. The diameter of $\mathcal{B}_p$ measured by ϕ_2 is $D = \sup_{x \in \mathcal{B}_p} \phi_2(x) = \sup_{x \in \mathcal{B}_p} \frac{1}{2}\|x\|_2^2 = \frac{1}{2} d^{1-2/p}$. So FTRL guarantees $R_T \leqslant L\sqrt{2d^{1-2/p}T}$, which is optimal up to constants for $d \leqslant T$ as shown by the theorem below.

Theorem 4.2.1. *Fix $d \leqslant T$ and let $\mathcal{A}$ be any algorithm for OCO on $V = \mathcal{B}_p$. There exists a sequence of L-Lipschitz losses w.r.t. $\|\cdot\|_p$ such that $\mathcal{A}$ suffers a regret of at least $\frac{1}{\sqrt{6}} L\sqrt{d^{1-2/p}T}$.*

Optimality in low-dimension was established by Levy and Duchi (2019) (both upper and lower bounds). However, the lower bound we present above contains better constants and a simpler analysis stemming from the "probabilistic" method instead of the reductions from estimation to testing used by Levy and Duchi (2019).

Proof. We have $d \leqslant T$. Let $k = \lfloor T/d \rfloor \geqslant 1$. Let $Y_{i,j}$ be i.i.d. Rademacher random variables for $1 \leqslant i \leqslant d$, $1 \leqslant j \leqslant k$, i.e. $\mathbb{P}(Y_{i,j} = 1) = \mathbb{P}(Y_{i,j} = -1) = 1/2$. Let $e_1, ..., e_d$ be the canonical basis of $\mathbb{R}^d$. Define $g_t = LY_{i,j} \cdot e_i$ where $t = k(i-1) + j$ (for k rounds we stick to the same coordinate and draw i.i.d. Rademacher random variables). Denote the point played by $\mathcal{A}$ by x_t and fix the loss to be $\tilde{\ell}_t(x) = g_t^T x$ (for $t > dk$, fix $\tilde{\ell}_t(x) = 0$). The gradient is g_t, which is bounded by L in $\ell_{p_\star}$-norm. The point x_t depends on the losses up to time $t-1$ but not on $\tilde{\ell}_t$ and is independent of Y_t, so for all t:

$$\mathbb{E}[\tilde{\ell}_t(x_t)] = \mathbb{E}[Y_t L \cdot e_t^T x_t] = \mathbb{E}[Y_t] L e_t^T \mathbb{E}[x_t] = 0 \implies \mathbb{E}[\sum_{t=1}^{T} \tilde{\ell}_t(x_t)] = 0.$$

On the other hand, $u = -d^{-1/p}\sum_{i=1}^{d}\operatorname{sign}\Big\{\sum_{j=1}^{k} Y_{i,j}\Big\}e_i \in \mathcal{B}_p$ gives

$$\begin{aligned}\min_{x\in\mathcal{B}_p}\sum_{t=1}^{T}\tilde{\ell}_t(x) &\leqslant \sum_{t=1}^{T}\tilde{\ell}_t(u)\\ &= -Ld^{-1/p}\Big(\sum_{i=1}^{d}\operatorname{sign}\Big\{\sum_{j=1}^{k} Y_{i,j}\Big\}e_i\Big)^T\Big(\sum_{i=1}^{d}\sum_{j=1}^{k} Y_{i,j}e_i\Big)\\ &= -Ld^{-1/p}\sum_{i=1}^{d}\Big|\sum_{j=1}^{k} Y_{i,j}\Big|.\end{aligned}$$

We now make use of a result from Esfandiari et al. (2021) (proof of Lemma 7.2): fix $B > 0$, consider $X = \sum_{i=1}^{B} t_i R_i$ where t_i are positive integers such that $\sum_{i=1}^{B} t_i = k$ and R_i are i.i.d Rademacher random variables. Then $\mathbb{E}\big[|X|\big] \geqslant k/\sqrt{3B}$.
In our case, with $B = k$ and $t_i = 1$ for all i, we have that $\mathbb{E}[|\sum_{j=1}^{k} Y_{i,j}|] \geqslant \sqrt{k/3} \geqslant \sqrt{T/6d}$ (since $k = \lfloor T/d\rfloor \geqslant T/2d$ for $T \geqslant d$) which gives

$$\mathbb{E}\Big[\sum_{t=1}^{T}\tilde{\ell}_t(x_t) - \min_{x\in\mathcal{B}_p}\sum_{t=1}^{T}\tilde{\ell}_t(x)\Big] \geqslant 0 + Ld^{-1/p}\sum_{i=1}^{d}\sqrt{\frac{T}{6d}} = Ld^{-1/p}\sqrt{\frac{Td}{6}} = L\sqrt{\frac{Td^{1-2/p}}{6}}$$

The result follows by: $\sup_{\ell_1,\dots,\ell_T} R_T \geqslant \mathbb{E}\Big[\sum_{t=1}^{T}\tilde{\ell}_t(x_t) - \min_{x\in\mathcal{B}_p}\sum_{t=1}^{T}\tilde{\ell}_t(x)\Big] \geqslant L\sqrt{\frac{Td^{1-2/p}}{6}}$. □

4.2.2 High-dimensional regime

In this section, we consider OCO on $\mathcal{B}_p$ in the high-dimensional setting where $d > T$. We saw in the previous section that the optimal regret in low-dimensions of $O(\sqrt{d^{1-2/p}T})$ is polynomial in the dimension. As $d \to \infty$ for fixed T, this polynomial dependence on the dimension cannot remain optimal since $R_T \leqslant 2LT$ is bounded. Nevertheless, we will see that sub-linear regret bounds (in T) are possible for any $d > T$, even when d is such that $\sqrt{d^{1-2/p}T} \geqslant T$. However, this is not achievable using strongly-convex regularisation (we delay discussion of this failure to Section 4.4). Instead, in this section we consider uniformly-convex regularisation of degree $p > 2$ that enforces less curvature, allowing points in the corners of $\mathcal{B}_p$ to be more appropriately regularised in high dimensions. This allows us to obtain optimal regret bounds for the high-dimensional regime.

We consider FTRL on $\mathcal{B}_p$ with the regulariser $\phi_p(x) = \frac{1}{p}\|x\|_p^p$. Proposition 4.1.8 ensures that ϕ_p is uniformly-convex of degree p with respect to $\|\cdot\|_p$, allowing us to apply Corollary 4.1.11 with $r = p$. We have that $L_{\|\cdot\|_p} = L$ since the losses are L-Lipschitz with respect to $\|\cdot\|_p$. The diameter of $\mathcal{B}_p$ measured by ϕ_p is $D = 1/p$. So FTRL guarantees $R_T \leqslant L\big(2p_\star T\big)^{1/p_\star}$, which is optimal up

to constants for $d > T$ as shown by the theorem below.

Theorem 4.2.2. *Fix $d > T$ and let $\mathcal{A}$ be any algorithm for OCO on $V = \mathcal{B}_p$. There exists a sequence of L-Lipschitz losses w.r.t. $\|\cdot\|_p$ such that $\mathcal{A}$ suffers a regret of at least $LT^{1/p_\star}$.*

This lower bound follows from an online-to-batch conversion of the lower bound for high-dimensional (offline) Lipschitz convex optimisation (Nemirovskij and Yudin, 1983; Guzmán and Nemirovski, 2015; Guzmán, 2015) or an instantiation of the lower bound by Sridharan and Tewari (2010) (Lemma 15), which we present below.

Proof. We have $d > T$. For $t \in \{1, ..., T\}$, let Y_t be i.i.d. Rademacher random variables, i.e. $\mathbb{P}(Y_t = 1) = \mathbb{P}(Y_t = -1) = 1/2$. Let $e_1, ..., e_d$ be the canonical basis of $\mathbb{R}^d$. At time-step t, denote the point played by $\mathcal{A}$ by x_t and fix the loss to be $\tilde{\ell}_t(x) = Y_t L e_t^T x$. The gradient is $Y_t L e_t$, which is bounded by L in $\ell_{p_\star}$-norm. The point x_t depends on the losses up to time $t-1$ but not on $\tilde{\ell}_t$ and is independent of Y_t, so for all t:

$$\mathbb{E}[\tilde{\ell}_t(x_t)] = \mathbb{E}[Y_t L e_t^T x_t] = \mathbb{E}[Y_t] L e_t^T \mathbb{E}[x_t] = 0 \implies \mathbb{E}[\sum_{t=1}^{T} \tilde{\ell}_t(x_t)] = 0.$$

On the other hand,

$$\min_{x \in \mathcal{B}_p} \sum_{t=1}^{T} \tilde{\ell}_t(x) = L \min_{x \in \mathcal{B}_p} x^T \Big(\sum_{t=1}^{T} Y_t e_t \Big)$$

is attained at $x = -T^{-1/p} \sum_{t=1}^{T} Y_t e_t \in \mathcal{B}_p$, giving

$$\min_{x \in \mathcal{B}_p} \sum_{t=1}^{T} \tilde{\ell}_t(x) = -LT^{-1/p} \sum_{t,t'=1}^{T} Y_t Y_{t'} e_t^T e_{t'} = -LT^{-1/p} \sum_{t=1}^{T} Y_t^2 = -LT^{1-1/p} = -LT^{1/p_\star}.$$

The result follows by: $\sup_{\ell_1, ..., \ell_T} R_T \geqslant \mathbb{E}\Big[\sum_{t=1}^{T} \tilde{\ell}_t(x_t) - \min_{x \in \mathcal{B}_p} \sum_{t=1}^{T} \tilde{\ell}_t(x)\Big] = LT^{1/p_\star}$. □

4.3 Anytime optimality through adaptive regularisation

In the previous section, we saw that the optimal regret is achieved with strongly-convex regularisation in the low-dimensional setting ($d \leqslant T$) and with uniformly-convex regularisation in the high-dimensional setting ($d > T$). To be optimal, the learner with knowledge of the dimension d and the time horizon T can evaluate whether $d > T$ or $d \leqslant T$ and select the appropriate regularisation based on whether the problem is high or low dimensional. However, choosing the correct regulariser relies on knowing whether the problem is high ($d > T$) or low ($d \leqslant T$) dimensional,

which itself relies on knowing the horizon T. In this section, we consider how to achieve anytime optimal regret bounds which hold without knowledge of T.

We consider FTRL with regularisation that adapts to the dimension regime. Fix $t_0 = 3^{-2p/(p-2)}d$. Then, in early high-dimensional rounds, the uniformly-convex ϕ_p is used, until the threshold t_0 when the low-dimensional regime is reached and the regulariser switches to the strongly-convex ϕ_2. In both cases, the step-size used is the one in Corollary 4.1.11. Specifically, with $\phi_r(x) = \frac{1}{r}\|x\|_r^r$ for $r \geqslant 2$, we consider FTRL with regulariser at time t given by

$$\psi_t(x) = \begin{cases} \frac{1}{\eta_{t-1}}\phi_p(x), & \eta_{t-1} = \frac{1}{L(2p_\star t)^{1/p_\star}}, \quad \text{if } t \leqslant t_0, \\ \frac{1}{\eta_{t-1}}\phi_2(x), & \eta_{t-1} = \frac{\sqrt{d^{1-2/p}}}{L\sqrt{2t}}, \quad \text{if } t > t_0. \end{cases} \tag{4.4}$$

FTRL with this sequence of regularisers is anytime optimal as shown by the following theorem.

Theorem 4.3.1. *Let $V = \mathcal{B}_p$ ($p > 2$) and consider proper differentiable convex losses $(\ell_t)_{t=1}^T$ that are L-Lipschitz with respect to $\|\cdot\|_p$. Consider FTRL with regularisers given in (4.4). Then*

$$R_T \leqslant \begin{cases} L\big(2p_\star T\big)^{1/p_\star}, & \text{if } T \leqslant t_0, \\ L\sqrt{2Td^{1-2/p}}, & \text{if } T > t_0 \end{cases}$$

Proof. If $T \leqslant t_0$, then we have FTRL with fixed regulariser ϕ_p and from Corollary 4.1.11 we have $R_T \leqslant L\big(2p_* T\big)^{1/p*}$ as in Section 4.2.2. If $T > t_0$, Theorem 4.1.10 gives

$$\begin{aligned} R_T &\leqslant \psi_T(u) - \min_{x\in V}\psi_1(x) + \sum_{t=1}^{T}\Big\{\frac{(r_t-1)}{r_t\mu_t^{1/(r_t-1)}}\|g_t\|_{|t\star}^{r_t/(r_t-1)} + \psi_t(x_{t+1}) - \psi_{t+1}(x_{t+1})\Big\} \\ &\leqslant \frac{\phi_2(u)}{\eta_{T-1}} - \min_{x\in V}\phi_p(x) + \sum_{t=1}^{t_0}\Big\{2\frac{\eta_{t-1}^{p_\star-1}}{p_\star}\|g_t\|_{p_\star}^{p_\star}\Big\} + \sum_{t=1}^{t_0-1}\Big\{\phi_p(x_{t+1})\Big(\frac{1}{\eta_{t-1}} - \frac{1}{\eta_t}\Big)\Big\} \\ &\quad + \frac{\phi_p(x_{t_0+1})}{\eta_{t_0-1}} - \frac{\phi_2(x_{t_0+1})}{\eta_{t_0}} + \sum_{t=t_0+1}^{T}\Big\{\frac{\eta_{t-1}}{2}\|g_t\|_2^2 + \phi_2(x_{t+1})\Big(\frac{1}{\eta_{t-1}} - \frac{1}{\eta_t}\Big)\Big\} \\ &\leqslant \frac{\sup_{x\in\mathcal{B}_p}\phi_p(x)}{\eta_{t_0-1}} + \sum_{t=1}^{t_0}\Big\{2\frac{\eta_{t-1}^{p_\star-1}}{p_\star}\|g_t\|_{p_\star}^{p_\star}\Big\} + \frac{\phi_2(u)}{\eta_{T-1}} + \sum_{t=t_0+1}^{T}\Big\{\frac{\eta_{t-1}}{2}\|g_t\|_2^2\Big\}. \end{aligned}$$

The first two terms correspond to the regret of FTRL with fixed ϕ_p regularisation on t_0 rounds. Substituting the values of η_{t-1} and some algebra gives (see similar steps in the proof of Corollary

4.1.11)

$$\frac{\sup_{x\in\mathcal{B}_p}\phi_p(x)}{\eta_{t_0-1}}+\sum_{t=1}^{t_0}\Big\{2\frac{\eta_{t-1}^{p_\star-1}}{p_\star}\|g_t\|_{p_\star}^{p_\star}\Big\}\leqslant L\big(2p_\star t_0\big)^{1/p_\star}.$$

The last two terms correspond to the regret of FTRL with fixed ϕ_2 regularisation over the remaining $T-t_0$ rounds.

$$\begin{aligned}\frac{\phi_2(u)}{\eta_{T-1}}+\sum_{t=t_0+1}^{T}\Big\{\frac{\eta_{t-1}}{2}\|g_t\|_2^2\Big\}&=\frac{L\sqrt{d^{1-2/p}T}}{\sqrt{2}}+\frac{L\sqrt{d^{1-2/p}}}{2\sqrt{2}}\sum_{t=t_0+1}^{T}\frac{1}{\sqrt{t}}\\&\leqslant\frac{L\sqrt{d^{1-2/p}T}}{\sqrt{2}}+\frac{L\sqrt{d^{1-2/p}T}}{\sqrt{2}}-\frac{L\sqrt{d^{1-2/p}t_0}}{\sqrt{2}}\\&=L\sqrt{2d^{1-2/p}T}-L\sqrt{d^{1-2/p}t_0/2},\end{aligned}$$

where we used that $\sum_{t=t_0+1}^{T}\frac{1}{\sqrt{t}}\leqslant\int_{t_0}^{T}\frac{1}{\sqrt{x}}dx=\Big[2\sqrt{x}\Big]_{t_0}^{T}=2(\sqrt{T}-\sqrt{t_0})$. Combining, we have

$$R_T\leqslant L\sqrt{2d^{1-2/p}T}+L\big(2p_\star t_0\big)^{1/p_\star}-L\sqrt{d^{1-2/p}t_0/2}.$$

The proof is concluded by $t_0=3^{-2p/(p-2)}d$ guaranteeing $\big(2p_\star t_0\big)^{1/p_\star}-\sqrt{d^{1-2/p}t_0/2}<0$ since

$$(2p_\star t_0)^{1/p_\star}\leqslant\frac{3}{\sqrt{2}}t_0^{1/p_\star}=3t_0^{\frac{p-1}{p}-\frac{1}{2}}\sqrt{t_0/2}=3t_0^{\frac{p-2}{2p}}\sqrt{t_0/2}=\sqrt{d^{1-2/p}t_0/2}.$$

□

The proof consists of a careful application of Theorem 4.1.10. The time-step where the regulariser changes is handled by the specific value of the threshold $t_0=3^{-2p/(p-2)}d$. This value allows us to recover the same low-dimensional bound (including constants) as in Section 4.2.1 achieved using strongly-convex regularisation from the start. For the high-dimensional setting, there is no switch in regulariser so the algorithm and regret bounds are identical to those in Section 4.2.2. In other words, *being agnostic to the dimension regime comes at no cost to the regret bound*. The above procedure can be used with gradient-clipping techniques discussed by Tim van Erven (2021) to obtain a Lipschitz-adaptive anytime optimal algorithm.

Our anytime-optimal procedure is similar to a restarting technique, except that the step-size in the later low-dimensional time-steps accounts for the earlier time-steps. A doubling-trick approach could also be used, though at the cost of worse constants (see the procedure described in Section 4.1.2).

4.4 Necessity of adaptive regularisation

In the previous section, we demonstrated that adaptive regularisation achieves anytime optimal regret bounds for OCO on ℓ_p-balls with $p > 2$. In this section, we show that for sign-invariant regularisers, adaptive regularisation is necessary for anytime optimality. We first show that the regularisers we have considered up to now are provably anytime sub-optimal: in Section 4.4.1 we show that the strongly convex regulariser $\phi_2(x) = \frac{1}{2}\|x\|_2^2$ fails in high-dimension and the uniformly-convex regulariser $\phi_p = \frac{1}{p}\|x\|_p^p$ fails in low-dimension. Then, we present the main result on the failure of using a fixed sign-invariant regulariser in Section 4.4.2, establishing the necessity of the adaptive regularisation discussed in the previous section and a fundamental limitation of FTRL as an algorithmic framework.

As far as we are aware, results on the failure of fixed regularization are novel in online learning. However, algorithmic specific lower bounds (like the ones we have for specific regularizers in Proposition 4.4.1) have appeared in prior work (e.g. for online mirror descent with ℓ_2 or negative entropy regularisations in Theorems 3 & 4 in (Orabona and Pál, 2018)).

4.4.1 Failure of specific regularisers

We saw in Section 4.2.1 that the strongly-convex regulariser $\phi_2(x) = \frac{1}{2}\|x\|_2^2$ achieves the optimal $O(\sqrt{d^{1-2/p}T})$ regret guarantee in the low-dimensional setting ($d \leqslant T$) but this bound is sub-optimal in the high-dimensional setting ($d > T$) because of the polynomial dependence on the dimension. On the other hand, we saw in Section 4.2.2 that the uniformly-convex regulariser of degree p, $\phi_p(x) = \frac{1}{p}\|x\|_p^p$, achieves the optimal $O(\sqrt{T^{1-1/p}})$ regret guarantee in the high-dimensional setting ($d > T$) but this bound is sub-optimal in the low-dimensional setting ($d \leqslant T$) because of the slow rate with respect to T.

In this section, we show that both regularisers are provably suboptimal in their complementary regime. More generally, we establish that no regulariser of the form $\phi_r(x) = \frac{1}{r}\|x\|_r^r$ can interpolate optimally between the low and high dimensional regimes. Our results follow from the following proposition.

Proposition 4.4.1. *For $r \in [2, p]$, define $\phi_r(x) = \frac{1}{r}\|x\|_r^r$. There exists a sequence of linear L-Lipschitz losses (in ℓ_p-norm) for which FTRL with regulariser $\psi_t(x) = \frac{1}{\eta_{t-1}}\phi_r(x)$ and any*

sequence of decreasing η_{t-1} suffers regret

$$R_T \geqslant L \cdot \min\Big(\frac{T}{8r}, \frac{d^{(r_\star - p_\star)/r_\star p_\star} T^{1/r_\star}}{8}\Big).$$

We first present corollaries for the special cases $r = 2$ and $r = p$ before providing the proof below.

Failure of strong convexity in high-dimensions: with $r = 2$, Proposition 4.4.1 gives:

Corollary 4.4.2. *There exists a sequence of linear L-Lipschitz losses (in ℓ_p-norm) for which FTRL with regulariser $\psi_t(x) = \frac{1}{\eta_{t-1}}\phi_2(x)$ and any sequence of decreasing η_{t-1} suffers regret*

$$R_T \geqslant L \cdot \min\Big(\frac{T}{16}, \frac{1}{8}\sqrt{T d^{1-2/p}}\Big).$$

The regret of FTRL with regulariser ϕ_2 scales polynomially with d until it is linear in T. This demonstrates that the analysis from Section 4.2.1 is in fact tight for this regulariser, and this algorithm is sub-optimal in high-dimensions.

Failure of uniform convexity in low-dimension with $r = p$, Proposition 4.4.1 gives:

Corollary 4.4.3. *There exists a sequence of linear L-Lipschitz losses (in ℓ_p-norm) for which FTRL with regulariser $\psi_t(x) = \frac{1}{\eta_{t-1}}\phi_p(x)$ and any sequence of decreasing η_{t-1} suffers regret*

$$R_T \geqslant L \cdot \min\Big(\frac{T}{8p}, \frac{T^{1/p_\star}}{8}\Big).$$

The regret of FTRL with regulariser ϕ_p maintains the rate $T^{1-1/p}$ regardless of the dimension regime. This demonstrates the analysis from Section 4.2.1 is also tight for this regulariser, and this algorithm is sub-optimal in low-dimensions.

More generally, having seen that regularisers $\frac{1}{2}\|x\|_2^2$ (strong convexity) and $\frac{1}{p}\|x\|_p^p$ (uniform convexity of degree p) fail to achieve optimal anytime regret, we may be tempted to consider $\frac{1}{r}\|x\|_r^r$ for $r \in (2, p)$ that could trade-off the optimalities of strong-convexity in low-dimension and of uniform-convexity in high-dimension. However, Proposition 4.4.1 rules out this possibility. We now present its proof.

The loss construction used in the proof below is motivated by the following intuition: The gradients of the losses in the first half of the rounds cancel each other. The competitor is thus

only dependent on the losses in the second half of the rounds which are constant and place the competitor in the corner of the ℓ_p-ball. This two phase construction captures a bias-variance-like trade-off:

- If the step-size is small, FTRL will not suffer much regret in the first half of the rounds but in the second half it will not be able to reach the corner of the ball (the competitor) sufficiently quickly and suffer large-regret (it has high "bias").
- If the step-size is large then FTRL will be able quickly reach the corner of the ball (the competitor) in the second half of the rounds but in the first it will suffer large regret because it moves too fast through the space and every other round it will get to close to e_1 which will make it suffer large loss in the next round (it has high "variance").

This construction is designed so that the performance of FTRL is at its best when the step-size adequately balances the trade-off between the losses from both halves of the rounds. In the case of $\frac{1}{r}\|x\|_r^r$, we can exactly compute the points played by FTRL with the help of Lemma 4.1.1 and as a result explicitly compute the regret suffered by FTRL for all step-sizes and obtain the lower bound in Proposition 4.4.1.

Proof of Proposition 4.4.1. Assume that T is divisible by 4 (otherwise use $T-1$, $T-2$, or $T-3$). For clarity of exposition, we construct the losses to be 1-Lipschitz with respect to the ℓ_p norm; the general case of L-Lipschitz losses follows immediately by scaling the construction by a factor of L. We construct a sequence of linear losses of the form $\ell_t(x) = \langle x, g_t \rangle$, where $g_t \in \mathcal{B}_{p_\star}$ is defined as

$$g_t = \begin{cases} (-1)^t e_1, & t \leqslant \frac{T}{2}, \\ -v, & t > \frac{T}{2}, \end{cases}$$

and $v \in \mathcal{B}_{p_\star}$ is the vector with equal coordinates $v_i = d^{-1/p_\star}$, so that $\|v\|_{p_\star} = 1$ and $\|v\|_p = d^{1/p-1/p_\star}$.

Comparator loss. For any $x \in \mathcal{B}_p$, the cumulative loss of the comparator is

$$\sum_{t=1}^{T} \ell_t(x) = \frac{LT}{2}\langle x, v \rangle, \tag{4.5}$$

which yields

$$\min_{x\in\mathcal{B}_p}\sum_{t=1}^{T}\ell_t(x) = -\frac{LT}{2}\frac{\langle v, v\rangle}{\|v\|_p} = -\frac{LT}{2}.$$

Cumulative gradients. The cumulative sum of gradients appearing in the FTRL updates satisfies

$$\sum_{s=1}^{t-1} g_s = \begin{cases} -e_1, & \text{if } t \leqslant \frac{T}{2} \text{ and } t \text{ is even,} \\ 0, & \text{if } t \leqslant \frac{T}{2} \text{ and } t \text{ is odd,} \\ -\left(t-1-\frac{T}{2}\right)v, & \text{if } t > \frac{T}{2}. \end{cases} \tag{4.6}$$

FTRL iterates. By Lemma 4.1.1, the FTRL update can be written as

$$x_{t+1} = \arg\min_{x\in\mathcal{B}_p} D_{\phi_r}\left(x,\ \operatorname{sign}\Big(-\eta_t\sum_{s=1}^{t} g_s\Big)\Big|-\eta_t\sum_{s=1}^{t} g_s\Big|^{r_\star-1}\right).$$

Define $\alpha_{t-1} := \min\{1, \eta_{t-1}\}$. Using (4.6), the iterates take the following explicit form:

- If $t \leqslant T/2$ is odd, then $x_t = 0$.
- If $t \leqslant T/2$ is even, then $x_t = \alpha_{t-1}^{r_\star-1} e_1$, by Lemma 4.1.3.
- If $t > T/2$, then

$$\begin{aligned} x_t &= \min\left\{\frac{1}{\|w\|_p}, \left(\eta_{t-1} d^{-1/p_\star}\left(t-1-\frac{T}{2}\right)\right)^{r_\star-1}\right\} w \\ &= \min\left\{1, d^{1-r_\star/p_\star}\left(\eta_{t-1}\left(t-1-\frac{T}{2}\right)\right)^{r_\star-1}\right\}\frac{v}{\|v\|_p}, \end{aligned}$$

  by Lemma 4.1.2, where w denotes the all-ones vector.

Loss in the first half. Fix $\eta := \eta_{T/2-1}$ and $\alpha := \min\{1, \eta\}$. Since $\eta_{t-1} \geqslant \eta_t$, the cumulative loss incurred during the first $T/2$ rounds satisfies

$$\sum_{t=1}^{T/2}\ell_t(x_t) = \sum_{k=1}^{T/4}\ell_{2k}(x_{2k}) = \sum_{k=1}^{T/4}\alpha_{2k-1}^{r_\star-1} \geqslant \frac{\alpha^{r_\star-1}T}{4}.$$

If $\alpha > 1/2$, then $R_T \geqslant T/8 \geqslant T/(4r)$ and we are done. Henceforth assume $\alpha = \eta \leqslant 1/2$.

Loss in the second half. Define

$$k^\star := \Big\lfloor \frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{\eta} \Big\rfloor, \qquad m := \min\Big\{k^\star, \frac{T}{2}-1\Big\}.$$

Using $\langle v, v\rangle = \|v\|_p$, we lower bound the loss in the second half as

$$\begin{aligned}
\sum_{t=T/2+1}^{T} \ell_t(x_t) = -\sum_{t=T/2+1}^{T} x_t^T v &\geqslant -\sum_{k=1}^{T/2-1} \min\Big\{1, d^{1-r_\star/p_\star}(\eta k)^{r_\star-1}\Big\} \cdot \frac{v^T v}{\|v\|_p} \\
&= -\sum_{k=1}^{T/2-1} \min\Big\{1, d^{1-r_\star/p_\star}(\eta k)^{r_\star-1}\Big\} \\
&= -d^{1-r_\star/p_\star}\sum_{k=1}^{m}(\eta k)^{r_\star-1} - \Big(\frac{T}{2}-1-m\Big).
\end{aligned}$$

Bounding the sum by an integral yields

$$\sum_{k=1}^{m} k^{r_\star-1} \leqslant \int_0^m (x+1)^{r_\star-1}\,dx = \frac{1}{r_\star}\Big[(x+1)^{r_\star}\Big]_0^m \leqslant \frac{1}{r_\star}(m+1)^{r_\star}.$$

and hence

$$\sum_{t=T/2+1}^{T} \ell_t(x_t) \geqslant -d^{1-r_\star/p_\star}\frac{\eta^{r_\star-1}}{r_\star}(m+1)^{r_\star} - \left(\frac{T}{2}-1-m\right).$$

Regret decomposition. Combining both phases with (4.5), the regret satisfies

$$\begin{aligned}
R_T &\geqslant \frac{T}{2} + \frac{\eta^{r_\star-1}T}{4} - d^{1-r_\star/p_\star}\frac{\eta^{r_\star-1}}{r_\star}(m+1)^{r_\star} - \Big(\frac{T}{2}-1-m\Big) \\
&= \frac{\eta^{r_\star-1}T}{4} - d^{1-r_\star/p_\star}\frac{\eta^{r_\star-1}}{r_\star}(m+1)^{r_\star} + (1+m).
\end{aligned}$$

We now distinguish two cases.

Case 1: $k^\star \geqslant T/2-1$**.** Then $m = T/2-1$. By the definition of $k^\star$,

$$\begin{aligned}
\frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{\eta} \geqslant \Big\lfloor \frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{\eta} \Big\rfloor = k^\star \geqslant \frac{T}{2}-1 \implies \frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}}\Big(\frac{T}{2}-1\Big) \leqslant 1 \\
\implies \frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}}\frac{T}{2} \leqslant 1 + \frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}} \leqslant \frac{3}{2},
\end{aligned}$$

since $\eta \leqslant 1/2$ and $d^{(r_\star/p_\star-1)/(r_\star-1)} \geqslant 1$ (recall $r \leqslant p$). Substituting this into the regret bound gives

$$R_T \geqslant -d^{1-r_\star/p_\star}\frac{\eta^{r_\star-1}}{r_\star}\Big(\frac{T}{2}\Big)^{r_\star} + \frac{T}{2}$$

$$
\begin{aligned}
&= -\frac{1}{r_\star}\Big(\frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}}\frac{T}{2}\Big)^{r_\star-1}\frac{T}{2} + \frac{T}{2} \\
&\geqslant -\frac{1}{r_\star}\Big(\frac{3}{2}\Big)^{r_\star-1}\frac{T}{2} + \frac{T}{2} \\
&= \frac{T}{2}\Big(1 - \frac{1}{r_\star}\Big(\frac{3}{2}\Big)^{r_\star-1}\Big) \\
&\geqslant \frac{T}{4}\Big(1 - \frac{1}{r_\star}\Big) = \frac{T}{4r},
\end{aligned}
$$

where we used that $r_\star \in [1,2]$ and that $f(x) = 1 - \frac{(3/2)^{x-1}}{x} \geqslant \frac{1}{2}(1-1/x)$ for $x \in [1,2]$.

Case 2: $k^\star < T/2 - 1$**.** Then $m = k^\star$. Proceeding analogously,

$$
\begin{aligned}
\frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{\eta} \geqslant \Big\lfloor \frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{\eta} \Big\rfloor = k^\star \implies \frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}} k^\star \leqslant 1 \\
\implies \frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}}(k^\star+1) \leqslant 1 + \frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}} \leqslant \frac{3}{2}.
\end{aligned}
$$

We also have $k^\star + 1 \geqslant \frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{\eta}$. Substituting into the regret expression yields

$$
\begin{aligned}
R_T &\geqslant \frac{\eta^{r_\star-1}T}{4} - d^{1-r_\star/p_\star}\frac{\eta^{r_\star-1}}{r_\star}(k^\star+1)^{r_\star} + (1+k^\star) \\
&= \frac{\eta^{r_\star-1}T}{4} - \frac{k^\star+1}{r_\star}\Big(\frac{\eta}{d^{(r_\star/p_\star-1)/(r_\star-1)}}(k^\star+1)\Big)^{r_\star-1} + (1+k^\star) \\
&\geqslant \frac{\eta^{r_\star-1}T}{4} - \frac{1+k^\star}{r_\star}\Big(\frac{3}{2}\Big)^{r_\star-1} + (1+k^\star) \\
&= \frac{\eta^{r_\star-1}T}{4} + (1+k^\star)\Big(1 - \frac{1}{r_\star}\Big(\frac{3}{2}\Big)^{r_\star-1}\Big) \\
&\geqslant \frac{\eta^{r_\star-1}T}{4} + \frac{(1+k^\star)}{2}\Big(1 - \frac{1}{r_\star}\Big) \\
&= \frac{\eta^{r_\star-1}T}{4} + \frac{(1+k^\star)}{2r} \\
&\geqslant \frac{\eta^{r_\star-1}T}{4} + \frac{d^{(r_\star/p_\star-1)/(r_\star-1)}}{2r\eta} \\
&\geqslant \frac{{r_\star}^{1/r_\star}}{2^{1/r+2/r_\star}} d^{(r_\star/p_\star-1)/r_\star} T^{1/r_\star} \geqslant \frac{d^{(r_\star-p_\star)/r_\star p_\star} T^{1/r_\star}}{4}
\end{aligned}
$$

where again we used that $1 - \frac{1}{r_\star}\Big(\frac{3}{2}\Big)^{r_\star-1} \geqslant \frac{1}{2}(1-1/r_\star)$ since $r_\star \in [1,2]$ and in the last step we minimised over η.

Conclusion. Combining both cases,

$$
R_T \geqslant \min\left\{ \frac{T}{4r}, \frac{d^{(r_\star-p_\star)/(r_\star p_\star)} T^{1/r_\star}}{4} \right\}.
$$

If T is not divisible by 4, replacing T by $T-1$, $T-2$, or $T-3$ yields the stated bound for all $T \geqslant 6$. □

4.4.2 Failure of fixed sign-invariant regularisation

In this section, we study the anytime optimality of FTRL under a fixed regulariser. While the previous two sections established that the two specific regularisers we considered in Section 4.2 (which are each optimal for one regime) are not able to guarantee anytime optimality, it does not rule out the existence of a regulariser that could. If such a regulariser existed, the complexity of adaptive regularisation could be avoided entirely in favour of a static strategy. We now provide a definitive negative answer to this question.

We consider the class of sign-invariant regularisers defined as

$$\Psi = \big\{\psi : \mathcal{B}_p \to \mathbb{R} : \text{strictly convex, sign-invariant, } 0 = \operatorname*{arg\,min}_{x\in\mathbb{R}} \psi(x), \psi(0) = 0, \psi(1/2e_1) = 1\big\},$$

where a function $f : \mathcal{X} \subset \mathbb{R}^d \to \mathbb{R}$ is sign-invariant if for any $s \in \{-1, 1\}^d$, $f(s \cdot x) = f(x)$ for all $x \in \mathcal{X}$ where $s \cdot x$ denotes coordinate-wise multiplication.

The class Ψ is general since sign-invariance is a mild natural assumption for a regulariser (and any regulariser can be scaled to satisfy the other conditions). It encompasses virtually all standard regularisers used in the literature, including any based on ℓ_r-norms.

The following theorem represents the core theoretical contribution of this chapter on the failure of fixed sign-invariant regularisation. We present its proof in Section 4.5.

Theorem 4.4.4. *Let $V = \mathcal{B}_p$ ($p > 2$) and consider proper differentiable convex losses $(\ell_t)_{t=1}^T$ that are L-Lipschitz with respect to $\|\cdot\|_p$. FTRL (4.1) with fixed regulariser $\psi_t(x) = \frac{1}{\eta}\psi(x)$ for all t and $\psi \in \Psi$ cannot satisfy both:*

1. *There exists $c > 0$ such that for all T, d, $T > d$, there exists η such that $R_T \leqslant cL\sqrt{Td^{1-2/p}}$.*
2. *There exists $c' > 0$ such that for all T, d, $d > T$, there exists η such that $R_T \leqslant c'LT^{1-1/p}$.*

In other words, it cannot be optimal both in the low and high dimensional settings.

Theorem 4.4.4 reveals that the failure of fixed regularisation is not a mere technicality or an artifact of a poor choice of regulariser, but a structural property of ℓ_p-geometry when $p > 2$.

There is a necessity for the regularisation to be adaptive to the dimension-regime for anytime optimality when using FTRL with sign-invariant regularisers from Ψ. In particular, this highlights a fundamental limitation of the FTRL framework which cannot intrinsically adapt to changes in the geometry induced by the interaction between the dimension and the time horizon.

Proposition 4.4.1 establishes regret lower bounds for specific fixed regularisers of the form $\frac{1}{r}\|x\|_r^r$, showing that these fail to achieve optimal anytime regret. In particular, for $r < p$ and sufficiently large dimension, the regret becomes linear. Theorem 4.4.4 is much more general since it shows that no sign-invariant regulariser can achieve optimal anytime regret. Moreover, its proof reveals a stronger structural statement: any regulariser that achieves the optimal low-dimensional regret (corresponding to the first condition of the theorem) must incur linear regret in sufficiently high dimensions.

A complementary impossibility result to universality. This result provides a critical missing piece to the theory of algorithmic universality. We have seen in Section 4.1.2 that FTRL is a universally optimal OCO learning algorithm capable of matching any optimal rate up to logarithmic factors, even when the rate with respect to T is non-constant as in our setting. However, this result uses a doubling trick where different regularisers are used across separate intervals. Theorem 4.4.4 establishes that FTRL with fixed regularisation is in fact not universal for general OCO problems where the optimal rate of regret is horizon or dimension dependent. This highlights a fundamental boundary in OCO: while mirror-descent type algorithms are powerful, their optimality in complex geometries depends on their capacity to adapt.

Remark 4.4.5. *Our results also hold in an unconstrained domain where the competitor is constrained to be in $\mathcal{B}_p$ using the reduction from Cutkosky and Orabona (2018).*

Remark 4.4.6. *Theorem 4.4.4 is stated for constant step-size η. We believe the result should hold for any decreasing step-size in the same way as Proposition 4.4.1. Decreasing step-sizes do circumvent the key ideas of the proof but would make it even more complicated and tedious than it already is.*

4.5 Proof of Theorem 4.4.4

In this section, we detail the proof of Theorem 4.4.4. For clarity of exposition, as in the proof of Proposition 4.4.1 we construct the losses to be 1-Lipschitz with respect to the ℓ_p norm; the

general case of L-Lipschitz losses follows immediately by scaling the construction by a factor of L.

The argument is as follows. We assume condition 1 in the statement of the theorem is satisfied, i.e.

1. There exists $c > 0$ such that for all T, d, $T > d$, there exists η such that $R_T \leqslant c\sqrt{Td^{1-2/p}}$,

and show that condition 2 cannot be satisfied.

Step 1 - dimension dependent regularisation: We show that to satisfy condition 1, there exists a point w_d in the ℓ_p-ball where the value of the regulariser ψ must scale with d. We construct this point iteratively. Fix $\varepsilon = \frac{p-2}{4}$ and $\lambda < \frac{3}{4}\left(\varepsilon \log^{\varepsilon} 3\right)^{1/p}$. For $n \geqslant 4$, define

$$w_n = \lambda \cdot \sum_{i=4}^{n} \frac{1}{(i \log^{1+\varepsilon} i)^{1/p}} e_i.$$

We have $w_n \in \mathcal{B}_p$ for all n since:

$$\|w_n\|_p^p = \lambda^p \cdot \sum_{i=4}^{n} \frac{1}{i \log^{1+\varepsilon} i} \leqslant \lambda^p \int_3^{\infty} \frac{1}{x \log^{1+\varepsilon} x} dx = \lambda^p \int_{\log 3}^{\infty} u^{-1-\varepsilon} du = \lambda^p \cdot \Big[-\frac{1}{\varepsilon} u^{-\varepsilon}\Big]_{\log 3}^{\infty} = \frac{\lambda^p}{\varepsilon \log^{\varepsilon} 3} < 1,$$

where we used a $u = \log x$ substitution.

The following proposition formalises the growth of the regulariser along the sequence of w_n when satisfying condition 1. We prove it in Section 4.5.1.

Proposition 4.5.1. *Fix $d_0 := \exp\Big((64c^2)^{1/(1/2-1/p)}\Big)$. If for all T, d, $T > d > d_0$, there exists η such that FTRL with fixed regulariser $\psi_t(x) = \frac{1}{\eta}\psi(x)$ for all t and $\psi \in \Psi$ satisfies $R_T \leqslant c\sqrt{Td^{1-2/p}}$ against any sequence of 1-Lipschitz losses (in ℓ_p norm), then for all $d \geqslant d_0$*

$$\psi(w_d) - \psi(w_{d-1}) \geqslant \frac{\lambda^2}{d \log d}.$$

Since we have assumed condition 1, we have for $d \geqslant d_0$

$$\psi(w_d) \geqslant \psi(w_{d-1}) \geqslant \sum_{i=d_0}^{d-1} \frac{\lambda^2}{i \log i} \geqslant \int_{d_0}^{d} \frac{\lambda^2}{x \log x} dx = \lambda^2 [\log\log x]_{d_0}^{d} = \lambda^2(\log\log d - \log\log(d_0)),$$

which is the scaling with d we were seeking in the first part of this proof.

Step 2 - linear regret in high dimension: by exploiting the dimension dependent regularisation, we can construct a high-dimensional loss sequence on which FTRL with this regulariser will suffer linear regret, contradiction condition 2. This is established formally in the following proposition that is proved in Section 4.5.4.

Proposition 4.5.2. *Let $\psi \in \Psi$ satisfy for $d \geqslant d_0$ and $\lambda < \frac{3}{4}(\varepsilon \log^{\varepsilon} 3)^{1/p}$, $\psi(w_d) \geqslant \lambda^2(\log\log d - \log\log(d_0))$. Fix T. If*

$$d \geqslant \exp\left(\exp\left(\log\log d_0 + \frac{8T}{(\varepsilon \log^{\varepsilon} 3)^{2/p}}\right)\right), \qquad \text{recall } \varepsilon = \frac{p-2}{4}$$

there exists a sequence of linear 1-Lipschitz losses (in ℓ_p-norm) for which FTRL with fixed regulariser $\psi_t(x) = \frac{1}{\eta}\psi(x)$ for all t and any η suffers regret $R_T \geqslant \frac{1}{8}T$.

This contradicts condition 2 and concludes the proof. □

4.5.1 Proof of Proposition 4.5.1

We provide the proof of Proposition 4.5.1. Suppose the claim does not hold, so that there exists a $d \geqslant d_0$ such that

$$\psi(w_d) - \psi(w_{d-1}) < \frac{\lambda^2}{d\log d} \tag{4.7}$$

The proof consists in reaching a contradiction and the main idea is to construct a loss sequence, which

- in the first half of the rounds drags the points played by FTRL to w_{d-1}.
- in the second half of the rounds, where the points played by FTRL can be thought of as centred at w_{d-1}, we alternate the losses between $\pm e_d^T x$ (whose gradients $\pm e_d$ are in the direction of $w_d - w_{d-1}$).

We can show that the regret suffered in the second half of the rounds will be large because there is only little difference in regularisation between w_{d-1} and w_d due to our assumption.
Formally, fix $T > d$ a multiple of 4 such that

$$\frac{1}{32c^2\cdot\lambda^2} d\log(d)^{2-2(1+\varepsilon)/p} \leqslant T \leqslant \frac{1}{16c^2\cdot\lambda^2} d\log(d)^{2-2(1+\varepsilon)/p}.$$

This choice of a multiple of 4 is guaranteed by $d \geqslant d_0$, which also guarantees $T > d$. The statement of the proposition assumes that there exists η such that FTRL with fixed regulariser

$\psi_t(x) = \frac{1}{\eta}\psi(x)$ for all t and $\psi \in \Psi$ satisfies $R_T \leqslant c\sqrt{Td^{1-2/p}}$. In Section 4.5.2, we show that to satisfy this assumption, we must have

$$\eta \geqslant \frac{1}{2c\sqrt{Td^{1-2/p}}} \tag{4.8}$$

We now describe the details of the the loss construction on which we compute a lower bound on the regret suffered by FTRL.

Step 1 - First half of the rounds, dragging FTRL to w_{d-1}: We construct the losses in such a way that $x_{T/2} = w_{d-1}$. Define $G = -\frac{1}{\eta}\nabla\psi(w_{d-1})$ and for $t < T/2$, $\ell_t(x) = g_t^T x$ with $g_t = \frac{1}{T/2-1} \cdot G$. We have $\|g_t\|_{p_\star} \leqslant 1$ and $g_t \in \mathcal{B}_{p_\star}$ as required since we show in see Section 4.5.3 that

$$\|G\|_{p_\star} \leqslant T/2 - 1. \tag{4.9}$$

We obtain

$$\begin{aligned} x_{T/2} &= \arg\min_{x\in\mathcal{B}_p}\Big\{\psi(x) + \eta \sum_{t=1}^{T/2-1} \langle g_t, x\rangle\Big\} \\ &= \arg\min_{x\in\mathcal{B}_p}\big\{\psi(x) + \eta\langle G, x\rangle\big\} \\ &= w_{d-1} \quad \text{since } G = -\frac{1}{\eta}\nabla\psi(w_{d-1}) \iff \nabla\psi(w_{d-1}) + \eta \cdot G = 0 \text{ and } w_{d-1} \in \operatorname{int} \mathcal{B}_p. \end{aligned}$$

Note that $G^T e_d = 0$ from the sign invariance of ψ. To see this, observe that by strict convexity of ψ, we have that 0 is the unique minimiser of $g(z) = \psi(w_{d-1} + z \cdot e_d)$. If there was a $z \neq 0$ that was a minimiser, then $-z$ would also be a minimiser by sign invariance but this would contradict the strict convexity of g induced by ψ. Hence $\nabla\psi(w_{d-1})^T e_d = 0 \implies G^T e_d = 0$.

Step 2 - Second half of the rounds, large regret via alternating losses in direction of $e_d \propto w_d - w_{d-1}$: For $t \geqslant T/2$, set the loss to be $\ell_t(x) = (-1)^{t+1} e_d^T x$ (i.e. we alternate between $\pm e_d$). This gives for all $k = 0, 1, ..., T/4 - 1$:

$$\begin{aligned} x_{T/2+1+2k} = x_{T/2+1} &= \arg\min_{x\in\mathcal{B}_p}\big\{\psi(x) + \eta \cdot \langle G - e_d, x\rangle\big\} \\ x_{T/2+2k} &= x_{T/2} = w_{d-1}. \end{aligned}$$

We want to lower-bound $e_d^T x_{T/2+1}$, the loss suffered in each of these alternating rounds. Recall

that $w_d = \lambda \cdot \sum_{i=4}^{d} \frac{1}{(i \log^{1+\varepsilon} i)^{1/p}} e_i = w_{d-1} + \frac{\lambda}{(d \log^{1+\varepsilon} d)^{1/p}} e_d$. We begin with the FTRL objective at round $T/2+1$ evaluated at w_d:

$$\begin{aligned}
\psi(w_d) + \eta \cdot (G - e_d)^T w_d &= \underbrace{\psi(w_d)}_{\text{by (4.7)}} + \eta \cdot G^T w_{d-1} - \eta \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} \\
&< \overbrace{\psi(w_{d-1}) + \frac{\lambda^2}{d \log d}} + \eta \cdot G^T w_{d-1} - \eta \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} \\
&= \underbrace{\psi(x_{T/2}) + \eta \cdot G^T x_{T/2}}_{\text{by the definition of } x_{T/2}} + \frac{\lambda^2}{d \log d} - \eta \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} \\
&\leqslant \overbrace{\psi(x_{T/2+1}) + \eta \cdot G^T x_{T/2+1}} + \frac{\lambda^2}{d \log d} - \eta \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} \\
&= \underbrace{\psi(x_{T/2+1}) + \eta \cdot (G - e_d)^T x_{T/2+1}}_{\text{by the definition of } x_{T/2+1}} + \eta \cdot e_d^T x_{T/2+1} + \frac{\lambda^2}{d \log d} - \eta \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} \\
&\leqslant \overbrace{\psi(w_d) + \eta \cdot (G - e_d)^T w_d} + \eta \cdot e_d^T x_{T/2+1} + \frac{\lambda^2}{d \log d} - \eta \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} \\
\implies e_d^T x_{T/2+1} &> \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} - \frac{1}{\eta} \frac{\lambda^2}{d \log d} \\
&\geqslant \frac{\lambda}{(d \log^{1+\varepsilon}(d))^{1/p}} - 2c\sqrt{T d^{1-2/p}} \cdot \frac{\lambda^2}{d \log d} \quad \text{by condition on } \eta \text{ in (4.8)}
\end{aligned}$$

Recall that T satisfies

$$\begin{aligned}
\frac{1}{32c^2 \cdot \lambda^2} d \log(d)^{2-2(1+\varepsilon)/p} \quad &\leqslant \quad T \quad \leqslant \quad \frac{1}{16c^2 \cdot \lambda^2} d \log(d)^{2-2(1+\varepsilon)/p} \\
\implies \frac{\lambda}{d^{1/p} \log^{(1+\varepsilon)/p}(d)} - 2c\sqrt{T d^{1-2/p}} \cdot \frac{\lambda^2}{d \log d} &\geqslant \frac{\lambda}{2 d^{1/p} \log^{(1+\varepsilon)/p}(d)} \\
\implies R_T \geqslant \frac{T}{4} \cdot e_d^T x_{T/2+1} &> \frac{T\lambda}{8 d^{1/p} \log^{(1+\varepsilon)/p}(d)} \\
&\geqslant \frac{1}{256c^2 \cdot \lambda} d^{1-1/p} \cdot \log(d)^{2-3(1+\varepsilon)/p}.
\end{aligned}$$

Now note that $c\sqrt{T d^{1-2/p}} \leqslant \frac{1}{4\lambda} \cdot d^{1-1/p} \cdot \log(d)^{1-(1+\varepsilon)/p}$ so $R_T \leqslant c\sqrt{T d^{1-2p}}$ is contradicted if

$$\begin{aligned}
\frac{1}{256c^2 \cdot \lambda} d^{1-1/p} \cdot \log(d)^{2-3(1+\varepsilon)/p} &> \frac{1}{4\lambda} \cdot d^{1-1/p} \cdot \log(d)^{1-(1+\varepsilon)/p} \\
&\iff \log(d)^{1-2(1+\varepsilon)/p} > 64c^2 \\
&\iff \log(d)^{1/2-1/p} > 64c^2 \quad \text{since } \varepsilon = \frac{p-2}{4} \\
&\iff d > \exp((64c^2)^{1/(1/2-1/p)}) = d_0.
\end{aligned}$$

This condition on d was imposed in the statement of the proposition so we have $R_T > c\sqrt{T d^{1-2/p}}$

and we get a contradiction that concludes the proof.

4.5.2 Lower bound on η

In the previous section, we used a lower-bound on η in (4.8) that we prove here. We show that if FTRL with a fixed regulariser $\psi_t(x) = \frac{1}{\eta}\psi(x)$ for all t, where $\psi \in \Psi$, satisfies a regret bound of the form

$$R_T \leqslant c\sqrt{Td^{1-2/p}},$$

then necessarily

$$\eta \geqslant \frac{1}{2c\sqrt{Td^{1-2/p}}}.$$

The idea is to consider identical losses across all rounds and show that if η is too small, the FTRL updates are too slow to respond to the accumulated loss, leading to large regret.
Formally, consider the linear losses $\ell_t(x) = e_1^\top x$ for all t. The regret can then be written as

$$R_T = \max_{u \in V} \sum_{t=1}^{T} \big(\ell_t(x_t) - \ell_t(u)\big) = T - \sum_{t=1}^{T} e_1^\top x_t.$$

- **Lower bound on the iterate.** We first show that

$$e_1^\top x_{\lceil 2c\sqrt{Td^{1-2/p}}\rceil} \geqslant \frac{1}{2}.$$

Suppose instead that $e_1^\top x_t < \frac{1}{2}$ for all $t \leqslant \lceil 2c\sqrt{Td^{1-2/p}}\rceil$. Then

$$\begin{aligned} R_T &= T - \sum_{t=1}^{T} e_1^\top x_t \\ &> T - \sum_{t=1}^{\lceil 2c\sqrt{Td^{1-2/p}}\rceil} \frac{1}{2} - \sum_{t=\lceil 2c\sqrt{Td^{1-2/p}}\rceil+1}^{T} 1 \\ &= \frac{1}{2}\lceil 2c\sqrt{Td^{1-2/p}}\rceil > c\sqrt{Td^{1-2/p}}, \end{aligned}$$

which contradicts the assumed regret bound. Therefore,

$$e_1^\top x_{\lceil 2c\sqrt{Td^{1-2/p}}\rceil} \geqslant \frac{1}{2}.$$

Note that the condition $\lceil 2c\sqrt{Td^{1-2/p}}\rceil < T$ holds whenever $T > 4c^2 d^{1-2/p}$.

- **Lower bound on the step size.** By the definition of the FTRL update,

$$x_t = \arg\min_{x \in V} \left\{ \psi(x) - \eta \sum_{s=1}^{t-1} e_1^\top x \right\}.$$

Applying this at $t = \lceil 2c\sqrt{Td^{1-2/p}} \rceil$ and comparing the optimum to $x = 0$ yields

$$\psi\Big(x_{\lceil 2c\sqrt{Td^{1-2/p}}\rceil}\Big) - \eta \lceil 2c\sqrt{Td^{1-2/p}} \rceil e_1^\top x_{\lceil 2c\sqrt{Td^{1-2/p}}\rceil} \leqslant \psi(0) - \eta \sum_{s=1}^{t-1} e_1^\top 0 = 0.$$

Using the fact that $\psi(x_{\lceil 2c\sqrt{Td^{1-2/p}}\rceil}) \geqslant \psi(\frac{1}{2}e_1)$ since any coordinate other than the first being non-zero increases ψ and $e_1^\top x_{\lceil 2c\sqrt{Td^{1-2/p}}\rceil} \leqslant 1$, we obtain

$$\begin{aligned} &\psi\big(\tfrac{1}{2}e_1\big) \leqslant \eta \lceil 2c\sqrt{Td^{1-2/p}} \rceil \\ &\implies \eta \geqslant \frac{\psi(\frac{1}{2}e_1)}{\lceil 2c\sqrt{Td^{1-2/p}} \rceil} \geqslant \frac{\psi(\frac{1}{2}e_1)}{2c\sqrt{Td^{1-2/p}}} = \frac{1}{2c\sqrt{Td^{1-2/p}}}. \end{aligned}$$

4.5.3 Upper bound on $\|G\|_{p_\star}$

In this section, we continue to assume that FTRL with the fixed regulariser $\psi_t(x) = \frac{1}{\eta}\psi(x)$ for all t, where $\psi \in \Psi$, satisfies a regret bound of the form $R_T \leqslant c\sqrt{Td^{1-2/p}}$.

We want to show (4.9), an upper bound on the $\ell_{p_\star}$ norm of the gradient of $\psi(w_{d-1})$, which in the proof of Postposition 4.5.1 allows us to show that w_{d-1} is reachable in $T/2$ rounds by the version of FTRL we consider.

Claim 4.5.3. *We have*

$$\|\nabla\psi(w_{d-1})\|_{p_\star} \leqslant \eta\left(\frac{T}{2} - 1\right).$$

Proof. For notational convenience, set

$$u := \nabla\psi(w_{d-1}), \qquad q := p_\star, \qquad T_0 := \frac{T}{2} - 1.$$

The intuition is that if the claim were false, then the step size η would be too small for FTRL to move sufficiently towards w_{d-1} within T_0 rounds, resulting in large regret. To formalise this idea, we consider linear losses for which the optimal comparator is aligned with w_{d-1}:

$$\ell_t(x) := -\frac{1}{\|u\|_q}\langle x, u\rangle, \qquad t = 1, \dots, T.$$

The corresponding FTRL updates satisfy

$$x_{t+1} = \arg\min_{x \in \mathcal{B}_p} \left\{ \psi(x) - \frac{\eta t}{\|u\|_q} \langle x, u \rangle \right\}.$$

- **Lower bound on the iterate.** We first show that

$$\frac{1}{\|u\|_q} \langle x_{T_0+1}, u \rangle \geqslant \frac{3}{4}.$$

Suppose otherwise. Then, using the fact that $\langle x_t, u \rangle \leqslant \langle x_{T_0+1}, u \rangle$ for all $t \leqslant T_0$, we obtain

$$\begin{aligned} R_T &= T - \sum_{t=1}^{T} \ell_t(x_t) \\ &\geqslant T_0 \left(1 - \frac{1}{\|u\|_q} \langle x_{T_0+1}, u \rangle \right) \geqslant \frac{T_0}{4} = \frac{T}{8} - \frac{1}{4}, \end{aligned}$$

which contradicts the assumed regret bound $R_T \leqslant c\sqrt{T d^{1-2/p}}$ for sufficiently large T.

To justify $\langle x_t, u \rangle \leqslant \langle x_{T_0+1}, u \rangle$, note that since x_{t+1} and x_{T_0+1} attain the arg min of the update at their respective time-steps,

$$\begin{aligned} \psi(x_{t+1}) - \frac{\eta t}{\|u\|_q} \langle x_{t+1}, u \rangle &\leqslant \psi(x_{T_0+1}) - \frac{\eta t}{\|u\|_q} \langle x_{T_0+1}, u \rangle \\ &= \psi(x_{T_0+1}) - \frac{\eta T_0}{\|u\|_q} \langle x_{T_0+1}, u \rangle + \frac{\eta (T_0 - t)}{\|u\|_q} \langle x_{T_0+1}, u \rangle \\ &\leqslant \psi(x_{t+1}) - \frac{\eta T_0}{\|u\|_q} \langle x_{t+1}, u \rangle + \frac{\eta (T_0 - t)}{\|u\|_q} \langle x_{T_0+1}, u \rangle \\ \implies \frac{\eta (T_0 - t)}{\|u\|_q} \langle x_{t+1}, u \rangle &\leqslant \frac{\eta (T_0 - t)}{\|u\|_q} \langle x_{T_0+1}, u \rangle \\ \implies \langle x_{t+1}, u \rangle &\leqslant \langle x_{T_0+1}, u \rangle. \end{aligned}$$

- **Separation from w_{d-1}.** By Hölder's inequality,

$$\left\langle w_{d-1}, \frac{u}{\|u\|_q} \right\rangle \leqslant \|w_{d-1}\|_p \left\| \frac{u}{\|u\|_q} \right\|_q \leqslant \frac{\lambda}{(\varepsilon \log^{\varepsilon} 3)^{1/p}} < \frac{3}{4},$$

which implies

$$\langle x_{T_0+1} - w_{d-1}, u \rangle > 0.$$

- **Conclusion.** By the optimality of x_{T_0+1} at time T_0,

$$\psi(w_{d-1}) - \frac{\eta T_0}{\|u\|_q}\langle w_{d-1}, u\rangle \geqslant \psi(x_{T_0+1}) - \frac{\eta T_0}{\|u\|_q}\langle x_{T_0+1}, u\rangle.$$

Using convexity of ψ, we further have

$$\psi(x_{T_0+1}) \geqslant \psi(w_{d-1}) + \langle u, x_{T_0+1} - w_{d-1}\rangle.$$

Combining the two inequalities yields

$$\left(\frac{\eta T_0}{\|u\|_q} - 1\right)\langle x_{T_0+1} - w_{d-1}, u\rangle \geqslant 0.$$

Since $\langle x_{T_0+1} - w_{d-1}, u\rangle > 0$, it follows that $\|u\|_q \leqslant \eta T_0$, completing the proof.

□

4.5.4 Proof of Proposition 4.5.2

We show that when the dimension is large enough, FTRL with a regulariser that imposes dimension-dependent regularisation on w_d suffers linear regret. We distinguish two cases depending on the step size η.

Case 1: $\eta > 4$**.** In this regime, the step size is overly large, causing the algorithm to react too aggressively to previously observed losses. This instability can be exploited by a simple alternating loss construction.

Assume that T is divisible by 4 and define the losses

$$\ell_t(x) := (-1)^t\langle x, e_1\rangle.$$

When t is odd, the FTRL update satisfies

$$x_t = \arg\min_{x\in\mathcal{B}_p} \psi(x) = 0.$$

When t is even, we have

$$x_t = \arg\min_{x\in\mathcal{B}_p} \{\psi(x) - \eta\langle x, e_1\rangle\}$$

$$\implies \psi(x_t) - \eta\langle x_t, e_1\rangle \leqslant \psi\left(\tfrac{1}{2}e_1\right) - \tfrac{\eta}{2} = 1 - \tfrac{\eta}{2}.$$

Rearranging yields

$$\ell_t(x_t) = \langle x_t, e_1\rangle \geqslant \frac{1}{2} - \frac{1}{\eta} \geqslant \frac{1}{4},$$

where the final inequality uses $\eta > 4$. Consequently,

$$R_T = \sum_{t=1}^{T} \ell_t(x_t) \geqslant \frac{T}{8}.$$

Case 2: $\eta \leqslant 4$**.** In this regime, the step size is moderate, but the ambient dimension is sufficiently large that the regularisation imposed by ψ at the point w_d becomes prohibitively strong. As a result, FTRL cannot move quickly enough towards w_d within T rounds to achieve sublinear regret.

As in Section 4.5.3, define

$$u := \nabla\psi(w_d), \qquad q := p_\star,$$

and consider the linear losses

$$\ell_t(x) := -\frac{1}{\|u\|_q}\langle x, u\rangle, \qquad t = 1, \dots, T.$$

The corresponding FTRL updates are

$$x_{t+1} = \operatorname*{arg\,min}_{x \in \mathcal{B}_p} \left\{ \psi(x) - \frac{\eta t}{\|u\|_q}\langle x, u\rangle \right\}.$$

Case 2.1: $\frac{1}{\|\boldsymbol{u}\|_q}\langle \boldsymbol{x_{T+1}}, \boldsymbol{u}\rangle < \frac{3}{4}$**.** In this case, the algorithm fails to make sufficient progress in the direction of u, which directly implies linear regret:

$$\begin{aligned} R_T &= T - \sum_{t=1}^{T} \ell_t(x_t) \\ &\geqslant T\left(1 - \frac{1}{\|u\|_q}\langle x_{T+1}, u\rangle\right) \geqslant \frac{T}{4}. \end{aligned}$$

Case 2.2: $\frac{1}{\|\boldsymbol{u}\|_q}\langle \boldsymbol{x_{T+1}}, \boldsymbol{u}\rangle \geqslant \frac{3}{4}$**.** Following the same argument as in Section 4.5.3, we obtain

$$\|u\|_q \leqslant \eta T.$$

As in the proof of Proposition 4.5.1, we may therefore instead consider losses of the form $\ell_t(x) = \langle g_t, x\rangle$, where

$$g_t := -\frac{1}{\eta T} u \in \mathcal{B}_{p_\star},$$

which ensures that $x_{T+1} = w_d$. By the optimality of w_d, we have

$$\begin{aligned} w_d &= \operatorname*{arg\,min}_{x \in \mathcal{B}_p} \{\psi(x) - \langle u, x\rangle\} \\ &\implies \psi(w_d) - \langle u, w_d\rangle \leqslant 0. \end{aligned}$$

Using the condition on $\psi(w_d)$, Hölder's inequality and $\eta \leqslant 4$, we obtain

$$\begin{aligned} \lambda^2 \left(\log\log d - \log\log d_0\right) \leqslant \psi(w_d) &\leqslant \langle u, w_d\rangle \\ &\leqslant \|u\|_q \|w_d\|_p \leqslant \frac{\eta T \lambda}{\left(\varepsilon \log^\varepsilon 3\right)^{1/p}} \leqslant \frac{4T\lambda}{\left(\varepsilon \log^\varepsilon 3\right)^{1/p}}. \end{aligned}$$

Rearranging yields

$$\log\log d \leqslant \log\log d_0 + \frac{4T}{\lambda \left(\varepsilon \log^\varepsilon 3\right)^{1/p}}.$$

This leads to a contradiction whenever

$$d > \exp\left(\exp\left(\log\log d_0 + \frac{4T}{\lambda \left(\varepsilon \log^\varepsilon 3\right)^{1/p}}\right)\right),$$

with $\lambda = \frac{1}{2}\left(\varepsilon \log^\varepsilon 3\right)^{1/p}$. Therefore, Case 2.2 cannot occur, and we must be in Case 2.1, which implies

$$R_T \geqslant \frac{T}{4}.$$

4.6 Online Mirror Descent

All the results we present for FTRL also hold for its closely related cousin Online Mirror Descent (OMD, Section 2.3).

For general regret upper bounds, this can be obtained for OMD using a uniformly convex regulariser can be found in Sridharan and Tewari (2010) (Proposition 9). This can be generalised to a result with decreasing step-size akin to Theorem 6.10 in Orabona (2019). This yields anytime guarantees provided an upper bound on $D_{\max}$, the maximum Bregman divergence between the

competitor and the OMD iterates, is available. Alternatively, anytime bounds depending on the Bregman divergence to the initial point x_1 can be obtained via standard doubling-trick arguments (see, e.g., Lattimore and Szepesvári (2020)).

OMD on ℓ_p-balls is optimal in the low and high dimensional settings respectively using the same corresponding regularisers as for FTRL that we discuss in Section 4.2. To obtain anytime optimal bounds (as in Theorem 4.3.1), we can use a similar procedure as in Section 4.3, where we switch regulariser when a certain threshold is reached.

The results from Section 4.4 also generalise relatively straightforwardly to OMD. In particular, many of the steps of the proof of our main result Theorem 4.4.4 in Section 4.5 hold because FTRL and OMD with constant step-size are equivalent as long as the iterates stay within the interior of the space. Only some minor straightforward modifications are needed to handle cases where the step-size is large enough and iterates are on the boundary on the considered loss constructions. We also note that our results hold for natural gradient descent along dual manifolds due to the equivalence with OMD (Raskutti and Mukherjee, 2015).

4.7 Bandit feedback

In this section, we consider OCO on ℓ_p-balls with bandit feedback and linear losses. In the bandit feedback environment, the learner only observes the loss evaluated at the point played $\ell_t(x_t)$ instead of the full loss function ℓ_t. Similarly to the full information setting studied above, the optimal regret with bandit feedback also depends on the dimension regime. However, our main result in this section will show that for bandit feedback, sub-linear regret is not possible in the high dimensional regime, establishing a stronger separation between low and high dimensional settings.

The linear bandit problem has been extensively studied (see e.g. Lattimore and Szepesvári (2020)). A $\widetilde{O}(d^{1/2}T^{1/2})$ pseudo-regret[3] bound was established by Bubeck et al. (2018) for ℓ_p-balls ($p \in (1, 2]$) and more generally for strongly-convex action sets by Kerdreux et al. (2021c). For $p = 1$, the same bound can be achieved via a reduction to the multi-armed bandit problem (see Bubeck et al. (2018)). Kerdreux et al. (2021c) also established $\widetilde{O}(d^{1/p}T^{1-1/p})$ pseudo-regret

[3]Pseudo-regret is defined as $\bar{R}_T = \mathbb{E}\big[\sum_{t=1}^T \ell_t(x_t)\big] - \min_{x\in V} \mathbb{E}\big[\sum_{t=1}^T \ell_t(x)\big]$ where the expectation is with respect to the randomness in the learner's actions.

bounds for uniformly-convex sets of degree p, which apply to ℓ_p-balls with $p > 2$. However, since these regret guarantees are dimension-dependent they become vacuous in high-dimensions. The following result shows that with bandit feedback sub-linear regret bounds on ℓ_p-balls are unachievable in high-dimensions.

Theorem 4.7.1. *Fix T, $\delta > 0$ and $p \geqslant 1$. For any dimension d sufficiently large and any OCO algorithm with bandit feedback on $V = \mathcal{B}_p$, there exists a sequence of random linear losses $(\ell_t)_{t\in[T]}$ with gradients $(g_t)_{t\in[T]}$ such that $\|g_t\|_{p_\star} \leqslant L$ for all rounds t with probability at least $1-\delta$ and $\mathbb{E}\big[\bar{R}_T\big] \geqslant \frac{LT}{80}$, where the expectation is with respect to the randomness of the losses.*

The proof can be found in Section 4.10 and is based on information-theoretic arguments. We note that lower bounds in Chapter 24 of Lattimore and Szepesvári (2020) give linear regret for high-dimensional stochastic linear bandits with $p = 1, 2$. However, as discussed in their Chapter 29, the noise in the stochastic case is outside the inner-product so these bounds do not apply for adversarial linear bandits. We also remark that dimension-dependent bandit lower bounds from prior work such as the one in (Bubeck et al., 2018) only hold in the low-dimensional setting (e.g. in (Bubeck et al., 2018), their lower bound Theorem 4 only holds for $T \geqslant d^{2/(1-q/2)}$ and does not give linear regret in high dimension).

4.8 Related works

High-dimensional Online Learning: The setting where $d > T$ has been considered mostly for the stochastic linear bandit problem (Hao et al., 2020; Li et al., 2022; Chakraborty et al., 2023; Li et al., 2021; Komiyama and Imaizumi, 2023), where only the loss $\ell_t(x_t)$ at the points played by the learner are observed. Stochastic and linear refers to the losses being fixed and linear but observed with i.i.d. noise as opposed to the harder fully adversarial nature of our setting. Beyond stochastic linear bandits, little attention has been given to the high-dimensional setting. Although the high-dimensional setting was not explicitly studied in Levy and Duchi (2019), the results provided for OCO on ℓ_p-balls for $p \in [1, 2]$ fully characterise regret optimality across all dimensions (see Remark 4.0.1). Similarly, the results we present for the high-dimensional case of ℓ_p-balls with $p > 2$ in Section 4.2.2 follow from prior work not explicitly studying the high-dimensional setting (Sridharan and Tewari, 2010; Bhaskara et al., 2020).

Adaptive or anytime algorithms: A central objective in online learning is the development of algorithms that operate under minimal assumptions regarding problem-dependent quantities.

To this end, significant research has focused on parameter-free algorithms for unbounded domains that do not require prior knowledge of the competitor's location in advance (see Section 10 in (Orabona, 2019)). Related efforts have produced Lipschitz-adaptive algorithms that adjust to the scale of loss gradients (Orabona and Pál, 2018; Tim van Erven, 2021), as well as methods designed to adapt to the underlying nature of the loss sequence, such as best-of-both-world algorithms for both adversarial and stochastic environments (Bubeck and Slivkins, 2012; Luo and Schapire, 2015). Our work contributes to the pursuit of anytime bounds that hold across all time horizons or sample sizes without prior knowledge of T (Luo and Schapire, 2014; Degenne and Perchet, 2016; Harvey et al., 2023; Orabona and Jun, 2024). While the doubling trick (Cesa-Bianchi et al., 1997a) is a standard technique for achieving anytime guarantees, it can come at the detriment of adaptivity to other problem parameters (Degenne and Perchet, 2016) and is generally undesirable due to algorithmic complexity (Luo and Schapire, 2014; Harvey et al., 2023).

Uniform Convexity of functions (Definition 4.1.6, see also (Zălinescu, 1983; Azé and Penot, 1995; Nesterov, 2018; Kerdreux et al., 2021b)) is the key ingredient to obtain dimension-independent regret bounds in high-dimensions for OCO on ℓ_p-balls. Uniformly-convex functions have been considered as regularisers for offline (d'Aspremont et al., 2018) and online optimisation (Sridharan and Tewari, 2010; Bhaskara et al., 2020), and also as objectives (Iouditski and Nesterov, 2014) and losses (Sridharan, 2012). **Uniform convexity of sets**[4] (Clarkson, 1936; Hanner, 1956; Ball et al., 1994) allows for interpolation of the set curvature between strong convexity and absence of curvature. In optimisation, curvature of the action set such as strong convexity typically leads to accelerated convergence rates (Huang et al., 2017; Molinaro, 2023; Garber and Hazan, 2015; Mhammedi, 2022). Uniformly-convex sets then allow interpolation between the faster rates of strongly-convex sets and the slower rates of sets without curvature (Dekel et al., 2017; Kerdreux et al., 2021a,c; Tsuchiya and Ito, 2024) (see Kerdreux et al. (2021b) for an overview). In this chapter, we consider a natural class of uniformly-convex sets, ℓ_p-balls for $p > 2$. These balls also interpolate between strongly-convexity ($p = 2$) and absence of curvature ($p = \infty$), and in high dimensions we recover a connection between curvature and faster rates ($T^{1-1/p}$), which is absent in low-dimensions if we consider the dimension as fixed where the rate is $O(\sqrt{T})$ for all values of $p \geqslant 2$.

[4]Uniform convexity for sets and uniform convexity for functions are connected through an equivalence of uniform convexity between a set and the set-induced norm (Kerdreux et al., 2021b).

4.9 Conclusion

In this chapter, we studied OCO on ℓ_p-balls in $\mathbb{R}^d$ for $p > 2$, distinguishing between high-dimensional ($d > T$) and low-dimensional ($d \leqslant T$) regimes. In high-dimensions, FTRL achieves the optimal regret of $O(T^{1-1/p})$ with a uniformly-convex regulariser, while in low-dimensions it achieves the optimal regret of $O(T^{1/2}d^{1/2-1/p})$ with a strongly-convex regulariser. Importantly, we proved neither regulariser is optimal across both regimes. Therefore, when the dimension regime is unknown, we showed that FTRL with adaptive regularisation is anytime optimal. Furthermore, we established that this adaptivity is necessary to achieve anytime optimality for sign-invariant regularisers, highlighting the non-universality of FTRL with fixed regularisation for general OCO problems. Overall, *our results establish a fundamental separation between low and high dimensional regimes, both in the optimal regret and in the gradient-based methods that achieve it.* This separation is amplified in the linear bandit problem, where we ruled out the possibility for sub-linear regret bounds in high-dimension. These results underscore the role of dimension and geometry in achieving optimal performance in OCO.

4.10 Missing Details - Proofs of Section 4.7

Throughout this section, we will use the notation R_T for the pseudo-regret. In fact since a randomized learner is equivalent to a random choice of deterministic learners, we will consider in the proofs below deterministic learners and the regret is equal to the pseudo-regret. In addition, since our loss constructions are oblivious to the learner's actions, even for randomised learners, the pseudo-regret is equal to the regret.

We split the proof into the case where $p > 4/3$ is "large" and the case where $p \in [1, 4/3]$ is "small" and consider separate loss constructions for each. We first highlight the intuition of the loss constructions.

- **For $p > 4/3$**, we take inspiration from the loss construction which Bubeck et al. (2018) use to prove a $\Omega(d\sqrt{T})$ lower bound for low-dimensional ($d^2 < T$) ℓ_p-balls with $p > 2$. The construction consists of linear Gaussian losses where the mean of each coordinate is the same distance from 0 but the learner does not know the sign. When the dimension is large enough, the learner does not acquire enough information to determine the signs of these means in the T rounds to get sub-linear regret. **This construction will not work when**

$p \leqslant 4/3$ because when p is close to 1, the lack of a distinct corner in the ℓ_p-ball allows any point on the boundary (including $\pm e_1$) with correct signs to achieve similar loss to the competitor (a corner). The learner can therefore focus on $\pm e_1$ simplifying the problem to one-dimension where sub-linear regret is achievable.

- **For $p \leqslant 4/3$**, the construction consists of linear Gaussian losses where the mean vector has a single non-zero positive entry, unknown to the learner. When the dimension is large enough, it does not acquire enough information to determine the non-zero coordinate of the mean in the T rounds to get sub-linear regret. **This construction will not work when $p > 4/3$** because for $p \gg 1$ the learner can exploit the ℓ_p-ball's proximity to the hypercube by playing points with all coordinates close to -1, bypassing the need to identify the correct non-zero mean coordinate.

We present the proofs for a Lipschitz constant $L = 1$; the results extend to arbitrary $L > 0$ by linear rescaling.

4.10.1 Case $p > 4/3$

Theorem 4.10.1. *Fix T and $\delta > 0$. Consider $p > 4/3$ and*

$$d > \max\Big\{16T, \frac{1}{c_1}\log\frac{C_1 T}{\delta}, \Big(\frac{1}{c_1 p_\star}\log\frac{C_1 T}{\delta}\Big)^{p_\star/2}, e^2\Big\},$$

for some universal constants c_1, C_1. For any OCO algorithm with bandit feedback on $V = \mathcal{B}_p$, there exists a sequence of random linear losses $(\ell_t)_{t\in[T]}$ with gradients $(g_t)_{t\in[T]}$ such that $\|g_t\|_{p_\star} \leqslant 1$ for all rounds t with probability at least $1 - \delta$ and

$$\mathbb{E}\big[R_T\big] \geqslant \frac{T}{80},$$

where the expectation is with respect to the randomness of the losses.

Proof. The following loss construction and analysis is inspired from the proof of Theorem 4 of Bubeck et al. (2018). Their construction is designed for the low-dimensional setting in such a way that the learner has to balance exploration and exploitation rounds. We only consider the losses corresponding to exploration rounds and generalize the analysis to the high-dimensional setting.

Let $\varepsilon > 0$ be such that $\varepsilon^{p_\star} = 1/d$. Let $T < \alpha d$ (with $\alpha = 1/16$). For a fixed $\xi \in \{-1, 1\}^d$, define the losses as $\ell_t(x) = x^T g_t^\xi$ where $g_t^\xi \sim \mathcal{N}(\varepsilon\xi, \frac{1}{d^{2/p_\star}} I_d)$ (i.i.d.). We show that (when ξ is sampled uniformly at the start and fixed throughout the rounds)

$$\mathbb{E}_\xi \mathbb{E}_{g_t^\xi}\big[R_T\big] \geqslant \frac{T}{16}.$$

We use $\mathbb{E}_\xi$ for the expectation with respect to ξ and $\mathbb{E}_{g_t^\xi}$ for the expectation with respect to g_t^ξ with ξ fixed. We will also use $x_{t,i}$ to mean the i-th coordinate of x_t.

For fixed ξ: $\mathbb{E}_{g_t^\xi}\big[\ell_t(x)\big] = \mathbb{E}\big[x^T g_t^\xi\big] = \varepsilon \cdot x^T \xi$. So the competitor $x^\star = \arg\min_{x \in V} \varepsilon \cdot \xi^T x = -d^{-1/p}\xi$. Let us define $\bar{x} = \frac{1}{T}\sum_{t=1}^T \mathbb{E}[x_t]$. In particular one has

$$\mathbb{E}_{g_t^\xi}\big[R_T\big] = \varepsilon T \cdot \xi^T(\bar{x} - x^\star).$$

The following lemma expresses the expected regret in terms of the expected number of rounds and coordinates for which the learner plays on the wrong side of ξ. The proof is in Section 4.10.1.

Lemma 4.10.2 (Generalization of Lemma 6 of Bubeck et al. (2018))**.**

$$\mathbb{E}_{g_t^\xi}\big[R_T\big] \geqslant \frac{\varepsilon^{p_\star}}{p_\star} \cdot \mathbb{E}_{g_t^\xi}\Big[\sum_{t=1}^T \sum_{i=1}^d \mathbb{I}\big\{x_{t,i}\xi_i \geqslant 0\big\}\Big].$$

And now the next lemma shows that the expected number of rounds and coordinates for which the learner plays on the wrong side of ξ is linear in both T and d. The proof is in Section 4.10.1.

Lemma 4.10.3. *With $T < \alpha d = \frac{1}{16}$, we have $\mathbb{E}_\xi \mathbb{E}_{g_t^\xi} \sum_{t=1}^T \sum_{i=1}^d \mathbb{I}\big\{x_{t,i}\xi_i \geqslant 0\big\} \geqslant \frac{dT}{4}$.*

Combining both lemmas, we have

$$\mathbb{E}\big[R_T\big] \geqslant \frac{1}{p_\star}\varepsilon^{p_\star} \cdot \mathbb{E}\Big[\sum_{t=1}^T \sum_{i=1}^d \mathbb{I}\big\{x_{t,i}\xi_i \geqslant 0\big\}\Big] \geqslant \frac{1}{4p_\star}\varepsilon^{p_\star} T d = \frac{1}{4p_\star} T \geqslant \frac{T}{16},$$

since $p > 4/3$ so $p_\star \leqslant 4$. Now to ensure the Lipschitz-condition with high-probability, we get an extra factor of $1/5$ (see the next section), concluding the proof. □

Bound on gradients

Recall that $p > 4/3$ so $p_\star \leqslant 4$. Fix $\xi \in \{-1,1\}^d$. $g_t \sim \mathcal{N}(\varepsilon\xi, d^{-2/p_\star} I_d)$. So $g_t = d^{-1/p_\star} X + \varepsilon\xi$ where $X \sim \mathcal{N}(0, I_d)$. From Winkelbauer (2012), we have

$$\mathbb{E}\big[\|X\|_{p_\star}\big] \leqslant \Big(\mathbb{E}\Big[\sum_{i=1}^d |X_i|^{p_\star}\Big]\Big)^{1/p_\star} = \Big(\mathbb{E}\Big[\sum_{i=1}^d 2^{p_\star/2}\frac{\Gamma((p_\star+1)/2)}{\sqrt{\pi}}\Big]\Big)^{1/p_\star} \leqslant \sqrt{2}\Big(\frac{3}{4}\Big)^{1/p_\star} d^{1/p_\star} \leqslant 2d^{1/p_\star}.$$

$$\implies \mathbb{E}\big[\|g_t\|_{p_\star}\big] \leqslant 2 + \varepsilon d^{1/p_\star} = 2 + 1 = 3.$$

Fix $\delta > 0$.

- **For $p_\star \leqslant 2$:** By Theorem 1.1 in Paouris et al. (2017) for some constants $C_1, c_1 > 0$,

$$\mathbb{P}\Big(\|X\|_{p_\star} \leqslant (1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\Big) \geqslant 1 - C_1\exp(-c_1\beta^2 d).$$

  Assuming $d \geqslant \frac{1}{c_1}\log\frac{C_1 T}{\delta}$, we have $\beta = \sqrt{\frac{1}{c_1 d}\log\frac{C_1 T}{\delta}} \leqslant 1$ and

$$\mathbb{P}\Big(\|X\|_{p_\star} \leqslant (1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\Big) \geqslant 1 - \frac{\delta}{T}.$$

- **For $2 < p_\star \leqslant 4$:** By Theorem 1.1 in Paouris et al. (2017) for some constants $C_1, c_1 > 0$,

$$\mathbb{P}\Big(\|X\|_{p_\star} \leqslant (1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\Big) \geqslant 1 - C_1\exp(-c_1\beta p_\star d^{2/p_\star}).$$

  Assuming $d \geqslant \Big(\frac{1}{c_1 p_\star}\log\frac{C_1 T}{\delta}\Big)^{p_\star/2}$, we have $\beta = \frac{1}{c_1 q d^{2/p_\star}}\log\frac{C_1 T}{\delta} \leqslant 1$ and

$$\mathbb{P}\Big(\|X\|_{p_\star} \leqslant (1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\Big) \geqslant 1 - \frac{\delta}{T}.$$

In both cases, with probability at least $1 - \delta/T$

$$\|X\|_{p_\star} \leqslant (1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big] \leqslant 2\mathbb{E}\big[\|X\|_{p_\star}\big] \leqslant 4d^{1/p_\star},$$
$$\implies \|g_t\|_{p_\star} \leqslant d^{-1/p_\star}\|X\|_{p_\star} + \varepsilon d^{1/p_\star} \leqslant 4 + 1 = 5.$$

By a union bound over all rounds, with probability $1-\delta$, $\|g_t\|_{p_\star} \leqslant 5$ for all rounds t. So rescaling the losses by a factor of 5 gives gradients whose $\ell_{p_\star}$-norm is bounded by 1 with high-probability

and a regret bound of:

$$\mathbb{E}\big[R_T\big] \geqslant \frac{T}{80}.$$

Proof of Lemma 4.10.2

Let $W_t = \Big\{i \in [d] : x_{t,i}\xi_i < 0\Big\}$ and $S = \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{d} \mathbb{I}\big\{x_{t,i}\xi_i \geqslant 0\big\}\Big]$. We have that

$$\begin{aligned}
\mathbb{E}_{g_t^\xi}\big[R_T\big] &= \varepsilon T \cdot \xi^T(\bar{x} - x^\star) \\
&= \varepsilon \sum_{t=1}^{T} \mathbb{E}_{g_t^\xi}\Big[\sum_{i \notin W_t} \xi_i x_{t,i}\Big] + \varepsilon \sum_{t=1}^{T} \mathbb{E}_{g_t^\xi}\Big[\sum_{i \in W_t} \xi_i x_{t,i}\Big] + \varepsilon T d^{1/p_\star} \\
&\geqslant \varepsilon \sum_{t=1}^{T} \mathbb{E}_{g_t^\xi}\Big[\sum_{i \in W_t} \xi_i x_{t,i}\Big] + \varepsilon T d^{1/p_\star}
\end{aligned}$$

Therefore, it is sufficient to show that

$$\varepsilon \sum_{t=1}^{T} \mathbb{E}_{g_t^\xi}\Big[\sum_{i \in W_t} \xi_i x_{t,i}\Big] + \varepsilon T d^{1/p_\star} \geqslant \frac{\varepsilon^{p_\star} S}{p_\star}.$$

Since $\|x_{t,W_t}\|_p \leqslant 1$ (we use x_{t,W_t} to mean that the coordinates of x_t that are not in W_t are 0), by Holder's inequality we know that

$$\varepsilon \sum_{i \in W_t} \xi_i x_{t,i} = (x_{t,W_t})^T(\varepsilon \xi_{W_t}) \geqslant -\|x_{t,W_t}\|_p \| - \varepsilon \xi_{W_t}\|_{p_\star} \geqslant -|W_t|^{1/p_\star}\varepsilon.$$

Noting that (see (4.11) below)

$$|W_t|^{1/p_\star}\varepsilon = ((d - |W_t^C|)\varepsilon^{p_\star})^{1/p_\star} \leqslant (d\varepsilon^{p_\star})^{1/p_\star} - \frac{1}{p_\star}\varepsilon^{p_\star}|W_t^C|, \tag{4.10}$$

we have

$$\begin{aligned}
&\varepsilon \sum_{i \in W_t} \xi_i x_{t,i} \geqslant \frac{1}{p_\star}\varepsilon^{p_\star}|W_t^C| - (d\varepsilon^{p_\star})^{1/p_\star} = \frac{1}{p_\star}\varepsilon^{p_\star} \sum_{i=1}^{d} \mathbb{I}\big\{x_{t,i}\xi_i \geqslant 0\big\} - (d\varepsilon^{p_\star})^{1/p_\star} \\
\implies &\varepsilon \sum_{t=1}^{T} \mathbb{E}_{g_t^\xi}\Big[\sum_{i \in W_t} \xi_i x_{t,i}\Big] \geqslant \frac{\varepsilon^{p_\star}}{p_\star}\mathbb{E}_{g_t^\xi}\Big[\sum_{t=1}^{T}\sum_{i=1}^{d} \mathbb{I}\big\{x_{t,i}\xi_i \geqslant 0\big\}\Big] - T(d\varepsilon^{p_\star})^{1/p_\star} = \frac{\varepsilon^{p_\star} S}{p_\star} - \varepsilon T d^{1/p_\star},
\end{aligned}$$

which concludes the proof.

Proof of (4.10): Since $x^{1/p_\star}$ is concave: for all $x, y \in \mathbb{R}$, $x^{1/p_\star} \leqslant y^{1/p_\star} + \frac{1}{p_\star}y^{-1/p}(x - y)$. In

particular, with $x = \varepsilon^{p_\star}(d-s)$, $y = \varepsilon^{p_\star} d$, we have

$$\varepsilon(d-s)^{1/p_\star} \leqslant \varepsilon d^{1/p_\star} - \frac{1}{p_\star}\frac{\varepsilon^{p_\star} s}{(\varepsilon^{p_\star} d)^{1/p}} = \varepsilon d^{1/p_\star} - \frac{1}{p_\star}\varepsilon^{p_\star} s, \tag{4.11}$$

since $\varepsilon^{p_\star} d = 1$. Using $s = |W_t^C|$ gives the result.

Proof of Lemma 4.10.3

At round t conditioned on ξ, the observed feedback is

$$f_t^\xi := x_t^T g_t^\xi \sim \mathcal{N}(\varepsilon \cdot x_t^T \xi, \sigma_t^2), \text{ where } \sigma_t^2 = \frac{\|x_t\|_2^2}{d^{2/p_\star}}.$$

Denote p_ξ for the law of the observed feedback up to time T conditioned on ξ, i.e., the law of $(f_1^\xi, ..., f_T^\xi)$. Consider ξ and ξ' differing only in coordinate $i \in d$. By Pinsker's inequality we have

$$d_{\mathrm{TV}}(p_\xi, p_{\xi'}) \leqslant \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_\xi, p_{\xi'})}.$$

By the chain rule for the KL divergence / operations on conditional densities:

$$\begin{aligned} D_{\mathrm{KL}}(p_\xi, p_{\xi'}) &= \mathbb{E}_{(f_1,...,f_T)\sim p_\xi}\Big[\log \frac{p_\xi\big(f_1, ..., f_T\big)}{p_{\xi'}\big(f_1, ..., f_T\big)}\Big] \\ &= \sum_{t=1}^T \mathbb{E}_{(f_1,...,f_t)\sim p_\xi}\Big[\log \frac{p_\xi\big(f_t|f_{t-1}..., f_1\big)}{p_{\xi'}\big(f_t|f_{t-1}..., f_1\big)}\Big] \\ &= \sum_{t=1}^T \mathbb{E}_{(f_1,...,f_{t-1})\sim p_\xi}\Big\{\mathbb{E}_{f_t\sim p_\xi(\cdot|f_1,...,f_{t-1})}\Big[\log \frac{p_\xi\big(f_t|f_{t-1}..., f_1\big)}{p_{\xi'}\big(f_t|f_{t-1}..., f_1\big)}\Big]\Big\}. \end{aligned}$$

Now since x_t is a deterministic function of $f_1, ..., f_{t-1}$ and given x_t, $f_t \sim \mathcal{N}(\varepsilon x_t^T \xi, \sigma_t^2)$ under p_ξ, we have that the inner expectation is a KL divergence between Gaussians:

$$\begin{aligned} &\mathbb{E}_{f_t\sim p_\xi(\cdot|f_1,...,f_{t-1})}\Big[\log \frac{p_\xi\big(f_t|f_{t-1}..., f_1\big)}{p_{\xi'}\big(f_t|f_{t-1}..., f_1\big)}\Big] = \frac{\big(\varepsilon x_t^T \xi - \varepsilon x_t^T \xi'\big)^2}{2\sigma_t^2} = \frac{4\varepsilon^2 x_{t,i}^2}{2\sigma_t^2} = \frac{2\varepsilon^2 x_{t,i}^2}{\sigma_t^2} \\ \implies & D_{\mathrm{KL}}(p_\xi, p_{\xi'}) = 2\sum_{t=1}^T \mathbb{E}_{(f_1,...,f_{t-1})\sim p_\xi}\Big[\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}\Big] = 2\sum_{t=1}^T \mathbb{E}_{p_\xi}\Big[\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}\Big] \\ \implies & d_{\mathrm{TV}}(p_\xi, p_{\xi'}) \leqslant \sqrt{\sum_{t=1}^T \mathbb{E}_{p_\xi}\Big[\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}\Big]}. \end{aligned}$$

Now, using ξ_{-i} to refer to all the coordinates of ξ except the i-th and ξ_{i+} (resp. $\xi_{i,-}$) to denote

that the i-th coordinate of ξ is $+1$ (resp. -1),

$$\begin{aligned}
\mathbb{E}_{\xi}\Big[\mathbb{E}_{p_\xi}\big[\sum_{t=1}^{T}\mathbb{I}\{x_{t,i}\xi_i \geqslant 0\}\big]\Big] &= \mathbb{E}_{\xi_{-i}}\mathbb{E}_{\xi_i}\Big[\mathbb{E}_{p_\xi}\big[\sum_{t=1}^{T}\mathbb{I}\{x_{t,i}\xi_i \geqslant 0\}\big]|\xi_{-i}\Big] \\
&= \frac{1}{2}\mathbb{E}_{\xi_{-i}}\Big[\mathbb{E}_{p_{\xi_{i+}}}\big[\sum_{t=1}^{T}\mathbb{I}\{x_{t,i}\cdot 1 \geqslant 0\}\big] + \mathbb{E}_{p_{\xi_{i-}}}\big[\sum_{t=1}^{T}\mathbb{I}\{x_{t,i}\cdot(-1) \geqslant 0\}\big]\Big] \\
&= \frac{1}{2}\mathbb{E}_{\xi_{-i}}\Big[\mathbb{E}_{p_{\xi_{i+}}}\big[\sum_{t=1}^{T}\mathbb{I}\{x_{t,i} \geqslant 0\}\big] + \mathbb{E}_{p_{\xi_{i-}}}\big[\sum_{t=1}^{T}1-\mathbb{I}\{x_{t,i} > 0\}\big]\Big] \\
&\geqslant \frac{T}{2} + \frac{1}{2}\sum_{t=1}^{T}\mathbb{E}_{\xi_{-i}}\Big[\mathbb{E}_{p_{\xi_{i+}}}\big[\mathbb{I}\{x_{t,i} \geqslant 0\}\big] - \mathbb{E}_{p_{\xi_{i-}}}\big[\mathbb{I}\{x_{t,i} \geqslant 0\}\big]\Big] \\
&\geqslant \frac{T}{2} - \frac{1}{4}\sum_{t=1}^{T}\mathbb{E}_{\xi_{-i}}\Big[d_{\mathrm{TV}}(p_{\xi_{i+}}, p_{\xi_{i-}}) + d_{\mathrm{TV}}(p_{\xi_{i-}}, p_{\xi_{i+}})\Big] \quad \text{using Pinsker's inequality} \\
&= \frac{T}{2} - \frac{T}{4}\mathbb{E}_{\xi_{-i}}\Big[d_{\mathrm{TV}}(p_{\xi_{i+}}, p_{\xi_{i-}}) + d_{\mathrm{TV}}(p_{\xi_{i-}}, p_{\xi_{i+}})\Big] \\
&\geqslant \frac{T}{2} - \frac{T}{4}\mathbb{E}_{\xi_{-i}}\Big[\sqrt{\sum_{t=1}^{T}\mathbb{E}_{g_t^{\xi_{i+}}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}} + \sqrt{\sum_{t=1}^{T}\mathbb{E}_{g_t^{\xi_{i-}}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}}\Big] \\
&= \frac{T}{2} - \frac{T}{2}\mathbb{E}_{\xi}\Big[\sqrt{\sum_{t=1}^{T}\mathbb{E}_{g_t^{\xi}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}}\Big].
\end{aligned}$$

Summing over all possible coordinates i, we get:

$$\frac{1}{T}\sum_{i=1}^{d}\mathbb{E}_{\xi}\Big[\mathbb{E}_{p_\xi}\big[\sum_{t=1}^{T}\mathbb{I}\{x_{t,i}\xi_i \geqslant 0\}\big]\Big] \geqslant \frac{d}{2} - \frac{1}{2}\sum_{i=1}^{d}\mathbb{E}_{\xi}\Big[\sqrt{\sum_{t=1}^{T}\mathbb{E}_{g_t^{\xi}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}}\Big].$$

Note that due to the concavity of the square-root:

$$\begin{aligned}
\sum_{i=1}^{d}\mathbb{E}_{\xi}\Big[\sqrt{\sum_{t=1}^{T}\mathbb{E}_{g_t^{\xi}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}}\Big] &\leqslant \sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}\mathbb{E}_{\xi,g_t^{\xi}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}} \\
&= d\frac{1}{d}\sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T}\mathbb{E}_{\xi,g_t^{\xi}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}} \\
&\leqslant d\sqrt{\frac{1}{d}\sum_{i=1}^{d}\sum_{t=1}^{T}\mathbb{E}_{\xi,g_t^{\xi}}\frac{\varepsilon^2 x_{t,i}^2}{\sigma_t^2}} \\
&= \sqrt{d\sum_{t=1}^{T}\mathbb{E}_{\xi,g_t^{\xi}}\frac{\varepsilon^2\|x_t\|_2^2}{\sigma_t^2}}.
\end{aligned}$$

So we get

$$\begin{aligned}\frac{1}{T}\mathbb{E}_{\xi}\mathbb{E}_{g_i^{\xi}}\sum_{t=1}^{T}\sum_{i=1}^{d}\mathbb{I}\{x_{t,i}\xi_i \geqslant 0\} &\geqslant \frac{d}{2} - \sqrt{d\sum_{t=1}^{T}\mathbb{E}_{\xi,g_i^{\xi}}\frac{\varepsilon^2\|x_t\|_2^2}{\sigma_t^2}} \\ &= \frac{d}{2} - \sqrt{d^{1+2/p_\star}T\varepsilon^2} \\ &= \frac{d}{2} - \sqrt{dT} \\ &\geqslant d\Big(\frac{1}{2} - \sqrt{\alpha}\Big) \\ &= \frac{d}{4}\end{aligned}$$

since $\varepsilon = (1/d)^{1/p_\star}$ and $T < \alpha d = d/16$. This gives the result.

4.10.2 Case $1 < p \leqslant 4/3$

Theorem 4.10.4. *Fix T and $\delta > 0$. Consider $p \in (1, 4/3]$ and*

$$d > \max\Big\{(128p_\star T)^2, \Big(\frac{1}{c_1 p_\star}\log\frac{C_1 T}{\delta}\Big)^{p_\star/2}, e^2\Big\},$$

for some universal constants c_1, C_1. For any OCO algorithm with bandit feedback on $V = \mathcal{B}_p$, there exists a sequence of random linear losses $(\ell_t)_{t\in[T]}$ with gradients $(g_t)_{t\in[T]}$ such that $\|g_t\|_{p_\star} \leqslant 1$ for all rounds t with probability at least $1-\delta$ and

$$\mathbb{E}\big[R_T\big] \geqslant \frac{T}{16},$$

where the expectation is with respect to the randomness of the losses.

Proof. Before the start of the game, draw $Y \sim Unif(1, ..., d)$ and define the losses as $\ell_t(x) = x^T g_t^Y$ where

$$g_{t,i}^Y \sim \begin{cases}\mathcal{N}(0,\sigma^2), & \text{if } i \neq Y,\\ \mathcal{N}(1/2,\sigma^2), & \text{if } i = Y,\end{cases}$$

where $\sigma = (8\sqrt{p_\star}d^{1/p_\star})^{-1}$. We show that $\mathbb{E}_Y\mathbb{E}_{\ell_1,\ldots,\ell_T}R_T \geqslant T/16$.
Fix $\alpha \in [0,1]$ and define $A_i(\alpha) = \{t : y_{t,i} \geqslant -\alpha\}$. Then

$$\mathbb{E}\big[R_T|Y=i\big] = \frac{T}{2} + \frac{1}{2}\mathbb{E}\Big[\sum_{t=1}^{T} y_{t,i}|Y=i\Big]$$

$$
\begin{aligned}
&= \frac{T}{2} + \frac{1}{2}\mathbb{E}\Big[\sum_{t\notin A_i(\alpha)} y_{t,i} + \sum_{t\in A_i(\alpha)} y_{t,i} | Y = i\Big] \\
&\geqslant \frac{T}{2} + \frac{1}{2}\mathbb{E}\Big[\sum_{t\notin A_i(\alpha)} (-1) + \sum_{t\in A_i(\alpha)} (-\alpha) | Y = i\Big] \\
&= \frac{T}{2} - \frac{1}{2}\mathbb{E}\Big[T - |A_i(\alpha)| + \alpha|A_i(\alpha)||Y = i\Big] \\
&= (1-\alpha)\frac{1}{2}\mathbb{E}\Big[|A_i(\alpha)||Y = i\Big] \\
&= (1-\alpha)\frac{1}{2}\mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\} | Y = i\Big].
\end{aligned}
$$

The following lemma bounds the expected number of rounds where the learner suffers large regret (as measured by α) when $Y = i$ compared to an environment where all coordinates of g_t are 0-mean for all t (i.e. there is no better direction). We denote E_{p_0} expectations with respect to this environment. The proof is in Section 4.10.2.

Lemma 4.10.5. *Let $\sigma_t^2 = \|y_t\|_2^2 \cdot \sigma^2$, then*

$$
\mathbb{E}\big[\sum_{t=1}^{T} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\} | Y = i\big] \geqslant \mathbb{E}_{p_0}\big[\sum_{t=1}^{T} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\}\big] - T\sqrt{\sum_{t=1}^{T} \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]}
$$

From the lemma, we have

$$
\mathbb{E}\big[R_T | Y = i\big] \geqslant (1-\alpha)\frac{1}{2}\Big\{\mathbb{E}_{p_0}\big[\sum_{t=1}^{T} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\}\big] - T\sqrt{\sum_{t=1}^{T} \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]}\Big\}.
$$

Taking an expectation with respect to Y we have:

$$
\begin{aligned}
\mathbb{E}\big[R_T\big] &= \frac{1}{d}\sum_{i=1}^{d} \mathbb{E}\big[R_T | Y = i\big] \\
&\geqslant \frac{(1-\alpha)}{2d}\sum_{i=1}^{d}\Big\{\mathbb{E}_{p_0}\big[\sum_{t=1}^{T} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\}\big] - T\sqrt{\sum_{t=1}^{T} \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]}\Big\} \\
&= \frac{(1-\alpha)}{2d}\Big\{\mathbb{E}_{p_0}\big[\sum_{t=1}^{T}\sum_{i=1}^{d} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\}\big] - T\sum_{i=1}^{d}\sqrt{\sum_{t=1}^{T} \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]}\Big\}.
\end{aligned}
$$

For the first term: fix $\alpha = (d/2)^{-1/p}$ and note that if $\sum_{i=1}^{d} \mathbb{I}\big\{y_{t,i} \geqslant -\alpha\big\} < d/2$, then there are

more than $d/2$ coordinates of y_t whose value is less than $-\alpha$, this means

$$\|y_t\|_p^p > \frac{d}{2}\alpha^p = \frac{d}{2}\frac{2}{d} = 1,$$

which contradicts $y_t \in \mathcal{B}_p$ so we must have $\sum_{i=1}^d \mathbb{I}\{y_{t,i} \geqslant -\alpha\} \geqslant d/2$.

For the second term, using Jensen's inequality and the concavity of $\sqrt{x}$,

$$\frac{1}{d}\sum_{i=1}^d \sqrt{\sum_{t=1}^T \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]} \leqslant \sqrt{\frac{1}{d}\sum_{i=1}^d\sum_{t=1}^T \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]} = \sqrt{\frac{1}{d}\sum_{t=1}^T \mathbb{E}_{p_0}\Big[\frac{\|y_t\|_2^2}{8\sigma_t^2}\Big]} = \frac{1}{2\sigma}\sqrt{\frac{T}{2d}}.$$

Combining we have

$$\mathbb{E}\big[R_T\big] \geqslant \frac{1-\alpha}{2}\Big(\frac{T}{2} - T\frac{1}{2\sigma}\sqrt{\frac{T}{2d}}\Big).$$

- Since $d \geqslant e^2 \geqslant 2^{p+1}$, $1-\alpha = 1-(2/d)^{1/p} \geqslant 1-2^{-p/p} = 1/2$.
- If $d \geqslant (128p_\star T)^2$ then $d \geqslant 2T/\sigma^2$ and:

$$\frac{T}{2} - T\frac{1}{2\sigma}\sqrt{\frac{T}{2d}} \geqslant \frac{T}{2} - \frac{T}{4} = \frac{T}{2}.$$

The condition on d follows from the definition of σ and $p_\star \geqslant 4$:

$$d \geqslant (128p_\star T)^2 \geqslant (128qT)^{1/(1-2/p_\star)} \implies d^{1-2/p_\star} \geqslant 128qT \implies d \geqslant \frac{128p_\star T}{d^{2/p_\star}} = \frac{2T}{\sigma^2}.$$

We hence have $\mathbb{E}\big[R_T\big] \geqslant T/16$. The following section ensures that the Lipschitz-condition is satisfied with high-probability.

Bound on gradients

Recall that $p \leqslant 4/3$, so $p_\star \geqslant 4$. Given $Y = i$, $g_t^Y \sim \mathcal{N}(\frac{1}{2}e_i, \sigma^2 I_d)$. So $g_t = \sigma X + \frac{1}{2}e_i$ where $X \sim \mathcal{N}(0, I_d)$. From Winkelbauer (2012), we have

$$\mathbb{E}\big[\|X\|_{p_\star}\big] \leqslant \Big(\mathbb{E}\Big[\sum_{i=1}^d |X_i|^{p_\star}\Big]\Big)^{1/p_\star} = \Big(\mathbb{E}\Big[\sum_{i=1}^d 2^{p_\star/2}\frac{\Gamma((p_\star+1)/2)}{\sqrt{\pi}}\Big]\Big)^{1/p_\star}.$$

From Batir (2017) (Theorem 2.2), we have

$$\Gamma\Big(\frac{p_\star+1}{2}\Big)=\Gamma\Big(\frac{p_\star-1}{2}+1\Big)\leqslant\sqrt{2\pi}\Big(\frac{p_\star-1}{2}\Big)^{(p_\star-1)/2}\exp\Big(-\frac{p_\star-1}{2}\Big)\sqrt{2\frac{p_\star-1}{2}}\leqslant 2\sqrt{\pi}{p_\star}^{p_\star/2}e^{-(p_\star-1)/2}$$
$$\implies\Big(2^{p_\star/2}\frac{\Gamma((p_\star+1)/2)}{\sqrt{\pi}}\Big)^{1/p_\star}\leqslant 2\sqrt{p_\star}$$
$$\implies\mathbb{E}\big[\|X\|_{p_\star}\big]\leqslant 2\sqrt{p_\star}d^{1/p_\star}$$
$$\implies\mathbb{E}\big[\|g_t\|_{p_\star}\big]\leqslant 2\sqrt{p_\star}\sigma d^{1/p_\star}+\frac{1}{2}.$$

Fix $\delta>0$. By Theorem 1.1 in Paouris et al. (2017) for some constants $C_1,c_1>0$,

$$\mathbb{P}\Big(\|X\|_{p_\star}\leqslant(1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\Big)\geqslant 1-C_1\exp(-c_1\beta p_\star d^{2/p_\star}).$$

Assuming $d\geqslant\Big(\frac{1}{c_1p_\star}\log\frac{C_1T}{\delta}\Big)^{p_\star/2}$, we have $\beta=\frac{1}{c_1qd^{2/p_\star}}\log\frac{C_1T}{\delta}\leqslant 1$ and

$$\mathbb{P}\Big(\|X\|_{p_\star}\leqslant(1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\Big)\geqslant 1-\frac{\delta}{T}.$$

So with probability at least $1-\delta/T$

$$\|X\|_{p_\star}\leqslant(1+\beta)\mathbb{E}\big[\|X\|_{p_\star}\big]\leqslant 2\mathbb{E}\big[\|X\|_{p_\star}\big]\leqslant 4\sqrt{p_\star}d^{1/p_\star},$$
$$\implies\|g_t\|_{p_\star}\leqslant 4\sqrt{p_\star}\sigma d^{1/p_\star}+\frac{1}{2}\leqslant 1,$$

where the final inequality follows from $\sigma\leqslant\frac{1}{8\sqrt{p_\star}d^{1/p_\star}}$

By a union bound over all rounds, with probability $1-\delta$, $\|g_t\|_{p_\star}\leqslant 1$ for all rounds t.

Proof of Lemma 4.10.5

Given $Y=i$, the observed feedback at round t is exactly

$$f_t^Y:=y_t^Tg_t^Y\sim\mathcal{N}(\frac{1}{2}y_{t,i},\sigma_t^2),\ \text{where}\ \sigma_t^2=\|y_t\|_2^2\sigma^2.$$

Denote p_i for the law of the observed feedback up to time T given $Y=i$, i.e., the law of $(f_1^i,...,f_T^i)$. Denote p_0 (use $Y=0$ in notation), the law of the observed feedback up to time T when all coordinates of g_t are 0-mean for all t (i.e. there is no better direction). Under p_0, $f_t^0\sim\mathcal{N}(0,\sigma_t^2)$. By the definition of the total-variation distance (d_{TV}),

$$\mathbb{E}\big[\mathbb{I}\{y_{t,i}\geqslant-\alpha\}|Y=0\big]-\mathbb{E}\big[\mathbb{I}\{y_{t,i}\geqslant-\alpha\}|Y=i\big]\leqslant d_{\mathrm{TV}}(p_0,p_i)$$

$$\implies \mathbb{E}\big[\sum_{t=1}^{T} \mathbb{I}\{y_{t,i} \geqslant -\alpha\} | Y = 0\big] - \mathbb{E}\big[\sum_{t=1}^{T} \mathbb{I}\{y_{t,i} \geqslant -\alpha\} | Y = i\big] \leqslant T\, d_{\mathrm{TV}}(p_0, p_i).$$

By Pinsker's inequality we have

$$d_{\mathrm{TV}}(p_0, p_i) \leqslant \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p_0, p_i)}.$$

By the chain rule for the KL divergence / operations on conditional densities:

$$\begin{aligned} D_{\mathrm{KL}}(p_0, p_i) &= \mathbb{E}_{(f_1,\dots,f_T)\sim p_0}\Big[\log \frac{p_0\big(f_1,\dots,f_T\big)}{p_i\big(f_1,\dots,f_T\big)}\Big] \\ &= \sum_{t=1}^{T} \mathbb{E}_{(f_1,\dots,f_t)\sim p_0}\Big[\log \frac{p_0\big(f_t|f_{t-1}\dots,f_1\big)}{p_i\big(f_t|f_{t-1}\dots,f_1\big)}\Big] \\ &= \sum_{t=1}^{T} \mathbb{E}_{(f_1,\dots,f_{t-1})\sim p_0}\Big\{\mathbb{E}_{f_t\sim p_0(\cdot|f_1,\dots,f_{t-1})}\Big[\log \frac{p_0\big(f_t|f_{t-1}\dots,f_1\big)}{p_i\big(f_t|f_{t-1}\dots,f_1\big)}\Big]\Big\}. \end{aligned}$$

Now since y_t is a deterministic function of $f_1, \dots, f_{t-1}$ and given y_t, $f_t \sim \mathcal{N}(0, \sigma_t^2)$ under p_0 and $f_t \sim \mathcal{N}(\frac{1}{2} y_{t,i}, \sigma_t^2)$ under p_i, we have that the inner expectation is a KL divergence between Gaussians:

$$\begin{aligned} \mathbb{E}_{f_t\sim p_0(\cdot|f_1,\dots,f_{t-1})}\Big[\log \frac{p_0\big(f_t|f_{t-1}\dots,f_1\big)}{p_i\big(f_t|f_{t-1}\dots,f_1\big)}\Big] &= \frac{\big(0 - y_{t,i}/2\big)^2}{2\sigma_t^2} = \frac{y_{t,i}^2}{8\sigma_t^2} \\ \implies D_{\mathrm{KL}}(p_0, p_i) &= \sum_{t=1}^{T} \mathbb{E}_{(f_1,\dots,f_{t-1})\sim p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big] = \sum_{t=1}^{T} \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big] \\ \implies d_{\mathrm{TV}}(p_0, p_i) &\leqslant \sqrt{\sum_{t=1}^{T} \mathbb{E}_{p_0}\Big[\frac{y_{t,i}^2}{8\sigma_t^2}\Big]}. \end{aligned}$$

Combining gives the result. □

4.10.3 Case $p \to 1$

Theorem 4.10.6. *Fix T and $\delta > 0$. Consider $d > \max\Big\{T/\delta, 8^4 e^4 T^2, 8\Big\}$ and $p \in [1, 1 + 1/\log d]$. For any OCO algorithm with bandit feedback on $V = \mathcal{B}_p$, there exists a sequence of random linear losses $(\ell_t)_{t\in[T]}$ with gradients $(g_t)_{t\in[T]}$ such that $\|g_t\|_{p_\star} \leqslant 1$ for all rounds t with probability at least $1 - \delta$ and*

$$\mathbb{E}\big[R_T\big] \geqslant \frac{T}{16},$$

where the expectation is with respect to the randomness of the losses.

Proof. We use almost the same loss construction as the proof of Theorem 4.10.4. Before the start of the game, draw $Y \sim Unif(1, ..., d)$ and define the losses as $\ell_t(x) = x^T g_t^Y$ where

$$g_{t,i}^Y \sim \begin{cases} \mathcal{N}(0, \sigma^2), & \text{if } i \neq Y, \\ \mathcal{N}(1/2, \sigma^2), & \text{if } i = Y, \end{cases}$$

where $\sigma = (4\sqrt{2}\exp(1)\sqrt{\log d})^{-1}$. The only difference with the proof of Theorem 4.10.4 being the value of σ. We also follow the same steps until we reach for $\alpha = (2/d)^{1/p}$:

$$\mathbb{E}\big[R_T\big] \geqslant \frac{1-\alpha}{2}\Big(\frac{T}{2} - T\frac{1}{2\sigma}\sqrt{\frac{T}{2d}}\Big).$$

- From $d \geqslant 8$, we have $p \leqslant 2$ and $2^{p+1} \leqslant 8 \leqslant d$ so $1 - \alpha = 1 - (2/d)^{1/p} \geqslant 1 - 2^{-p/p} = 1/2$.
- From the definition of $\sigma = (4\sqrt{2}\exp(1)\sqrt{\log d})^{-1}$

$$\frac{1}{2\sigma}\sqrt{\frac{T}{2d}} = 2\exp(1)\sqrt{\frac{T\log d}{d}} \leqslant 2\exp(1)\sqrt{\frac{T\sqrt{d}}{d}} = 2\exp(1)\sqrt{\frac{T}{\sqrt{d}}} \leqslant \frac{1}{4}$$

since $d \geqslant 8^4 e^4 T^2$. This gives:

$$\frac{T}{2} - T\frac{1}{2\sigma}\sqrt{\frac{T}{2d}} \geqslant \frac{T}{2} - \frac{T}{4} = \frac{T}{2}.$$

We hence have $\mathbb{E}\big[R_T\big] \geqslant T/16$. The following section ensures that the Lipschitz-condition is satisfied with high-probability. □

Bound on gradients

Recall that $p \leqslant 1 + \frac{1}{\log d}$ so $p_\star \geqslant \frac{\log d}{1} + 1$. We have

$$\mathbb{E}\big[\|X\|_\infty\big] \leqslant \sqrt{2\log d}.$$

By the Borell-TIS inequality,

$$\mathbb{P}\Big(\|X\|_\infty > \mathbb{E}\big[\|X\|_\infty\big] + \beta\Big) \leqslant \exp(-\beta^2/2).$$

So with probability $1 - \delta/T$, we have (using $d > T/\delta$)

$$\|X\|_\infty \leqslant \mathbb{E}\big[\|X\|_\infty\big] + \sqrt{2\log\frac{T}{\delta}} \leqslant \sqrt{2\log d} + \sqrt{2\log\frac{T}{\delta}} \leqslant 2\sqrt{2\log d}$$
$$\implies \|g_t\|_{p_\star} \leqslant \sigma\|X\|_{p_\star} + \frac{1}{2} \leqslant \sigma d^{1/p_\star}\|X\|_\infty + \frac{1}{2} \leqslant \sigma\exp(1)2\sqrt{2\log d} + \frac{1}{2} = 1$$

where the final equality follows from $\sigma = (4\sqrt{2}\exp(1)\sqrt{\log d})^{-1}$ and we also used that $d^{1/p_\star} \leqslant e$. By a union bound over all rounds, with probability $1 - \delta$, $\|g_t\|_{p_\star} \leqslant 1$ for all rounds t.

Chapter 5

Sample-Efficiency in Multi-Batch Reinforcement Learning: The Need for Dimension-Dependent Adaptivity

Data collection in Reinforcement Learning (RL) usually falls into two main paradigms: online and offline. An online learner interacts with the environment, immediately receiving feedback and adapting its decisions in real-time. In contrast in offline RL, the dataset is collected in a single batch prior to observing any feedback. Many practical applications consider RL algorithms with limited adaptivity, which fall between online and offline RL. For example, in clinical trials, groups of patients undergo multiple treatments simultaneously and the treatment allocations are only be updated once the outcomes of the all the previous group have been observed (Yu et al., 2021). Similar settings with parallelized data-collection include marketing, advertising and numerical simulations (Esfandiari et al., 2021). There are also applications where the data collection needs to be validated prior to deployment by a human due to concerns of safety (Dann et al., 2019) and fairness (Koenecke et al., 2020), which with large scale data collection, is not feasible to have at every data point.

This motivates considering the multi-batch learning model (Perchet et al., 2015) where a data-set

of size n is collected in K batches, and feedback from within a batch is only observed at the end of a batch. This means only data from previous batches can be used to select how the next batch is collected. This is also known as growing batch RL (Lange et al., 2012). We refer to it as *multi-batch RL* to avoid any confusion with offline RL, also called batch RL, which considers a single batch.

The multi-batch setting covers different levels of adaptivity to feedback, which refers to how frequently the learner can process feedback and use it to update its data collection strategy. We measure it by the number of batches K. At one extreme, all the data is collected in a single batch ($K = 1$, offline-RL, no adaptivity). At the other extreme, each batch contains a single data-point ($K = n$, full adaptivity[1]). We say a learner is *adaptive* when $K > 1$. In some of the applications that parallelize data-collection mentioned above, algorithms with low-adaptivity where K is as small as possible are of interest since there may be a cost associated to unifying and processing the parallelized data.

We study adaptivity in infinite-horizon discounted Markov Decision Processes (MDPs). We focus on the best policy identification (BPI) problem, where the learner is tasked with finding a near-optimal policy. A desirable property of the learner is that it is **sample-efficient**, i.e. it only needs a dataset size that is polynomial in the dimension of the problem (e.g. state space size). We aim to understand the minimum level of adaptivity necessary for sample efficient learning.

Linear Function Approximation. MDPs faced in practice often have state spaces $\mathcal{S}$ or action spaces $\mathcal{A}$ that are infinite or too large to handle directly (Silver et al., 2016). Function approximation is used to reduce the learning problem to a smaller set of parameters that leverage structure in the problem. We consider a form of linear function approximation (Bellman et al., 1963) that assumes the (action)-value of a policy is a linear combination of known features of the state-action pair with an unknown parameter, each of dimension d. The sample complexity of algorithms is then measured with respect to the smaller dimension d instead of the dimensions of the MDP ($|\mathcal{S}|, |\mathcal{A}|$).

Adaptive vs Non-Adaptive. It is known there is a sample-efficiency separation between offline RL ($K = 1$) and fully-adaptive RL ($K = n$) under linear function approximation. Al-

[1]Typically, online refers to a setting with full-adaptivity along a single trajectory (sequence of transitions from a starting state, potentially with restarts). Our notion of full-adaptivity covers this, but is more general since we allow settings with a generative model where samples from any state-action pair can be drawn.

gorithms have been shown to be sample-efficient in the fully-adaptive setting (Lattimore et al., 2020) and under various assumptions in the offline setting (Duan et al., 2020; Xie and Jiang, 2020). Without assumptions in the offline setting, it has been shown information-theoretically that no sample-efficient algorithms can solve the BPI problem (Zanette, 2021) even under the best possible offline dataset, showing the separation. However, the MDP constructions of Zanette (2021) are easily solved with algorithms using $K = 2$ batches (see proof of their Theorem 3), suggesting that the boundary of this separation may be between offline RL ($K = 1$) and RL with adaptivity ($K > 1$). This motivates studying the values of K where sample-efficiency is not possible and asking the following questions:

Does the boundary of sample-efficiency under linear function approximation lie between offline RL ($K = 1$) and RL with adaptivity ($K > 1$)? If not, is the boundary dimension-dependent?

In this chapter, we establish a $\Omega(\log \log d)$ lower-bound on the number of batches K, required to solve a BPI problem sample-efficiently. This answers the first question negatively and the second positively. This is achieved through a non-trivial extension of the framework of (Zanette, 2021) for the offline setting to the multi-batch setting, which we describe next.

Learning Process. When faced with an unknown MDP within a known class of MDPs, we consider a learner over K rounds. In a round, the learner chooses a set of state-action queries with knowledge of the feedback from previous rounds. The following characteristics strengthen our lower bounds:

- Exact feedback: the feedback for a state-action query is the reward and transition *functions*, not a single sample. This makes a query equivalent to observing infinite samples from the reward and transition functions if these are stochastic and removes any hardness due to uncertainty.
- At each round the learner explicitly specifies the state-action queries and our results hold for any such set of queries whose size is polynomial in d.
- Realizability: the MDPs considered satisfy the linear representation of the action-values for *every* policy.

Tabular and finite-horizon MDPs and linear bandits are easily solved in the offline setting ($K = 1$) under this framework but infinite-horizon discounted MDPs are not (Zanette, 2021). Our work studies what happens beyond the offline setting ($K > 1$) for infinite-horizon discounted MDPs.

Contributions: Our results show that a number of batches K constant with respect to the dimension d is not enough to solve BPI problems sample-efficiently. Specifically, we show that if we restrict the total number of queries (over all batches) to be polynomial in d (sample-efficiency), then there are BPI problems that require $K = \Omega(\log \log d)$ to be solved to arbitrary accuracy, even if *all* policies satisfy linear realizability of their action-values.

These results show that adaptivity ($K > 1$) does not guarantee sample-efficiency. The level of adaptivity, or number of batches K, needed to guarantee sample-efficiency scales with the dimension d of the linear representation. In particular, the boundary of sample-efficiency does not lie between offline RL ($K = 1$) and adaptive RL ($K > 1$). Instead this boundary must lie within a regime of adaptivity scaling with dimension: $\Omega(\log \log d) \leqslant K \leqslant n$. **This establishes that high-dimensional problems in this setting may be fundamentally harder than low-dimensional ones and optimal methods for the latter are not for the former.**

The class of MDPs considered in Zanette (2021) can be solved with $d + 1$ queries and $K = 2$ batches (observing feedback from the first batch is enough to select queries in the second that fully solve the MDPs) [Zanette (2021), Theorem 4]. Our results show that this is not possible in general and that the class of MDPs we use for our results is fundamentally harder. From a technical perspective, our work uses tools from the theory of subspace packing with chordal distance (Soleymani and Mahdavifar, 2021). This enables the environment to erase information across multiple dimensions (m-dimensional subspaces, see Section 5.3.1) in response to queries, instead of a single direction as in Zanette (2021), which ultimately allows us to achieve lower-bounds for $K > 1$.

Roadmap. In Section 5.1, we define the problem setting. In Section 5.2, we present and discuss the main results of this chapter. In Section 5.3, we provide the analysis of our main result, including some intuition, an introduction to subspace packing theory and the full details of the proof of Theorem 5.2.1. In Section 5.4, we discuss related works. In Section 5.5, we conclude and discuss future-work. Finally, in Section 5.6, we provide technical proofs omitted earlier in the chapter.

Bibliographic remark: This chapter is based on Johnson et al. (2024) and is my own work. It is licensed under a Creative Commons Attribution 4.0 International License.

Setting	Performance Criteria	Main Theorem(s)	Publication Source
Multi-Batch RL	Best-Policy Identification	Theorem 5.2.1	Johnson et al. (2024)

Table 5.1: Chapter overview.

5.1 Problem Setting

In this section, we formally define the reinforcement learning (RL) setting and the learning model we consider. We borrow the framework from the work of Zanette (2021) and extend it beyond the offline RL setting. We begin with some preliminaries on Markov Decision Processes.

5.1.1 Markov Decision Processes

We refer the reader to Chapter 3 for the general definition of a Markov Decision Process (MDP) and associated concepts and notation. We specify some details relevant to this chapter below.

We consider discounted MDPs where the discount factor γ is in $[0, 1)$ and the state and action space is not necessarily finite (for which we assume all necessary conditions so that the MDP is well defined). The action set is also not the same in all state: $\mathcal{A} = \bigcup_{s\in\mathcal{S}}\{\mathcal{A}_s\}$ where $\mathcal{A}_s$ is the action space in state $s \in \mathcal{S}$. The reward function $r(s, a) \in [-1, 1]$ is deterministic.

For a policy π, we recall the Bellman evaluation operator $\mathcal{T}^\pi$ for action-value functions,

$$(\mathcal{T}^\pi Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}[Q(s', \pi(s'))].$$

and the Bellman optimality operator $\mathcal{T}^\star$,

$$(\mathcal{T}^\pi Q)(s, a) = r(s, a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)}[\sup_{a'\in\mathcal{A}_{s'}} Q(s', a')].$$

Both Q^π and $Q^\star_M$ are the unique fixed points of $\mathcal{T}^\pi$ and $\mathcal{T}^\star$, respectively (Puterman, 1994).

Notation. Given an MDP M, we will sometimes write V^π_M to denote the value of a policy π in the MDP M (similarly for $V^\star_M, Q^\pi_M, Q^\star_M, p_M, r_M, \mathcal{T}^\pi_M$). $\mathcal{B} = \mathcal{B}_2 = \left\{x \in \mathbb{R}^d : \|x\|_2 \leqslant 1\right\}$ denotes the Euclidean unit ball, and $\partial\mathcal{B} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$ its boundary. For n vectors $v_1, ..., v_n \in \partial\mathcal{B}$, we denote the subspace spanning the vectors by $\langle v_1, ..., v_n\rangle$.

In this chapter, we are interested in solving the best policy identification problem. We formally define it in the following section.

5.1.2 Best Policy Identification (BPI)

Let $\mathcal{M}$ be a class of MDPs with the same $\mathcal{S}$ and $\mathcal{A}$. A BPI problem defined by $\mathcal{M}$ consists of:

1. **An instance** $(\bar{s}, M, \mathcal{M})$, where $M \in \mathcal{M}$ is an MDP in $\mathcal{M}$ and $\bar{s} \in \mathcal{S}$ is a starting state. $\mathcal{M}$ and $\bar{s}$ are known but M is unknown.

2. **An interaction procedure** with the MDP M to collect a dataset $\mathcal{D}$ (see Section 5.1.3).

3. **An objective:** Based on $\mathcal{D}$, the learner produces an output $\widehat{\pi}_{\mathcal{D}}$ of a near-optimal policy for M. The performance of the learner is evaluated by (ε, δ)-soundness (Definition 5.1.1), i.e. the sub-optimality of the output policy on any instance.

Definition 5.1.1. *A learner is (ε, δ)-sound for BPI problems characterised by $\mathcal{M}$ if for all $M \in \mathcal{M}$, the learner faced with instance $(\bar{s}, M, \mathcal{M})$ outputs $\widehat{\pi}_{\mathcal{D}}$ that is ε-optimal with probability*[2] *at least $1-\delta$, i.e. it satisfies $\mathbb{P}\Big((V_M^\star - V_M^{\widehat{\pi}_{\mathcal{D}}})(\bar{s}) < \varepsilon\Big) > 1-\delta$.*

5.1.3 Multi-Batch Learning Model

In order to solve the BPI problem described above, the learner uses data coming from interactions with the environment. We consider the multi-batch learning model (Perchet et al., 2015) where a data-set of size n is collected in K batches, and feedback from within a batch is only observed at the end of a batch. This means only data from previous batches can be used to select how the next batch is collected. This is also known as growing batch RL (Lange et al., 2012). We refer to it as *multi-batch RL* to avoid any confusion with offline RL, also called batch RL, which considers a single batch.

A **query** is a state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}_s$ that is submitted to the unknown MDP M and for which feedback is returned. A query formalises how the learner interacts with an MDP, the feedback received is defined next.

Definition 5.1.2 (Query-Feedback)**.** *In return for a query (s, a) the environment provides as feedback the reward $r_M(s, a)$ and the transition function $p_M(\cdot|s, a)$.*

[2]There is no randomness in the feedback of the dataset $\mathcal{D}$ (see Section 5.1.3) so the probabilities are with respect to randomness arising from the learner's query selection or output strategies.

Remark 5.1.3. *The learner receives the transition function $p_M(\cdot|s,a)$ for a query (s,a), instead of a sample from $p_M(\cdot|s,a)$. This removes any statistical uncertainty and is equivalent to observing infinite samples, which strengthens any lower-bounds proven under this frame-work.*

Algorithm 1 Multi-Batch Learning Model

1: (Input) BPI problem and a number of batches K.
2: Initialise $\bar{\mathcal{D}}_0 = \varnothing$.
3: **for** $k = 1, ..., K$ **do**
4: (Query Selection) With knowledge of $\bar{\mathcal{D}}_{k-1}$, learner chooses a set of queries μ_k.
5: (Data Collection) Environment receives queries μ_k and returns to learner $\mathcal{D}_k$ (set of queries + feedback for all queries in μ_k - see Definition 5.1.2). Learner updates $\bar{\mathcal{D}}_k = \bigcup_{i=1}^{k} \mathcal{D}_i$.
6: **end for**
7: Set $\mathcal{D} = \bar{\mathcal{D}}_K = \bigcup_{i=1}^{K} \mathcal{D}_i$.
8: (Output) Learner returns $\widehat{\pi}_{\mathcal{D}}$.

We summarise the learning model in Algorithm 1. The learner has access to a generative model and explicitly selects the set of queries μ_k at round k (line 4) by selecting a set of state-action pairs. The learner has access to the dataset $\bar{\mathcal{D}}_{k-1}$, which contains the feedback of the queries from the previous rounds from the MDP M the learner is faced with. Let $\mathcal{M}_k \subset \mathcal{M}$ be the set of MDPs in $\mathcal{M}$ that would produce exactly the feedback in dataset $\bar{\mathcal{D}}_{k-1}$ given the queries in the previous rounds. Given the queries this is deterministic since there is no randomness in the feedback of a query (see Definition 5.1.2). The learner can use $\mathcal{M}_k$ in the selection of the queries at round k but the specific MDP it is interacting with remains unknown if $|\mathcal{M}_k| > 1$.

We denote $n_k = |\mu_k|$ the number of queries at round k and $n = \sum_{k=1}^{K} n_k$ the total number of queries.

Adaptivity: The multi-batch learning model encompasses different levels of adaptivity to feedback, which is measured by the number of batches K:

- For $K = 1$, the learner is **non-adaptive** and the dataset $\mathcal{D}$ is collected in a single batch. This is the model considered by Zanette (2021) for offline RL that we extend for general K.
- For $K > 1$, the learner is **adaptive** and the dataset $\mathcal{D}$ is collected in multiple batches. Queries for a batch are selected based on feedback from previous batches.
- For $K = n$ (i.e. each batch contains a single data-point), the learner is **fully-adaptive**. The queries are selected sequentially and depend on the feedback from previous queries.

5.1.4 Linear Function Approximation

MDPs faced in practice are often very high-dimensional and have state spaces $\mathcal{S}$ or action spaces $\mathcal{A}$ that are infinite or too large to handle directly (Silver et al., 2016). Function approximation is used to reduce the learning problem to a smaller set of parameters that leverage structure in the problem. We consider a form of linear function approximation (Bellman et al., 1963) that assumes the (action)-value of *every* policy is a linear combination of known features of the state-action pair with an unknown parameter, each of dimension d. The sample complexity of algorithms is then measured with respect to the smaller dimension d instead of the dimensions of the MDP ($|\mathcal{S}|, |\mathcal{A}|$). Specifically:

Assumption 5.1.4 (Q^π-Realizability for every policy (Zanette (2021), Assumption 3))**.** *Given any BPI $(\bar{s}, M, \mathcal{M})$ problem instance, there exists a known feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ s.t. $\|\phi(\cdot,\cdot)\|_2 \leqslant 1$ and for any policy π there exists $\theta_M^\pi \in \mathcal{B}$ s.t. for all $(s,a) \in \mathcal{S} \times \mathcal{A}$,*

$$Q_M^\pi(s,a) = \phi(s,a)^T \theta_M^\pi.$$

This is a form of linear function approximation that is strictly more general than linear MDPs (Zanette et al., 2020), which assume the reward and transition functions are linearly representable. We assume the learner is aware that this assumption holds.

5.1.5 Sample Efficiency

A desirable property of the learner is that it is **sample-efficient**, i.e. it only needs a dataset size that is polynomial in the dimension of the problem:

Definition 5.1.5. *A learner for a BPI problem under Assumption 5.1.4 is sample-efficient if its total number of queries $n = \sum_{k=1}^K n_k$ scales polynomial with d.*

We aim to understand the minimum level of adaptivity (i.e. the smallest K can be in our multi-batch learning model) necessary for sample efficient learning. We are finally ready to present the main results of this chapter.

5.2 Main Result

In this section, we present the central theoretical contribution of this chapter: a lower bound on the number of batches K required to achieve sample efficiency in infinite-horizon MDPs. While

previous work by Zanette (2021) established that offline RL ($K = 1$) is insufficient for sample efficiency, it remained an open question whether a minimal amount of adaptivity such as a constant number of batches, would suffice to overcome these barriers. Our results demonstrate that the requirement for adaptivity is not binary but is instead intrinsically linked to the dimensionality of the feature space.

The following theorem establishes that for any learner to be sample-efficient, the number of batches K must scale with the dimension d.

Theorem 5.2.1 (Main Lower Bound)**.** *Fix d to be sufficiently large and the discount factor $\gamma > \sqrt{3/4}$. There exists a class of MDPs $\mathcal{M}$ satisfying Assumption 5.1.4 (Q^π-realizability for all π) such that any learner that is $(1, 1/2)$-sound using a total number of queries n polynomial in d must use at least*

$$K = \Omega(\log \log d)$$

adaptive batches.

The proof is technical and builds on results from subspace packing theory; we present the full details in Section 5.3.

Clarification on $(1, 1/2)$-soundness. To appreciate the strength of this lower bound, consider the scale of the value functions. Since rewards are in $[-1, 1]$, the values V^π are generally bounded within $[-\frac{1}{1-\gamma}, \frac{1}{1-\gamma}]$, but in the specific MDP class we construct for this result, the values V^π are actually contained within $[-1, 1]$. By setting $\varepsilon = 1$ and $\delta = 1/2$, we are stating that the learner cannot guarantee a sub-optimality better than half the range of the value, nor can it succeed with more than a coin-flip's probability. Essentially, a learner with $K = o(\log \log d)$ batches is performs no better than an uninformed guess.

The significance of Theorem 5.2.1 extends beyond the specific $\log \log d$ rate. It recharacterizes the complexity of RL in the following ways:

- **Dimension-dependent adaptivity.** The boundary of sample efficiency is not simply determined by whether $K = 1$ (offline) or $K > 1$ (adaptive). Instead, the number of batches required grows with the dimension d of the linear representation:

$$\Omega(\log \log d) \leqslant K \leqslant n.$$

This shows that minimal adaptivity alone is insufficient in high-dimensional settings.

- **A separation between low and high-dimensional problems.** For a fixed batch budget K, there exist low-dimensional problems where K is large enough to guarantee adaptivity sufficient for sample efficiency. However, as d increases, the same budget may fall below the threshold required by Theorem 5.2.1, making sample-efficient learning impossible. This demonstrates a fundamental separation between low- and high-dimensional RL problems.

Comparison to Zanette (2021). The results of Zanette (2021) show that there exists a class of MDPs $\mathcal{M}$ satisfying Assumption 5.1.4 (i.e., Q^π-realizability for all π) such that any learner that is $(1, 1/2)$-sound with $K = 1$ (offline RL) must use a number of queries

$$n > \frac{1}{(1-\gamma)^d}.$$

This demonstrates that, with a single batch, the sample complexity required to solve the BPI problem to meaningful accuracy scales exponentially with the dimension d. In particular, sample-efficient learning is impossible in the offline setting ($K = 1$), establishing a separation from the fully-adaptive setting ($K = n$) where sample-efficient learning is known to be achievable (Lattimore et al., 2020).

Our results go beyond the offline setting and consider sample-efficiency in the presence of adaptivity. When considering sample-efficiency, our lower bounds are strictly stronger: they encompass the offline case and extend to settings with low adaptivity. We also note the following:

- **Different type of lower bound:** Zanette (2021) lower-bound the number of queries n required when $K = 1$ is fixed, whereas our work lower-bounds the number of batches K necessary for sample-efficient learning with a polynomial query budget. In other words, they show that offline learning may require exponentially many queries, while we show that even with a polynomial query budget, insufficient adaptivity (small K) prevents sample-efficient learning.

- **Adaptivity changes the difficulty:** The MDP constructions in Zanette (2021) can be solved with just $K = 2$ batches; feedback from the first batch suffices to select queries in the second batch that fully reveal the MDP. This highlights that a small amount of

adaptivity is sufficient for their specific constructions, whereas our MDP constructions are fundamentally harder and require adaptivity scaling with the dimension.

A discussion of the differences in MDP constructions and proof techniques can be found in Section 5.3.1.

Comparison to Lattimore et al. (2020). The work of Lattimore et al. (2020) studies methods achieving sample-efficiency with a generative model under linear function approximation. Under Assumption 5.1.4, and assuming finite state and action spaces, they show that in the fully-adaptive setting (corresponding to $K = n$ in our multi-batch framework), the LSPI algorithm (Lagoudakis and Parr, 2003) produces an ε-optimal policy with probability at least $1 - \delta$ using

$$\tilde{O}\left(\frac{d\log(1/\delta)}{\varepsilon^2(1-\gamma)^8}\right)$$

queries, where we suppress logarithmic factors except for the dependence on δ. The sample-complexity is linear in the dimension d, showing that full adaptivity is sufficient for sample-efficient learning, whereas our results demonstrate that partial adaptivity can be insufficient in high-dimensions. This highlights a qualitative gap between low-adaptivity and high-adaptivity regimes in high-dimensional reinforcement learning under linear function approximation.

In fact, the full-adaptivity $K = n$ of LSPI can be reduced. In each iteration of approximate policy iteration, the action-value of the current policy under consideration is estimated at N state-action pairs by rolling out the policy over m parallel trajectories of length H. The feedback across the m trajectories and N points can be included within a single batch, the learner only needs to observe feedback between iterations of the approximate policy iteration and each step of the roll-outs of horizon H. Therefore, the LSPI algorithm only requires TH batches, where T is the number of approximate policy iterations. To obtain the query complexity above, $T = \frac{1}{1-\gamma}\log\left(\frac{2}{\varepsilon(1-\gamma)}\right)$ and $H = \frac{1}{1-\gamma}\log\left(\frac{8\sqrt{d}}{\varepsilon(1-\gamma)^3}\right)$. The number of batches used by LSPI can therefore be only $O(\log(d))$, which is only a logarithmic factor away from our lower-bound. It is unclear if the LSPI algorithm could be modified to only use $O(\log\log d)$ batches or if our lower-bound can be improved to $\Omega(\log d)$. Our result is a separation showing that constant adaptivity is insufficient, while the exact threshold remains open.

We assume an infinite action space in our proofs, whereas Lattimore et al. (2020) consider finite state–action spaces. This difference is not fundamental: our lower bounds would continue to hold under a sufficiently fine discretization of the action space.

Policy Evaluation & Policy-Induced Queries. Analogous lower bounds also hold for the *policy evaluation* problem, where the learner's objective is to estimate the action-value function Q^π of a target policy rather than to identify an optimal policy. Similar results further extend to an alternative query model in which the learner can collect data along trajectories generated by a policy. This latter extension, however, requires linear realizability only for the target policy π, rather than for all policies. The formal statements and proofs of these results are provided in Johnson et al. (2024) but the core ideas and proof techniques are the same as those presented in this chapter.

Remark 5.2.2. *The lower bound holds under exact transition feedback and exact Q^π-realizability for all π. Relaxing these assumptions could lead to a lower-bound with a larger dependence on d. However, it is unclear how relaxing these assumptions with the current construction (e.g. by introducing stochasticity into the deterministic transitions - see Section 5.3) would fundamentally change the dependence on d.*

5.3 Analysis

In this section, we present the proof of our main result, Theorem 5.2.1. The proof is technically involved and relies on tools from subspace packing theory. We therefore begin with a high-level overview of the proof strategy, which provides intuition and serves as a guide for the detailed arguments that follow.

5.3.1 Proof Sketch

Our construction extends the ideas of Zanette (2021) from the offline reinforcement learning setting to our multi-batch problem (see the end of this section for a comparison). We consider a BPI instance satisfying Assumption 5.1.4, with feature vectors $\phi(\cdot, \cdot)$ ranging over the unit Euclidean ball $\mathcal{B}$ (see Figure 5.1).

The MDPs we construct have deterministic transitions and a simple structure: the action set in the initial state corresponds to the Euclidean unit ball $\mathcal{B}$, while all subsequent states admit only

a single available action (the process never returns to the initial state).

Consider the first batch of data consisting of n_1 queries. Let (s_i, a_i) denote the i-th queried state–action pair and let s_i^+ be the corresponding (deterministic) successor state. Define the matrices

$$\Phi = \begin{bmatrix} \phi(s_1, a_1)^\top \\ \vdots \\ \phi(s_{n_1}, a_{n_1})^\top \end{bmatrix}, \quad r = \begin{bmatrix} r(s_1, a_1) \\ \vdots \\ r(s_{n_1}, a_{n_1}) \end{bmatrix}, \quad \Phi^+ = \begin{bmatrix} \phi(s_1^+, a)^\top \\ \vdots \\ \phi(s_{n_1}^+, a)^\top \end{bmatrix},$$

where a denotes the unique available action at each successor state. Since $Q_M^\star$ is the fixed point of the Bellman optimality operator, i.e.,

$$Q_M^\star(s, a) = (\mathcal{T}^\star Q_M^\star)(s, a),$$

and by Assumption 5.1.4 there exists a parameter vector $\theta_M^\star$ such that $Q_M^\star(s, a) = \phi(s, a)^\top \theta_M^\star$ for all (s, a), the learner seeks to recover a parameter vector θ satisfying the local Bellman equations

$$\Phi\theta = r + \gamma\Phi^+\theta \quad \iff \quad (\Phi - \gamma\Phi^+)\theta = r.$$

The key point that allows the learner to write the Bellman optimality equations above is the simplicity of the successor states: since each successor state admits a single available action, the learner does not need to reason about action selection there. As a result, the problem effectively reduces to the policy evaluation problem of determining $Q^\pi(\bar{s}, a)$ for all $a \in \mathcal{B}$ which is identical for all policies π (since after moving away from the initial state $\bar{s}$ all policies take the same actions). We describe next how an adversary can make this challenging for the learning.

Let $X := \Phi - \gamma\Phi^+$. If X is not full rank, the linear system above does not admit a unique solution, and the data collected in the first batch is insufficient to identify $\theta_M^\star$. Crucially, while the learner controls the choice of Φ through its queries, the environment having observed Φ, can adversarially choose the successor features Φ^+ so as to maximise the dimension of the null space[3] of X. Intuitively, this corresponds to erasing information along multiple directions of the feature space (see Figure 5.1, left).

[3]By the rank–nullity theorem, this is equivalent to minimising the rank of X.

This phenomenon is a manifestation of *bootstrapping*: the value of a policy at a given state depends on its value at successor states. In our construction, this dependence allows the environment to adversarially select successor states in a way that suppresses information revealed by the learner's queries. Bootstrapping is the key mechanism driving the hardness of best policy identification in the multi-batch setting under a generative model.

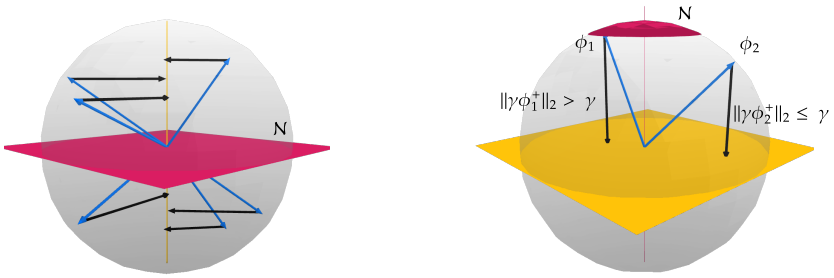

Figure 5.1: Left: **Information can be erased in multiple directions:** Consider the setting where information is being erased along the pink plane $\mathcal{N}$: the learner's queries $\phi(s_i, a_i)$ are shown in blue and the environment's responses $-\gamma\phi(s_i^+, a)$ are shown in black. The rows of X, $\phi(s_i, a_i) - \gamma\phi(s_i^+, a)$ (blue + black vectors) all lie on the yellow line so the learner acquires no information in the directions of the pink subspace $\mathcal{N}$, the null-space of X. Right: **Information cannot be erased in all directions:** Consider the opposite setting where information is being erased along the pink line $\mathcal{N}$. Because of the constraint $\|\phi(s, a)\|_2 \leqslant 1$ and $\gamma < 1$, a query ϕ_1 (in blue) in the pink cap cannot have $\phi_1 - \gamma\phi_1^+$ (blue + black vector) projected back onto the yellow plane (unless $\|\gamma\phi_1\|_2 > \gamma \implies \|\phi_1\|_2 > 1$). Despite the environment not being able to erase information in certain directions, if the number of queries is "small", it can find directions to erase.

Although the learner can prevent erasure along certain directions (Figure 5.1, right), if n_1 is polynomial in d, this only constrains a limited number of directions. We show that if n_1 is sub-exponential in $d^{1/4}$, then there exists a subspace of dimension $d^{1/4}$ that can be embedded in the null space of X.

Before selecting queries for the second batch, the learner observes the feedback from the first batch and can identify the directions along which information is missing. It may then focus its second-batch queries on these directions. However, the null space remains at least $d^{1/4}$-dimensional, and the same reasoning applies inductively. If n_2 is sub-exponential in $d^{1/16}$, the null space after two batches still has dimension at least $d^{1/16}$.

Proceeding recursively, after k rounds, if n_k is sub-exponential in $d^{1/4^k}$, then the null space of the accumulated Bellman equations remains at least $d^{1/4^k}$-dimensional. As long as $\exp(d^{1/4^k})$

exceeds a polynomial scaling in d, a sample-efficient learner cannot prevent a non-empty null space and hence cannot identify the optimal policy. Therefore, the learner must reach a round K such that $\exp(d^{1/4^K})$ becomes polynomial in d, which requires $K = \Omega(\log\log d)$, from which we get our lower-bound.

Formal MDP construction. The above intuition is formalised using the MDP class defined in Section 5.3.4. Each MDP is indexed by a nested sequence of subspaces and a sign $\pm$, with carefully designed reward and transition functions. We show that if $n_k = \text{poly}(d)$ for all k and $K = o(\log\log d)$, then with probability at least $1/2$ the learner cannot distinguish between two MDPs that induce opposite optimal actions in the initial state. Consequently, any learner must incur error with probability at least $1/2$, establishing $(1, 1/2)$-soundness. Full details appear in Section 5.3.7, with additional illustrations in Section 5.3.8.

Potential improvements. As discussed in the comparison to Lattimore et al. (2020) in Section 5.2, it is unclear if our lower-bound is tight. Strengthening the lower bound would require improved subspace packing results, specifically, packings of exponentially many subspaces of dimension cd (for some $c < 1$) with sufficiently large minimal chordal distance (see Section 5.3.2). Such a result would yield a tight $\Omega(\log d)$ lower bound. To the best of our knowledge, no such packing or matching impossibility result is currently available. Importantly, our MDP construction is modular and can incorporate any future advances in subspace packing theory.

Comparison to Zanette (2021): The work of Zanette (2021) erases information in the first round along a 1-dimensional subspace (X is of rank $d-1$). Therefore after having observed one round of feedback, a single additional query is sufficient to distinguish the MDP and solve the problem. Therefore their construction can only give the lower-bound $K > 1$. Our constructions erase information along m-dimensional subspaces where we attempt to maximise m so that even after having observed feedback revealing this subspace, a polynomial number of queries remains insufficient to distinguish the MDP and more adaptive rounds are needed.

5.3.2 Preliminaries on Subspace Packings

Before proving Theorem 5.2.1, we introduce several notions that underpin our lower-bound constructions. In particular, we define hyperspherical caps and sectors associated with subspaces, review subspace packings in Grassmannian spaces, and establish the existence of high-dimensional subspaces whose associated sectors avoid a prescribed finite set of points. These tools will enable

us to show that, when the number of queries is sub-exponential in dimension, the environment can define the transitions of the MDP instance to adversarially erase information along carefully chosen directions.

Hyper-Spherical Caps and Sectors for Subspaces

Recall that $\mathcal{B} = \{x \in \mathbb{R}^d : \|x\|_2 \leqslant 1\}$ is the d-dimensional unit ball.

Definition 5.3.1. *Fix $w \in \mathcal{B}$. Define the γ-hyperspherical cap of w as:*

$$\mathcal{C}_\gamma(w) = \Big\{x \in \mathcal{B} : \frac{x^T w}{\|w\|_2} > \gamma\Big\}.$$

A vector x is in the γ-hyperspherical cap of w if the angle θ between x and w satisfies

$$\gamma < \|x\|_2 \cos\theta \iff \theta < \arccos\left(\frac{\gamma}{\|x\|_2}\right).$$

With γ close to 1, these represent a set of vectors in the hyper-cone around w that are close to the boundary of $\mathcal{B}$ (see Figure 5.1 right). The key property that motivates considering vectors in this set is that they require a vector of norm greater than γ to be projected in a direction orthogonal to w (i.e. if $x^\perp$ is the projection of $x \in \mathcal{C}_\gamma(w)$ in a direction orthogonal to w, then $\|x^\perp - x\|_2 \geqslant \gamma$). We extend the notion of γ-hyperspherical caps to subspaces of multiple dimensions.

Definition 5.3.2. *Fix a subspace H of $\mathbb{R}^d$. Define the γ-hyperspherical sector of H as*

$$\overset{\triangle}{\mathcal{C}}_\gamma(H) = \Big\{x \in \mathcal{B} : \exists v \in H \text{ s.t. } \frac{x^T v}{\|v\|_2\|x\|_2} > \gamma\Big\}.$$

See Figure 5.2 for an illustration.

Beyond the extension to subspaces of multiple dimensions, Definition 5.3.2 differs from Definition 5.3.1 in two ways. It is a hyper-spherical sector rather than cap, which does not restrict the vectors to be close to the boundary of $\mathcal{B}$. It is two sided meaning that if $x \in \overset{\triangle}{\mathcal{C}}_\gamma(H)$, then $-x \in \overset{\triangle}{\mathcal{C}}_\gamma(H)$. Note that the subspace H is defined on $\mathbb{R}^d$ but $\overset{\triangle}{\mathcal{C}}_\gamma(H) \subset \mathcal{B}$.

Similar to the intuition in the 1-dimensional case, a vector x is in $\overset{\triangle}{\mathcal{C}}_\gamma(H)$ if there is a vector in H whose angle with x is "small". It is equivalent to taking the unions of the 1-dimensional γ-hyperspherical sectors of all the vectors in H. Again, the key property that motivates considering vectors in this set is that they require a vector of norm greater than γ to be projected in

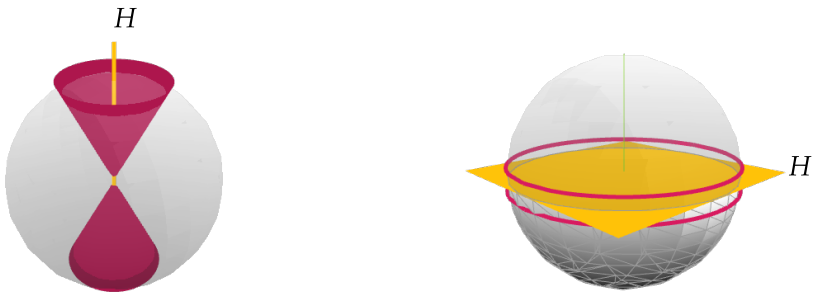

Figure 5.2: Illustration of a hyperspherical sector of a 1-dimensional subspace (left) and a 2-dimensional subspace (right - all vectors whose direction is within the two pink bands). In both cases, the subspace H is in yellow.

a direction orthogonal to H.

We also note that a vector x is not in $\overset{\triangle}{\mathcal{C}}_\gamma(H)$ if the subspace H does not intersect $\overset{\triangle}{\mathcal{C}}_\gamma(\langle x \rangle)$, the γ-hyperspherical sector of $\langle x \rangle$. This allows us to work with the γ-hyperspherical sectors of one dimensional subspaces (span of a single vector) instead of subspaces of multiple dimensions. We will also abuse notation and write $\overset{\triangle}{\mathcal{C}}_\gamma(x)$ instead of $\overset{\triangle}{\mathcal{C}}_\gamma(\langle x \rangle)$

Subspace Packing

We now formalize subspace packings in Grassmannian space. Let $\mathcal{G}_{m,d}(\mathbb{R})$ denote the set of all m-dimensional subspaces of $\mathbb{R}^d$, known as the **Grassmannian space**. An element $A \in \mathcal{G}_{m,d}(\mathbb{R})$ is an m-dimensional subspace of $\mathbb{R}^d$.

We present a measure of distance between subspaces known as the **chordal distance** (Conway et al., 1996). Fix two subspaces $A, B \in \mathcal{G}_{m,d}(\mathbb{R})$. The principal angles $\theta_1, ..., \theta_m \in [0, \pi/2]$ between A and B are defined as

$$\cos\theta_i = \max_{a \in A} \max_{B \in B} \frac{a^T b}{\|a\|_2 \|b\|_2} = a_i^T b_i,$$

for $i = 1, ..., m$ such that $\|a_i\|_2 = \|b_i\|_2 = 1$, $a^T a_j = 0$, $b^T b_j = 0$ for $1 \leqslant j \leqslant i-1$. The chordal distance d_c is then defined as

$$d_c(A, B) = \sqrt{\sin^2\theta_1 + \sin^2\theta_2 + ... + \sin^2\theta_m}.$$

We define the notion of subspace packing, which is the usual notion of a packing where the set is the Grassmanian space $\mathcal{G}_{m,d}(\mathbb{R})$ and the distance is the chordal distance.

Definition 5.3.3. *A subspace packing $\mathcal{C}$ of $\mathcal{G}_{m,d}(\mathbb{R})$ is a set of m-dimensional subspaces in $\mathbb{R}^d$ of size $|\mathcal{C}|$, i.e. it is a subset of $\mathcal{G}_{m,d}(\mathbb{R})$. The minimum distance between elements of $\mathcal{C}$ is measured by the chordal distance and is denoted*

$$d_{\min}(\mathcal{C}) = \min_{A,B\in\mathcal{C},A\neq B} d_c(A,B).$$

The following result establishes the existence of a packing of a certain size with lower bounded minimal chordal distance.

Lemma 5.3.4 (Soleymani and Mahdavifar (2021), Theorem 4)**.** *Fix $d = 2^N$ for $N \in \mathbb{N}$ and integers $k < d$, $m < d$. There exists a packing $\mathcal{C} \subset \mathcal{G}_{m,d}(\mathbb{R})$ with*

$$|\mathcal{C}| = \left(\frac{d}{2}\right)^{\lceil k/2\rceil-1}\left\lfloor\frac{d}{m}\right\rfloor \quad \text{and} \quad d_{\min}(\mathcal{C}) \geqslant \sqrt{m\Big(1-\frac{m(k-1)^2}{d}\Big)}.$$

Theorem 4 from Soleymani and Mahdavifar (2021) is presented for packings in $G_{m,d}(\mathbb{C})$ but they give (in Remark 1) a mapping from a packing in $G_{m/2,d/2}(\mathbb{C})$ to a packing in $G_{m,d}(\mathbb{R})$ that preserves the normalized distance δ_c ($= d_{\min}(\mathcal{C})/\sqrt{m}$) between the elements of the packing, giving the result presented above.

Existence of an Isolated Subspace

We now state the key lemma that underlies our lower-bound construction: for any "small" set of points in $\mathcal{B}$, there exists a high-dimensional subspace whose γ-sector avoids all points, allowing the environment to define MDP transitions that erase information along this subspace (see Section 5.3.1).

Lemma 5.3.5. *[Existence of an isolated subspace]*
Fix $d = 2^N$ for $N \geqslant 8$ and let $D = \{y_1, \ldots, y_n\} \subset \mathcal{B}$. If

$$n + 1 \leqslant \left(\frac{d}{2}\right)^{\frac{1}{8}g(\gamma)d^{1/4}}, \quad g(\gamma) = 2\gamma^2 - 1,$$

then there exists a subspace $A \in \mathcal{G}_{2^{\lceil N/4\rceil},d}(\mathbb{R})$ such that

$$y_i \notin \overset{\triangle}{\mathcal{C}}_\gamma(A) \quad \forall i \in [n].$$

The proof of Lemma 5.3.5 relies on the following lemma.

Lemma 5.3.6. *Fix $d = 2^N$ and $n+1 \leqslant (\frac{d}{2})^{\frac{1}{8}g(\gamma)d^{1/4}}$. If there exists a packing $\mathcal{C} \subset \mathcal{G}_{m,d}(\mathbb{R})$ with $|\mathcal{C}| \geqslant n+1$ and $d_{\min}(\mathcal{C}) \geqslant \sqrt{m - g(\gamma)^2}$, then for any n points $\{y_1, \ldots, y_n\} \subset \mathcal{B}$, there exists a subspace $H \in \mathcal{C}$ such that*

$$y_i \notin \overset{\triangle}{\mathcal{C}}_\gamma(H), \quad \forall i \in [n].$$

This lemma establishes that if $n+1$ subspaces are sufficiently far in terms of chordal distance, then for any n points there is a subspace whose γ-sector does not contain any of the n points. The proof is given in Section 5.6.1

Proof of Lemma 5.3.5. To prove Lemma 5.3.5, we show the existence of a subspace packing $\mathcal{C}$ in $\mathcal{G}_{m,d}(\mathbb{R})$ (with $m = 2^{\lceil N/4 \rceil}$) and use Lemma 5.3.6. The subspace packing must satisfy two conditions:

- $|\mathcal{C}| \geqslant n+1$.
- $d_{\min}(\mathcal{C}) \geqslant \sqrt{m - g(\gamma)^2}$.

To show the existence of a suitable subspace packing, we use Lemma 5.3.4 with $k = \lfloor \frac{g(\gamma)}{m}\sqrt{d}+1 \rfloor$ ($\leqslant d$). Lemma 5.3.4 gives a packing $\mathcal{C}$ of size $|\mathcal{C}| \geqslant \left(\frac{d}{2}\right)^{\frac{g(\gamma)}{2m}\sqrt{d}-1} \lfloor \frac{d}{m} \rfloor$ s.t

$$d_{\min}(\mathcal{C}) \geqslant \sqrt{m - g(\gamma)^2}.$$

Substituting in $m = 2^{\lceil N/4 \rceil}$ into the lower-bound on the size of $\mathcal{C}$ gives

$$\lfloor \frac{d}{m} \rfloor \left(\frac{d}{2}\right)^{\frac{g(\gamma)\sqrt{d}}{2m}-1} = \lfloor \frac{d}{m} \rfloor \left(\frac{d}{2}\right)^{-3/4} \left(\frac{d}{2}\right)^{\frac{g(\gamma)}{2}2^{N/2-\lceil N/4 \rceil}-\frac{1}{4}} \geqslant 2^{3/4}\frac{\lfloor 2^{3N/4} \rfloor}{2^{3N/4}} \left(\frac{d}{2}\right)^{\frac{g(\gamma)}{4}2^{N/4}-\frac{1}{4}} \geqslant \left(\frac{d}{2}\right)^{\frac{g(\gamma)}{8}d^{1/4}},$$

where we used

- $N/2 - \lceil N/4 \rceil \geqslant N/4 - 1$ because $\lceil x \rceil \leqslant x+1$.
- $2^{3/4}\frac{\lfloor 2^{3N/4} \rfloor}{2^{3N/4}} \geqslant 1$ if $N \geqslant 2$.
- $\frac{g(\gamma)}{4}2^{N/4} - \frac{1}{4} \geqslant \frac{g(\gamma)}{8}2^{N/4}$ if $N \geqslant 8$ and $\gamma \geqslant \sqrt{3/4}$.

Since the condition given in the statement of the lemma is

$$n+1 \leqslant \left(\frac{d}{2}\right)^{\frac{g(\gamma)}{8}d^{1/4}} \leqslant |\mathcal{C}|,$$

the packing $\mathcal{C}$ satisfies $|\mathcal{C}| \geqslant n+1$ and $d_{\min}(\mathcal{C}) \geqslant \sqrt{m - g(\gamma)^2}$. By Lemma 5.3.6 there exists a subspace $A \in G_{m,d}(\mathbb{R})$ of dimension $m = 2^{\lceil N/4 \rceil}$ s.t.

$$\forall x \in D, \qquad x \notin \overset{\triangle}{\mathcal{C}}_{\gamma}(A),$$

which concludes the proof of the lemma. □

5.3.3 Roadmap for the Rest of the Proof

The rest of this section provides the full details of the proof of Theorem 5.2.1. It proceeds in several steps.

First, in Section 5.3.4, we define the class of MDPs $\mathcal{M}$. The MDPs in the class share the same state space, action space, and feature map, but the instances differ in their transition and reward functions. Next, in Section 5.3.5, we specify the MDP instances, parameterised by a sequence of nested subspaces and a hidden direction w. The transition function is constructed so that, depending on which region of the action space is queried, information about w is progressively restricted to lower-dimensional subspaces. In Section 5.3.6, we show that Assumption 5.1.4 on the realizability of the linear representation is satisfied by all policies and MDPs in the class. Finally, in Section 5.3.7, we prove the lower bound itself. We combine all the components to construct an adversarial MDP instance based on the learner's queries and show that, unless the learner uses at least $\Omega(\log \log d)$ batches, it cannot distinguish between two MDPs with opposite optimal actions. As a result, any algorithm using fewer batches must incur constant error with non-trivial probability, completing the proof.

5.3.4 MDP Class

The BPI problem used in the proof of Theorem 5.2.1 is characterised by a class of MDPs $\mathcal{M}$. In this section, we define the class of MDPs $\mathcal{M}$. All MDPs $M \in \mathcal{M}$ in the class share the same state-space $\mathcal{S}$, action space $\mathcal{A}$ and feature map but differ in the transition function p_M and reward function r_M. This class of MDPs is similar to the one used by Zanette (2021) in the proof of their Theorem 3. Our construction differs in the transition functions which are defined in Section 5.3.5.

- **State-space:** $\mathcal{S} = \{\bar{s}\} \bigcup \mathcal{B}$ where $\bar{s}$ is the starting state disjoint from $\mathcal{B}$ (i.e. $\{\bar{s}\} \cap \mathcal{B} = \varnothing$).

- **Action-space:** Each state has a single action (which we denote by the state itself for convenience) other than the starting state $\bar{s}$ which can take actions in $\mathcal{B}$. Formally:

$$\mathcal{A}(s) = \begin{cases} \mathcal{B} & \text{if } s = \bar{s}. \\ \{s\} & \text{if } s \in \mathcal{B}. \end{cases}$$

  This notation enables that $\forall s \in \mathcal{S}, \mathcal{A}(s) \subset \mathcal{B}$.

- **Feature-map:** The feature map ϕ maps a state-action pair (s, a) to the action a, i.e. $\forall (s, a), \phi(s, a) = a$. Since $a \in \mathcal{B}$, the feature space is the unit-hypersphere $\mathcal{B}$ and $\|\phi(\cdot, \cdot)\|_2 \leqslant 1$ holds for all inputs.

5.3.5 Instance of the Class

Fix $K > 0$ and consider a sequence of K nested subspaces of $\mathbb{R}^d$:

$$B_K \subset B_{K-1} \subset ... \subset B_2 \subset B_1 \subset B_0 = \mathbb{R}^d.$$

s.t. $\dim B_K > 0$ and fix some $w \in B_K \cap \partial\mathcal{B}$. Set $B_{K+1} = \langle w \rangle$. Let $H_w = \{B_1, ..., B_K, B_{K+1}\}$ denote the set of nested subspaces (including $\langle w \rangle$). This is not defined as an ordered set, but the order can always be recovered since the sequence must be nested.

Every MDP $M \in \mathcal{M}$ is fully characterised by the sequence of nested subspaces H_w and a sign $+$ or $-$. Hence, they are denoted by $M_{H_w,+}$ or $M_{H_w,-}$.

Reward function: The reward function only depends on the vector w and the sign $\pm$ but we denote them with the same subscript as the MDP. Specifically it is defined as follows:

$$\text{on } M_{H_w,+}: \qquad r_{H_w,+}(s, a) = \begin{cases} 0, & \text{if } a \notin \mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w). \\ +(1-\gamma)a^T w, & \text{otherwise.} \end{cases}$$

$$\text{on } M_{H_w,-}: \qquad r_{H_w,-}(s, a) = \begin{cases} 0, & \text{if } a \notin \mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w). \\ -(1-\gamma)a^T w, & \text{otherwise.} \end{cases}$$

See Section 5.3.2 for the definition of hyper-spherical caps $\mathcal{C}_\gamma(w)$. We sometimes use a $w, \pm$ subscript to refer to both MDPs simultaneously.

Transition Function: The transition function for an MDP $M \in \mathcal{M}$ depends on the sequence of nested subspaces H_w but not on the sign $\pm$. The transition function is therefore the same for $M_{H_w,+}$ and $M_{H_w,-}$. If A is a subspace of $\mathbb{R}^d$, let $\text{proj}_A(x)$ denote the orthogonal projection of x onto A. See Section 5.3.2 for the definition of hyper-spherical caps $\mathcal{C}_\gamma(w)$ and sectors $\overset{\triangle}{\mathcal{C}}_\gamma(H)$. The successor state of a state-action pair (s,a) is deterministic and only depends on the chosen action so we will denote the unique successor state when taking action a by $s^+_{H_w}(a)$, which is defined as:

$$s^+_{H_w}(a) = \begin{cases} \frac{1}{\gamma}\text{proj}_{B_{k+1}}(a), & \text{if } a \in \overset{\triangle}{\mathcal{C}}_\gamma(B_k)\backslash\overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1}) \text{ for } k = 0, ..., K. \\ \frac{1}{\gamma}\text{proj}_{B_{K+1}}(a), & \text{if } a \in \overset{\triangle}{\mathcal{C}}_\gamma(B_{K+1})\backslash(\mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w)). \\ a, & \text{if } a \in \mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w). \end{cases}$$

Note that the starting state $\bar{s}$ is not a successor state so a policy trajectory can never return to $\bar{s}$. The transition function is defined for all $a \in \mathcal{B}$:

$$\Big[\bigcup_{k=0}^{K} \overset{\triangle}{\mathcal{C}}_\gamma(B_k)\backslash\overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1})\Big]\bigcup\Big[\overset{\triangle}{\mathcal{C}}_\gamma(B_{K+1})\backslash(\mathcal{C}_\gamma(w)\bigcup\mathcal{C}_\gamma(-w))\Big]\bigcup\Big[\mathcal{C}_\gamma(w)\bigcup\mathcal{C}_\gamma(-w)\Big] = \overset{\triangle}{\mathcal{C}}_\gamma(B_0) = \mathcal{B}.$$

and the pair-wise intersections are all empty (so they form a partition). The successor states are also well-defined:

- For $k = 0, ..., K$, if $a \in \overset{\triangle}{\mathcal{C}}_\gamma(B_k)\backslash\overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1})$, then $a \notin \overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1})$ and from Definition 5.3.2, this means that for any $u \in B_{k+1}$, $|u^T a|/\|u\|_2\|a\|_2 \leqslant \gamma$. Denoting $x = \text{proj}_{B_{k+1}}(a) \in B_{k+1}$,

$$\|x\|_2 = \frac{x^T x}{\|x\|_2} = \frac{|x^T a|}{\|x\|_2} \leqslant \gamma\|a\|_2 \leqslant \gamma,$$

  meaning that $\frac{1}{\gamma}\text{proj}_{B_{k+1}}(a) \in \mathcal{B}$.

- If $a \in \overset{\triangle}{\mathcal{C}}_\gamma(B_{K+1})\backslash(\mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w))$, , then $a \notin (\mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w))$ and from Definition 5.3.1, this means that $|w^T a| \leqslant \gamma$. Denoting $x = \text{proj}_{B_{K+1}}(a) \in B_{K+1}$,

$$\|x\|_2 = \frac{x^T x}{\|x\|_2} = \frac{|x^T a|}{\|x\|_2} \leqslant \gamma,$$

  meaning that $\frac{1}{\gamma}\text{proj}_{B_{K+1}}(a) \in \mathcal{B}$.

Crucially actions queried in $\overset{\triangle}{\mathcal{C}}_\gamma(B_k)\backslash\overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1})$ for $k \leqslant K$ may not reveal w. Without the knowledge of w, the reward function is unknown and even with the knowledge of w the reward

function is not fully identified, in which case $M_{H_w,+}$ and $M_{H_w,-}$ cannot be distinguished.

5.3.6 Realizability

We show that the action-value of *every* policy for any MDP $M \in \mathcal{M}$ can be linearly represented with the feature map defined in Section 5.3.4. The action-value of a policy π only depends on w and the sign $\pm$ of the reward function of the MDP, the sequence of nested subspace does not matter beyond w.

Lemma 5.3.7 (Realizability)**.** *For any $w \in \partial\mathcal{B}$ and sequence of nested subspaces H_w satisfying the construction from Section 5.3.5, let $Q^\pi_{H_w,+}$ and $Q^\pi_{H_w,-}$ be the action-value functions of an arbitrary policy π on $M_{H_w,+}$ and $M_{H_w,-}$, respectively. Then it holds that*

$$\forall (s,a), \qquad \begin{cases} Q^\pi_{H_w,+}(s,a) = +\phi(s,a)^T w & \text{on } M_{H_w,+}. \\ Q^\pi_{H_w,-}(s,a) = -\phi(s,a)^T w & \text{on } M_{H_w,-}. \end{cases}$$

Proof. Consider $M_{H_w,+}$ and set $Q(s,a) = \phi(s,a)^T w$. We will show that Q satisfies the Bellman evaluation equations for π at all state-actions pairs, which will imply that $Q(s,a) = Q^\pi_{H_w,+}(s,a)$. Apply $\mathcal{T}^\pi_{M_{H_w,+}}$ to Q at (s,a):

$$\mathcal{T}^\pi_{M_{H_w,+}}(Q)(s,a) = r_{H_w,+}(s,a) + \gamma Q(s^+_{H_w}(a), \pi(s^+_{H_w}(a))).$$

We consider the RHS of the above for all possible cases.

Case 1: If $a \in \overset{\triangle}{\mathcal{C}}_\gamma(B_k) \backslash \overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1})$ for some $k \in [0, ..., K]$, $s^+_{H_w}(a) = \frac{1}{\gamma}\text{proj}_{B_{k+1}}(a)$. Furthermore, π must return the only action available in the successor state (and the successor state is never $\bar{s}$), so $\pi(s^+_{H_w}(a)) = s^+_{H_w}(a)$ and we have:

$$\begin{aligned} r_{H_w,+}(s,a) + \gamma Q(s^+_{H_w}(a), \pi(s^+_{H_w}(a))) &= 0 + \gamma Q(s^+_{H_w}(a), s^+_{H_w}(a)) \\ &= \gamma \phi(s^+_{H_w}(a), s^+_{H_w}(a))^T w \\ &= \gamma s^+_{H_w}(a)^T w \\ &= \gamma \frac{1}{\gamma} \text{proj}^T_{B_{k+1}}(a) w \\ &= \text{proj}^T_{B_{k+1}}(a) w. \end{aligned}$$

However, recall that $w \in B_{k+1}$. Since $\text{proj}_{B_{k+1}}(a)$ is the **orthogonal** projection of a: $w^T \text{proj}_{B_{k+1}}(a) = w^T a$. Plugging into the above, we have: $\mathcal{T}^{\pi}_{M_{H_w,+}}(Q)(s,a) = a^T w = \phi(s,a)^T w = Q(s,a)$, which satisfies the Bellman evaluation equation.

Case 2: If $a \in \overset{\triangle}{\mathcal{C}}_\gamma(B_{K+1}) \backslash (\mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w))$, $s^+_{H_w}(a) = \frac{1}{\gamma}\text{proj}_{B_{K+1}}(a)$, and as before the policy can only take the only action available there, giving:

$$\begin{aligned}
r_{H_w,+}(s,a) + \gamma Q(s^+_{H_w}(a), \pi(s^+_{H_w}(a))) &= 0 + \gamma Q(s^+_{H_w}(a), s^+_{H_w}(a)) \\
&= \gamma \phi(s^+_{H_w}(a), s^+_{H_w}(a))^T w \\
&= \gamma s^+_{H_w}(a)^T w \\
&= \gamma \frac{1}{\gamma} \text{proj}^T_{B_{K+1}}(a) w \\
&= \text{proj}^T_{B_{K+1}}(a) w \\
&= a^T w = \phi(s,a)^T w = Q(s,a),
\end{aligned}$$

which satisfies the Bellman evaluation equation.

Case 3: If $a \in \mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w)$, the reward is no longer 0 and $s^+_{H_w}(a) = a$, and again as before the policy can only take the only action available there so we have:

$$\begin{aligned}
r_{H_w,+}(s,a) + \gamma Q(s^+_{H_w}(a), \pi(s^+_{H_w}(a))) &= +(1-\gamma) a^T w + \gamma Q(a,a) \\
&= (1-\gamma) a^T w + \gamma \phi(a,a)^T w \\
&= (1-\gamma) a^T w + \gamma a^T w \\
&= a^T w = \phi(s,a)^T w = Q(s,a),
\end{aligned}$$

which satisfies the Bellman evaluation equation.

For all cases, Q satisfies the Bellman evaluation equations, so it is the fixed point of the Bellman evaluation operator. In particular, it is the action-value of the policy π on $M_{H_w,+}$. The argument is identical for $M_{H_w,-}$ with $Q(s,a) = -\phi(s,a)^T w$. □

5.3.7 Proof of Theorem 5.2.1

Proof overview. The proof proceeds via an adversarial construction. We consider two cases depending on whether the total number of queries exceeds a threshold W. If $n > W$, we show directly that the number of batches must satisfy $K = \Omega(\log \log d)$. Otherwise, when $n \leqslant W$, we adaptively construct an MDP instance using a sequence of nested subspaces. We show that after K batches the learner cannot distinguish between two MDPs with opposite optimal actions, implying a constant error unless $K = \Omega(\log \log d)$.

Consider the MDP class $\mathcal{M}$ defined in Sections 5.3.4 and 5.3.5. By Lemma 5.3.7, all instances in $\mathcal{M}$ satisfy Assumption 5.1.4: for every policy π, the corresponding action-value function Q^π is realizable with respect to the feature map $\phi(\cdot, \cdot)$.

We construct a specific MDP $M \in \mathcal{M}$ depending on the learner's queries. Fix $K > 0$, and let n_k denote the number of queries made at round k, with total number of queries

$$n = \sum_{k=1}^{K} n_k.$$

Let A_k be the set of actions queried at round k, and define

$$\bar{A}_k = \bigcup_{i=1}^{k} A_i$$

to be the set of all actions queried up to round k.

Sample-efficiency assumption. Assume the learner is sample-efficient, so that n is polynomial in d: there exist constants $\alpha > 0$ and $T \in \mathbb{N}$ such that $n \leqslant \alpha d^T$. Let $N \in \mathbb{N}$ satisfy $2^N \leqslant d < 2^{N+1}$ and define $d_+ = 2^N \geqslant d/2$. Set

$$W = \exp\Big(\frac{1}{8} g(\gamma) d_+^{1/4^K}\Big), \qquad g(\gamma) = 2\gamma^2 - 1.$$

We now distinguish two cases.

Case 1: $W < n$**.** In this case we show that sample efficiency alone forces the number of batches K to scale as $\Omega(\log\log d)$. Indeed,

$$\begin{aligned} W < n &\implies W < \alpha d^T \\ &\implies \frac{g(\gamma)}{8} d_+^{1/4^K} < \log(\alpha d^T) \\ &\implies \frac{g(\gamma)}{8} (d/2)^{1/4^K} < \log(\alpha d^T) \\ &\implies (d/2)^{1/4^K} < \frac{8}{g(\gamma)} \log(\alpha d^T) \\ &\implies \frac{1}{4^K} \log(d/2) < \log\Big(\frac{8}{g(\gamma)} \log(\alpha d^T)\Big) \\ &\implies \frac{1}{\log 4} \log\left[\frac{\log(d/2)}{\log\left(\frac{8}{g(\gamma)} \log(\alpha d^T)\right)}\right] < K. \end{aligned}$$

For sufficiently large d, this implies $K \geqslant c' \log\log d$ for some constant $c' > 0$.

Case 2: $n \leqslant W$**.** In this case we show that even with K adaptive batches, the learner cannot identify the optimal action. The key step is to construct a sequence of nested subspaces that hide the optimal direction from all queried actions.

Since $n \leqslant W$, we have

$$n_k \leqslant \exp\Big(\frac{1}{8} g(\gamma) d_+^{1/4^k}\Big) \quad \text{for all } k = 1, \dots, K.$$

Claim (existence of nested subspaces). There exist nested subspaces

$$B_K \subset \cdots \subset B_1 \subset \mathbb{R}^d$$

and a vector $w \in B_K$ such that, for every $k = 1, \dots, K$,

$$\dim B_k = 2^{\lceil N/4^k \rceil} \geqslant 2^8, \qquad \bar{A}_k \cap \overset{\triangle}{\mathcal{C}}_\gamma(B_k) = \varnothing.$$

Proof of the claim. We proceed by induction on k.

Base case $(k = 1)$**.** At the first round, the learner queries $A_1 = \bar{A}_1$ with

$$|A_1| = n_1 \leqslant \exp\Big(\tfrac{1}{8} g(\gamma) d_+^{1/4}\Big) \leqslant \Big(\tfrac{d_+}{2}\Big)^{\frac{1}{8} g(\gamma) d_+^{1/4}} - 1,$$

where the final inequality holds for $d_+ \geqslant 12$. By Lemma 5.3.5, there exists a subspace B_1 of dimension $2^{\lceil N/4 \rceil}$ such that

$$A_1 \cap \overset{\triangle}{\mathcal{C}}_\gamma(B_1) = \varnothing.$$

What the learner observes at round 1. The learner observes the feedback

$$\{(r_{w,\pm}(s,a), s^+_{H_w}(a)) : a \in A_1\}.$$

Although w and future subspaces have not yet been fixed, the feedback depends only on B_1. Indeed, for all $a \in \bar{A}_1$,

$$a \notin \overset{\triangle}{\mathcal{C}}_\gamma(B_1) \implies s^+_{H_w}(a) = \tfrac{1}{\gamma}\mathrm{proj}_{B_1}(a), \qquad r_{w,\pm}(s,a) = 0.$$

Thus no information about w or future subspaces is revealed at this stage.

Induction hypothesis. Assume that nested subspaces $B_1, \ldots, B_k$ have been constructed such that

$$\dim B_i = 2^{\lceil N/4^i \rceil} \geqslant 2^8, \qquad \bar{A}_i \cap \overset{\triangle}{\mathcal{C}}_\gamma(B_i) = \varnothing \quad \text{for all } i \leqslant k.$$

Inductive step. The feedback observed from $\bar{A}_k$ depends only on $B_1, \ldots, B_k$: for all $a \in A_k$,

$$a \notin \overset{\triangle}{\mathcal{C}}_\gamma(B_k) \implies s^+_{H_w}(a) = \tfrac{1}{\gamma}\mathrm{proj}_{B_j}(a) \text{ for some } j \leqslant k, \qquad r_{w,\pm}(s,a) = 0.$$

Thus the learner has only observed projections of queried actions onto $B_1, \ldots, B_k$, and the only constraint on the MDP so far is that $w \in B_k$.

Let $d_k = \dim B_k = 2^{\lceil N/4^k \rceil}$. The learner selects A_{k+1} with

$$|A_{k+1}| = n_{k+1} \leqslant \exp\left(\tfrac{1}{8}g(\gamma)d_+^{1/4^{k+1}}\right) \leqslant \left(\tfrac{d_k}{2}\right)^{\frac{1}{8}g(\gamma)d_k^{1/4}} - 1.$$

Applying Lemma 5.3.5, there exists a subspace $B_{k+1} \subset B_k$ of dimension $2^{\lceil N/4^{k+1} \rceil}$ such that

$$A_{k+1} \cap \overset{\triangle}{\mathcal{C}}_\gamma(B_{k+1}) = \varnothing.$$

As before, the feedback observed at round $k+1$ depends only on $B_1, \ldots, B_{k+1}$.

Verification of dimensional conditions. To apply Lemma 5.3.5, we require $\dim B_k \geqslant 2^8$. Since $d_k \geqslant (d/2)^{1/4^k}$, this holds provided

$$(d/2)^{1/4^k} \geqslant 2^8 \iff k \leqslant c \log\log d$$

for a suitable constant $c > 0$ and sufficiently large d. If this condition fails, then

$$K > c \log\log d. \tag{5.1}$$

Otherwise, the induction goes through, establishing the claim.

End of proof of the claim.

Application of Lemma 5.3.5 in lower-dimensional ambient spaces. We briefly justify the repeated application of Lemma 5.3.5 within the subspaces B_k. Although the lemma is stated for subspaces of $\mathbb{R}^d$, any d_k-dimensional subspace of $\mathbb{R}^d$ is isomorphic to $\mathbb{R}^{d_k}$. Thus, at round k, we may identify B_k with $\mathbb{R}^{d_k}$, project the queried actions in A_{k+1} into B_k, and apply Lemma 5.3.5 in this ambient space to obtain a subspace $H \subset B_k$. This subspace can then be viewed naturally as a subspace of $\mathbb{R}^d$. The same argument applies in the base case, where d_+ plays the role of the ambient dimension.

Specification of the MDP. Let $w \in B_K \cap \partial\mathcal{B}$ and define $B_{K+1} = \langle w \rangle$. Let $H_w = \{B_1, \ldots, B_K, B_{K+1}\}$ and consider the MDPs $M_{H_w,+}$ and $M_{H_w,-}$. For all queried actions $a \in \bar{A}_K$,

$$a \notin \overset{\triangle}{\mathcal{C}}_\gamma(B_K) \implies a \notin \mathcal{C}_\gamma(w) \cup \mathcal{C}_\gamma(-w),$$

so the observed reward is zero. Hence the learner cannot distinguish between $M_{H_w,+}$ and $M_{H_w,-}$.

Lower bound on the learner's error. The learner must output a near-optimal policy at the initial state $\bar{s}$. However,

$$V^\star_{M_{H_w,+}}(\bar{s}) = Q^\star_{M_{H_w,+}}(\bar{s}, w) = \phi(\bar{s}, w)^\top w = 1,$$
$$V^\star_{M_{H_w,-}}(\bar{s}) = Q^\star_{M_{H_w,-}}(\bar{s}, -w) = -\phi(\bar{s}, -w)^\top w = 1.$$

If the learner outputs an action a with $a^\top w > 0$, it incurs error greater than 1 on $M_{H_w,-}$, and similarly for $M_{H_w,+}$ if it outputs an action a with $a^\top w \leqslant 0$. Even if the learner randomizes, with probability at least $1/2$ it incurs error at least 1 on one of the two MDPs.

Conclusion. To be more than $(1, 1/2)$-sound, the learner must either be in Case 1 or satisfy condition (5.1) in Case 2. In both cases,

$$K = \Omega(\log \log d),$$

which completes the proof. □

5.3.8 MDP Construction Illustration

In this section, we provide a high-level illustration of how the hard instance $M \in \mathcal{M}$ is constructed when $d = 3$ and the learner makes three queries per round. The reward vector of the MDP is 0 everywhere except in the γ-hyper-spherical sector $\overset{\triangle}{\mathcal{C}}_\gamma(w)$ (see Section 5.3.2) of a vector $w \in \mathcal{B}$, which is unknown to the learner. The MDP is constructed in such a way that $Q^\star(\bar{s}, a) = \pm\phi(s, a)^T w = \pm a^T w$. If the learner can identify w and make a query in its γ hyper-spherical sector (to distinguish between the + and the −), then it can fully solve the BPI problem. Therefore the aim of the learner in choosing its queries is to identify this vector w.

Round 1 (Figure 5.3).

- A: Three queries $\phi_i = \phi(s_i, a_i) = a_i$ $(i = 1, 2, 3)$ are made by the learner.
- B: The three queries with their hyper-spherical sectors $\overset{\triangle}{\mathcal{C}}_\gamma(\phi_i)$ $(i = 1, 2, 3)$ (we only show one side of the sector for clarity).
- C: There exists a 2-dimensional subspace B_1 that does not intersect the hyper-spherical sectors of the queries which the environment can use to further hide w - if B_1 does not intersect the γ hyper-spherical sectors of the queries, then the queries ϕ_i will not be in the γ hyper-spherical sector of any point in B_1 (in particular w and cannot identify the + or − in the reward).
- D: The transition function and successor states are constructed in such a way that realizability is satisfied with $w \in B_1$. In particular, $s_i^+ = \frac{1}{\gamma}\text{proj}_{B_1}(a_i)$ is valid because B_1 does

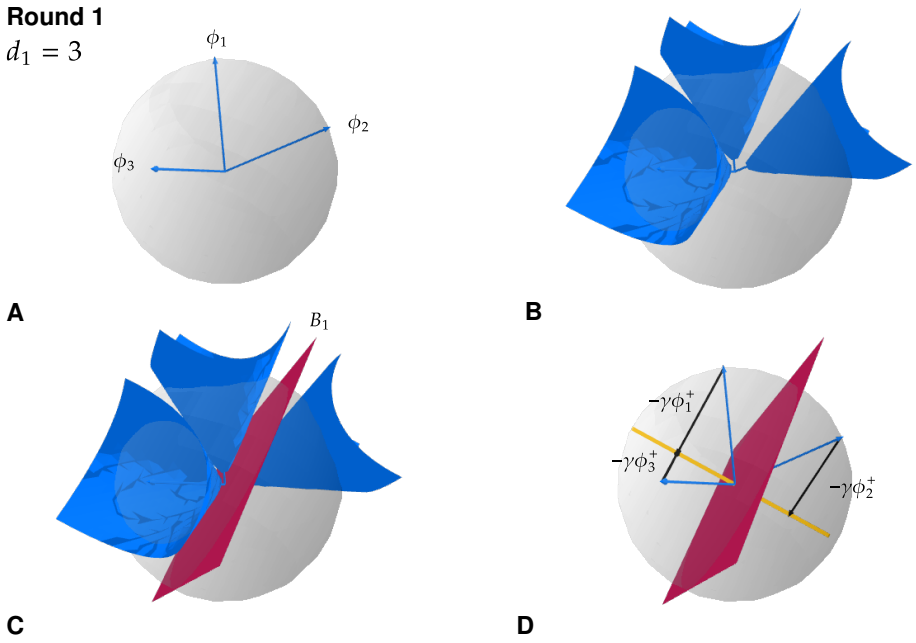

Figure 5.3: Round 1.

not intersect the sectors of the queries (see Section 5.3.5). This choice allows the component of ϕ in the direction of w to grow by $1/\gamma$ in ϕ^+, this will cancel with the γ discount in the bellman equation and is a key step in satisfying realizability. It also does not reveal any additional information about w other than $w \in B_1$. In addition, w or the actions taken by the policy within B_1 do not need to be specified yet and can depend on queries made by the learner in later rounds.

Since $\phi_i - \gamma\phi_i^+ = a_i - \text{proj}_{B_1}(a_i)$ is orthogonal to B_1, this construction admits the interpretation of erasing information along the directions of the subspace B_1. Hence, this provides the intuition discussed in Section 5.3.1 but is in fact exactly what is required to hide the w vector within the subspace B_1.

Round 2 (Figure 5.4). Without loss of generality, we assume the queries made by the learner are within B_1 (the MDP is fully identified outside of B_1 so the learner can gain no further information from queries outside of B_1) and only visualise B_1 as a circle embedded in the sphere from Figure 5.3.

- E: Three queries $\phi_i = \phi(s_i, a_i) = a_i$ $(i = 4, 5, 6)$ are made by the learner.

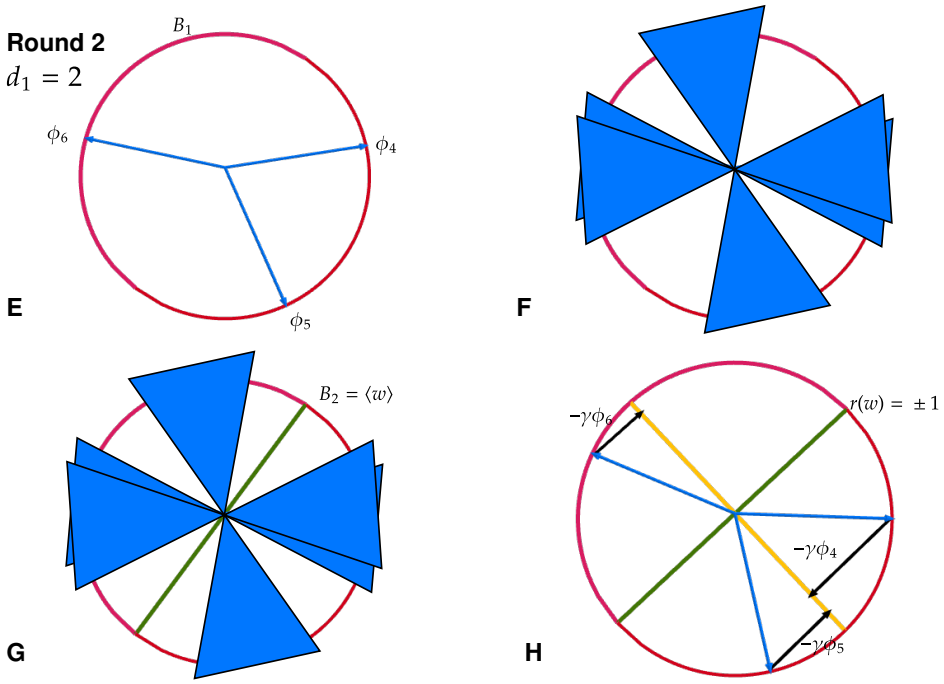

Figure 5.4: Round 2.

- F: The three queries with their (2-dimensional) hyper-spherical sectors $\overset{\triangle}{\mathcal{C}}_\gamma(\phi_i)$ $(i = 4, 5, 6)$.
- G: There exists a 1-dimensional subspace $B_2 = \langle w \rangle$ that does not intersect the hyper-spherical sectors of the queries. As in C above, if $\langle w \rangle$ does not intersect the γ hyper-spherical sectors of the queries, then the queries ϕ_i will not be in the γ hyper-spherical sector of w.
- H: Similar to D, the transition function and successor states are constructed within B_1 in such a way that realizability is satisfied. In particular, $s_i^+ = \frac{1}{\gamma}\text{proj}_{B_2}(a_i) = \frac{1}{\gamma}(w^T a_i)w$ is valid because $B_2 = \langle w \rangle$ does not intersect the sectors of the queries. The reward inside $\overset{\triangle}{\mathcal{C}}_\gamma(w)$ is characterised by a sign $+$ or $-$. No queries have been made inside $\overset{\triangle}{\mathcal{C}}_\gamma(w)$ and so a third round is required to identify the sign $+$ or $-$.

This illustration uses three queries per round. In the proof, we formally show that if the number of queries in the first round is polynomial in d, then there always exists a subspace of dimension $d^{1/4}$ that does not intersect the sectors of the queries in the first round. We make a similar argument for later rounds to show $\log \log d$ rounds are required to identify the vector w.

5.4 Related Works

The works discussed below are for infinite-horizon discounted MDPs unless stated otherwise.

Tabular MDPs ($|\mathcal{A}|, |\mathcal{S}|$ are "small") can be solved in the offline ($K = 1$) setting: model-based approaches (Li et al., 2020; Agarwal et al., 2020) under a generative model are minimax-optimal (Azar et al., 2013) with sample-complexity linear in the dimension of the MDP $|\mathcal{A}| \times |\mathcal{S}|$. These methods estimate the MDP by sampling equally from all state-action pair and then use dynamic programming approaches on the estimated MDP. In particular, all the samples are drawn in a single-batch. Beyond the generative model and under a form of queries produced along a policy trajectory, tabular offline RL is no longer sample efficient (Xiao et al., 2022) and requires a number of samples exponential in $|\mathcal{S}|$ or $1/(1-\gamma)$. Adaptivity ($K > 1$) is required to go beyond these exponential lower bounds.

Linear Function Approximation: In the offline setting, there are lower-bounds showing BPI is not possible with samples polynomial in the effective horizon $1/(1-\gamma)$ or the linear dimension (Amortila et al., 2020; Chen et al., 2021a; Zanette, 2021). The bound of Zanette (2021) is the strongest as it holds for any data-distribution. These exponential lower-bounds can be overcome with assumptions such as low-distribution shift (Chen et al., 2021a), low inherent Bellman-error (Xie and Jiang, 2020; Duan et al., 2020) or low local inherent Bellman error (Zanette, 2023). We refer the reader to the work of Zanette (2021) for a more in depth discussion of offline RL. In the fully-adaptive setting, there are sample-efficient algorithms under all-policy realizability (Lattimore et al., 2020), under $V^\star$-realizability only (Weisz et al., 2021) (if the action space is finite) and under linear MDPs (Taupin et al., 2023; Kitamura et al., 2023).

The multi-batch learning model has been studied extensively for bandit algorithms (Perchet et al., 2015; Jun et al., 2016; Gao et al., 2019; Esfandiari et al., 2021; Duchi et al., 2018; Han et al., 2020; Ruan et al., 2021). In RL, it has been studied in the regret-minimisation setting for finite-horizon tabular (Zihan et al., 2022) and linear MDPs (Wang et al., 2021) and MDPs under general function approximation (Xiong et al., 2023). A closely related notion is deployment efficiency (Matsushima et al., 2021), which constrains batches to be of a fixed size consisting of trajectories from a single policy. In finite-horizon linear MDPs, it has been shown that BPI can be solved to arbitrary accuracy with a number of deployments independent of the dimension d (Huang et al., 2022; Qiao and Wang, 2023) where the deployed policy is a finite mixture of deter-

ministic policies. Our results suggest that infinite-horizon discounted MDPs under more general linear representation of action-values are fundamentally harder since the number of deployments must scale with dimension.

Low-Switching Cost: Limited adaptivity in RL has mostly been studied in the context of regret-minimisation algorithms with low-switching cost for finite-horizon episodic MDPs, i.e. minimising the number of times the policy used changes from one episode to the next. These works are not directly comparable because they study regret-minimisation for finite-horizon MDPs and we study BPI in the discounted setting. Nevertheless, there are works studying switching costs in tabular MDPs (Qiao et al., 2022; Bai et al., 2019; Zhang et al., 2020b), linear MDPs (Gao et al., 2021; Wang et al., 2021; Qiao and Wang, 2023) and MDPs with a linear representation for the action values (Qiao et al., 2023).

The policy finetuning setting assumes access to an offline dataset that can be complemented with online trajectories (Xie et al., 2021) but is different from our setting since there is no adaptivity constraint in the online algorithm, i.e. once the initial dataset has been collected, the query selection strategy can be updated after each new observation (or episode in the episodic setting). However, if the additional trajectories are collected using a non-adaptive policy instead of an online algorithm, we can recover our setting with $K = 2$ batches. This is studied by Zhang and Zanette (2023) who show that for finite-horizon tabular MDPs, $K = 2$ is enough to solve the BPI problem to arbitrary accuracy. Our results rule out achieving a similar result for infinite-horizon discounted MDPs under policy-free queries and linear function approximation.

5.5 Conclusion

In this chapter, we have studied the connection between adaptivity and sample-efficiency for RL algorithms finding near-optimal policies under d-dimensional linear function approximation. For multi-batch learning, we have established $\Omega(\log \log d)$ lower-bounds on the number of batches K needed to solve the RL problems sample-efficiently (number of queries polynomial in d). In particular, having adaptivity ($K > 1$) does not guarantee sample-efficiency. Consequently the boundary of sample efficiency must not lie between batch RL ($K = 1$) and adaptive RL ($K > 1$) but rather within a regime of adaptivity scaling with dimension. These insights contribute to a deeper understanding of the fundamental differences between high-dimensional and low-dimensional in sample-efficient RL algorithms with linear function approximation.

Several open questions remain. In particular, it is unclear whether the optimal dependence should be $\log\log d$, $\log d$ or somewhere in between. More broadly, it would also be interesting to understand how this dependence changes under alternative assumptions or problem formulations, and in particular which structural properties of the MDP may permit sample-efficient learning with fewer, or even dimension-independent, batches, thereby narrowing the gap between low and high-dimensional problems.

5.6 Missing Details

5.6.1 Proof of Lemma 5.3.6

The proof of Lemma 5.3.6 itself relies on the following lemma.

Lemma 5.6.1. *Consider $(n+1)$ subspaces $A_1, ..., A_{n+1}$ of dimension m s.t. for any $i \neq j$,*

$$\max_{x \in A_i, z \in A_j} \frac{x^T z}{\|x\|_2 \|z\|_2} < g(\gamma).$$

Given n vectors $y_1, ..., y_n \in \mathcal{B}$ (w.l.o.g. all unit norm), then there is a subspace $H \in \{A_1, ..., A_{n+1}\}$ s.t. for all $y \in \{y_1, ..., y_n\}$,

$$\max_{w \in H} \frac{y^T w}{\|w\|_2} \leqslant \gamma,$$

meaning $y_i \notin \overset{\triangle}{\mathcal{C}}_\gamma(H)$ for $i = 1, ..., n$.

This lemma is similar to Lemma 5.3.6 but uses a more explicit notion of distance between subspaces.

Proof of Lemma 5.6.1. Recall that we consider $(n+1)$ subspaces $A_1, ..., A_{n+1}$ of dimension m s.t. for any $i \neq j$,

$$\max_{x \in A_i, z \in A_j} \frac{x^T z}{\|x\|_2 \|z\|_2} < g(\gamma) = 2\gamma^2 - 1. \tag{5.2}$$

Identify each y_i to the subspace $A_{h(i)}$ that is closest to y_i in terms of inner product or angle, formally

$$h(i) = \arg\max_j \max_{u \in A_j} y_i^T u / \|u\|_2.$$

Since there are n vectors y_i, there will be at least one subspace in $\{A_1, ..., A_{n+1}\}$ that is not associated to any of the y_is. Call this subspace H.

We show that for any $y \in \{y_i\}_{i=1}^n$**,** $\max_{w \in H} \frac{y^T w}{\|w\|_2} \leqslant \gamma$.

Fix $y \in \{y_i\}_{i=1}^n$ and let A be its corresponding subspace $A_{h(i)}$. Let

$$w = \text{argmax}_{w' \in H} \frac{y^T w'}{\|w'\|_2}.$$
$$a = \text{argmax}_{a' \in A} \frac{y^T a'}{\|a'\|_2}.$$

Assume both w and a are of unit norm w.l.o.g. We know:

- 1. $w^T y \leqslant a^T y$ from the definition of $h(i)$.
- 2. $w^T a < g(\gamma)$ from (5.2).

Let $x = \frac{w+a}{\|w+a\|_2}$ be the average of w and a, with $\|w + a\|_2^2 = w^T w + 2a^T w + a^T a = 2 + 2a^T w = 2(1 + a^T w)$.

Claim: $w^T y \leqslant w^T x$ (intuition: y is "closer" to a than w, so w should be "closer" to average of a and w than to y).

Proof of claim: Assume $w^T y > w^T x$. From 1. this implies $a^T y > w^T x$. Now

$$w^T x = w^T \frac{w + a}{\|w + a\|_2} = \frac{w^T w + w^T a}{\|w + a\|_2} = \frac{1 + w^T a}{\|w + a\|_2} = \frac{a^T a + w^T a}{\|w + a\|_2} = a^T x.$$

So we have $a^T y > a^T x \implies a^T(y - x) > 0$. From the initial assumption we also have $w^T(y - x) > 0$. Combining:

$$\begin{aligned}(w + a)^T(y - x) > 0 &\implies x^T(y - x) > 0 \\ &\implies x^T y - x^T x > 0 \\ &\implies x^T y - 1 > 0 \\ &\implies x^T y > 1,\end{aligned}$$

which is a contradiction since both x and y are of unit-norm. **End of proof of claim.**

We now show $w^T x \leqslant \gamma$. Using 2.,

$$w^T x = \frac{1 + w^T a}{\|w + a\|_2} = \frac{\sqrt{1 + a^T w}}{\sqrt{2}} \leqslant \frac{\sqrt{1 + g(\gamma)}}{\sqrt{2}} = \frac{\sqrt{1 + 2\gamma^2 - 1}}{\sqrt{2}} = \frac{\sqrt{2\gamma^2}}{\sqrt{2}} = \gamma.$$

Combining with the claim we have that $w^T y \leqslant \gamma$. Given the definition of w and that y was

arbitrary, we have shown

$$\arg\max_i \max_{w \in H} \frac{y_i^T w}{\|w\|_2} \leqslant \gamma,$$

which concludes the proof. □

We can now provide the proof of Lemma 5.3.6.

Proof of Lemma 5.3.6. Consider distinct $A, B \in \mathcal{C}$, i.e A, B are subspaces of dimension m s.t.

$$d_C(A, B) > \sqrt{m - g(\gamma)^2}.$$

Now letting $0 \leqslant \theta_1 \leqslant ... \leqslant \theta_m \leqslant \pi/2$ be the principal angles between A and B (see Section 5.3.2), we have

$$d_C(H, A) = \sqrt{\sin^2(\theta_1) + ... + \sin^2(\theta_m)} \leqslant \sqrt{m - 1 + \sin^2 \theta_1}.$$

Combining with the inequality above,

$$\sqrt{m - g(\gamma)^2} \leqslant \sqrt{m - 1 + \sin^2 \theta_1} \implies \sin^2(\theta_1) > 1 - g(\gamma)^2.$$

The first principal angle $\theta_1 = \arccos a^T b$ where $a \in A$, $b \in B$ are unit vectors chosen s.t.

$$a^T b = \max_{x \in A, y \in B} \frac{x^T y}{\|x\|_2 \|y\|_2}.$$

So we have

$$\max_{x \in A, y \in B} \frac{x^T y}{\|x\|_2 \|y\|_2} = a^T b = \cos \theta_1 = \sqrt{1 - \sin^2(\theta_1)} < g(\gamma).$$

Since this applies to arbitrary distinct $A, B \in \mathcal{C}$, there are $(n+1)$ subspaces $A_1, ..., A_{n+1} \in \mathcal{C}$ of dimension m s.t. for any $i \neq j$,

$$\max_{x \in A_i, z \in A_j} \frac{x^T z}{\|x\|_2 \|z\|_2} < g(\gamma) = 2\gamma^2 - 1.$$

By Lemma 5.6.1, there exists a subspace H in $\{A_1, ..., A_{n+1}\}$ s.t. for all $y \in \{y_1, ..., y_n\}$,

$$\max_{w \in H} \frac{y^T w}{\|w\|_2} \leqslant \gamma,$$

i.e. $y_i \notin \overset{\triangle}{\mathcal{C}}_\gamma(H)$ for $i = 1, ..., n$, which concludes the proof. □

Part II

Exploiting Sparsity

Chapter 6

Stochastic Shortest Path with Sparse Adversarial Costs

The Stochastic Shortest Path (SSP) problem is a fundamental model in reinforcement learning (Bertsekas and Tsitsiklis, 1991; Tarbouriech et al., 2020), which describes tasks where an agent interacts with an environment over episodes and must reach a designated goal state within each episode while minimizing accumulated costs. This covers problems such as car navigation while trying to avoid traffic jams, or internet routing. This classical setting has recently been extended to the adversarial regime, where costs may vary arbitrarily between episodes (Rosenberg and Mansour, 2021; Chen et al., 2021c; Chen and Luo, 2021; Zhao et al., 2022) and the goal is to obtain theoretical guarantees robust to any cost-generation mechanism. Under *full-information* feedback where the full cost vector is observed after each episode and known transitions, current algorithms (Rosenberg and Mansour, 2021; Chen et al., 2021c) achieve regret bounds that scale as $\mathcal{O}\left(\sqrt{DKT_\star \log SAT_\star}\right)$, where D is the diameter (the smallest expected time to reach the goal state of any policy from any state), $T_\star$ is the expected time to reach the goal state of the optimal policy, S is the number of states, A is the number of actions, and K is the number of episodes. These bounds are independent of any cost structure and are shown to be minimax optimal up to logarithmic factors in Chen et al. (2021c).

In SSP, the dimension of the problem is the size of the state-action space SA and appears in the minimax regret as $\sqrt{\log SA}$. While in the worst-case this is unimprovable (we show this in Theorem 6.3.1), many real-world high-dimensional problems have costs with structural proper-

ties that may be leveraged for improved regret. A common property often considered in the statistics and machine learning literature (Wainwright, 2019b) is sparsity, which can naturally arise for SSP problems. For instance, in the car navigation example, the number of traffic jams is usually much smaller than the number of roads. Motivated by this, we consider *sparsity* in high-dimensional SSP problems where M, the maximum number of state-action pairs with non-zero cost in an episode, can be much smaller than the dimension SA.

In such scenarios, the regret bounds should capture some dependence on M, reflecting an improvement in performance on easier sparse problems as $M \to 1$ and recovering the standard bounds on worst-case problems as $M \to SA$. In fact, for the so-called experts setting[1] ($S = 1$), the minimax regret scales with $\sqrt{KMA^{-1} \log A}$ instead of $\sqrt{K \log A}$ in the worst-case (Kwon and Perchet, 2016), providing a polynomial improvement in the dimension A. Furthermore, this is achieved with the gradient-based Online Mirror Descent (OMD) algorithm (see Chapter 2) with negative entropy regularization. However for SSP where the problem dimension also includes the size of the state space S, we show that existing approaches also based on OMD with the same negative entropy regularization (Chen et al., 2021c) fail to exploit sparse costs. We construct a sparse SSP problem where this algorithm suffers $\sqrt{\log S}$ regret (Theorem 6.3.1), providing no improvement in terms of S compared to the minimax regret for the non-sparse worst-case $M = SA$ problem. The failure of existing gradient-based SSP methods to exploit sparsity, let alone match the polynomial improvements from the experts setting, leads us to ask the following questions:

Can we effectively exploit sparsity in full-information feedback high-dimensional SSP problems (with gradient-based methods)?

We answer this question positively for the known-transition case by designing a family of regularisers based on ℓ_r-norms for $r \in (1, 2)$ for which when used with OMD we show a $\mathcal{O}(\sqrt{DKT_\star \log MT_\star})$ regret bound that depends logarithmically on the sparsity level M, rather than on the size of the state-action space SA. This family of regularisers interpolates between the negative entropy and the squared Euclidean norm (see Section 6.4), allowing flexibility for much weaker regularization on sparse points in sparse settings and recovering existing algorithms (and guarantees) in the non-sparse setting.

[1]The experts setting is the single-state full-information feedback A-action online learning problem (Cesa-Bianchi et al., 1997b; Freund and Schapire, 1997).

We show that the $\sqrt{\log M}$ dependence on M is unimprovable by constructing a matching lower bound. This establishes that the benefit of sparsity in SSP is logarithmic in SA instead of polynomial, as in the simpler experts problem. It also highlights that M plays the role of effective dimension, replacing the general dimension SA in controlling the scaling of the minimax regret.

Our results demonstrate that sparsity can effectively and optimally be exploited by gradient-based methods in high-dimensional full-information feedback SSP problems to obtain dimension-independent regret bounds that only depend on the sparsity level.

Contributions

- We make use of a family of ℓ_r-norm ($r \in (1, 2)$) regularisers for OMD that allows interpolation between the negative entropy and squared Euclidean norm, adjusting its geometry to the sparsity of the cost functions (see Section 6.4). The regulariser naturally plugs into the standard OMD analysis.
- We show OMD with the above regulariser achieves a regret bound of order $\mathcal{O}\left(\sqrt{DKT_\star \log MT_\star}\right)$ (Theorem 6.4.1) improving upon $\mathcal{O}\left(\sqrt{DKT_\star \log SAT_\star}\right)$ from Chen et al. (2021c).
- We establish a lower bound of order $\Omega\left(\sqrt{DKT_\star \log M}\right)$ (Theorem 6.4.2), matching our regret guarantees up to a logarithmic factor of $T_\star$ (already present in prior work Chen et al. (2021c)) and improving over Chen et al. (2021c) in the $M = SA$ non-sparse setting by including the $\sqrt{\log SA}$ dependence. For this result, we derive a result on the expectation of the maximum of asymmetric zero-mean random walks, generalizing the result for the symmetric case from Orabona and Pal (2015).
- We show that OMD with the negative entropy used in prior work Rosenberg and Mansour (2021); Chen et al. (2021c) suffers regret at least $\Omega(\sqrt{K \log S})$ even when $M = 3$ (Theorem 6.3.1). This rules out the negative entropy as a viable regulariser in the sparse setting and provides justification for the use of our regulariser. This lower-bound relies on the careful design of an MDP with skewed initial occupancy measures that highlights both the reasons for the failure of the negative entropy as well as the more general difficulty of the stochastic nature of SSP problems.

Setting	Performance Criteria	Main Theorem(s)	Publication Source
SSP (known transitions + full-information)	Regret Minimisation	Theorem 6.4.1, Theorem 6.4.2	Johnson et al. (2025b)

Table 6.1: Chapter overview.

Roadmap. In Section 6.1, we define the problem setting. In Section 6.2, we explain how the problem can be reduced to online linear optimisation for which we can use the gradient-based machinery such as OMD presented in Chapter 2. In Section 6.3, we show how the approach from prior work based on OMD with negative entropy regularisation fails to exploit sparsity. In Section 6.4, we present upper and lower bounds that establish how to optimally solve sparse adversarial stochastic shortest path problems under full-information feedback. In Section 6.5, we discuss results from related works. In Section 6.6, we conclude and discuss future-work. Finally, in Section 6.7, we provide missing details and proofs omitted earlier in the chapter.

Bibliographic remark: This chapter is based on Johnson et al. (2025b). It was done jointly with Alberto Rumi; I worked on all aspects of the work except the unknown transition lower bound. It is licensed under a Creative Commons Attribution 4.0 International License.

6.1 Problem Setting

We consider the *Stochastic Shortest Path* (SSP) problem with adversarial costs. The environment is modelled as a slight variation of Markov Decision Processes (MDP) presented in Chapter 3. Formally it is specified by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, s_0, g)$ along with a sequence of cost functions $\{c_k\}_{k=1}^K$ chosen by an oblivious adversary over K episodes. $\mathcal{S}$ is the state space with cardinality $S = |\mathcal{S}|$, and $s_0 \in \mathcal{S}$ is the fixed starting state. The goal state g is a special absorbing state not included in $\mathcal{S}$. $\mathcal{A}$ is the action space with cardinality $A = |\mathcal{A}|$ and we assume for simplicity that it is the same in every state. Let $\Gamma = \mathcal{S} \times \mathcal{A}$ denote the set of all state-action pairs. The dynamics in the MDP are given by the *known* transition function P, where $P(s'|s,a)$ specifies the probability of moving to state $s' \in \mathcal{S} \cup \{g\}$ after taking action a in state s.

Each episode begins in state s_0 and proceeds with the learner selecting actions until the goal state g is reached. When the goal state is reached, the current episode ends and a new one begins. At the start of each episode k, the adversary selects a cost function $c_k : \Gamma \to [0,1]$, assigning a cost to each state-action pair. We denote the sparsity level as $M = \max_k \sum_{(s,a)\in\Gamma} \mathbb{I}\{c_k(s,a) > 0\}$ the

maximum number of non-zero costs in an episode. We work in the full-information setting where the entire function c_k is revealed to the learner at the end of the episode. The objective is to minimize the total cost over all episodes, which requires a balance of minimizing the accumulated costs while ensuring the goal state is reached efficiently.

We use super-scripts to denote the time-step within an episode and sub-scripts to denote the episode: e.g. (s_k^t, a_k^t) refers to the state-action pair at the t-th time-step of the k-th episode. We sometimes omit the sub-script when referring to an arbitrary episode. We now define some important concepts:

- A **stationary policy** π is a mapping such that $\pi(\cdot|s)$ is a probability distribution over the choice of action $a \sim \pi(\cdot|s)$ in state s. A policy is called **proper** if it reaches the goal g in finite time from any initial state in $\mathcal{S}$ with probability one, and improper if not. Let Π_p be the set of all stationary proper policies. We assume the existence of at least one proper policy.

- The **expected hitting time** $T^\pi(s)$ is the expected number of steps required to reach g from state s under π. Letting $I_\pi(s)$ be the random number of time-steps used to reach the goal state when executing a policy π in an episode starting from state s, then $T^\pi(s) = \mathbb{E}[I_\pi(s)]$. For any proper policy π, $I_\pi(s)$ and $T^\pi(s)$ are finite for all $s \in \mathcal{S}$.

- The **fast policy** π_f is the deterministic policy that minimizes the worst-state expected hitting time, and the **diameter** D of the MDP is the corresponding expected hitting time:

$$\pi_f = \arg\min_{\pi \in \Pi_p} \max_{s \in \mathcal{S}} T^\pi(s), \qquad D = \max_{s \in \mathcal{S}} T^{\pi_f}(s) = \max_{s \in \mathcal{S}} \min_{\pi \in \Pi_p} T^\pi(s).$$

  Since the transition function P is known, both the fast policy π_f and the diameter D can be computed offline prior to the learning process. We assume $D \geqslant 1$.

- The **cost-to-go** function $J_c^\pi : \mathcal{S} \to [0, \infty)$ is the expected cost suffered during an episode executing policy π and starting from state s, given a cost function c and a proper policy π. It is defined as

$$J_c^\pi(s) = \mathbb{E}\Big[\sum_{t=1}^{I_\pi(s)} c(s^t, a^t) \Big| P, \pi, s^1 = s \Big],$$

  where the expectation is with respect to the randomness in the action sampling and state

transitions. We use J_k^π to denote the cost-to-go from the initial state s_0 using the cost function c_k in episode k. Note that s_0 denotes the fixed initial state of each episode, while s^1 denotes the state at the first time-step within an episode

- The **regret** R_K is the primary measure of performance by which the learner is evaluated. It is the difference between the total cost over all episodes of the policies $\pi_1, \dots, \pi_K$ chosen by the learner, and the total cost of the best proper deterministic policy in hindsight, $\pi^\star \in \arg\min_{\pi \in \Pi_p} \sum_{k=1}^K J_k^\pi$:

$$R_K = \sum_{k=1}^{K} \sum_{t=1}^{I_{\pi_k}(s_0)} c_k(s_k^t, a_k^t) - \sum_{k=1}^{K} J_k^{\pi^\star}.$$

- The **occupancy measure** $q_\pi \in \mathbb{R}^\Gamma_{\geqslant 0}$ of a proper policy π is the expected number of visits to state-action pairs in an episode executing policy π starting from s_0:

$$q_\pi(s,a) = \mathbb{E}\Big[\sum_{i=1}^{I_\pi(s_0)} \mathbb{I}\{s^i = s, a^i = a\} \Big| P, \pi, s^1 = s_0 \Big].$$

The marginal $q_\pi(s) = \sum_{a \in \mathcal{A}} q_\pi(s,a)$ gives the expected number of visits to state s. Given a vector $q \in \mathbb{R}^\Gamma_{\geqslant 0}$, if it corresponds to a valid occupancy measure, the corresponding policy π_q can be recovered via normalization as $\pi_q(a|s) = q(s,a) / \sum_{a'} q(s,a')$ (Zimin and Neu, 2013; Rosenberg and Mansour, 2021).

6.2 SSP as Online Linear Optimisation and Online Mirror Descent

Using occupancy measures, the cost-to-go to can be expressed in a linear form:

$$J_k^\pi = \sum_{(s,a) \in \Gamma} q_\pi(s,a) c_k(s,a) = \langle q_\pi, c_k \rangle.$$

If the learner executes a stationary proper policy π_k in episode k, the expected regret can thus be reformulated as an online linear optimisation problem on the space of occupancy measures:

$$\mathbb{E}\big[R_K\big] = \sum_{k=1}^{K} \Big\{ J_k^{\pi_k} - J_k^{\pi^\star} \Big\} = \sum_{k=1}^{K} \langle q_{\pi_k} - q_{\pi^\star}, c_k \rangle.$$

Online linear optimisation is a well studied problem and falls within the class of sequential decision-making problems considered in Chapter 2, which can be solved using Online Mirror Descent (OMD). In the SSP framework, OMD is applied on the space of occupancy measures corresponding to proper policies with expected hitting time bounded by some $T > 0$ defined as:

$$\Delta(T) = \left\{ q \in \mathbb{R}^{\Gamma} \,:\, \sum_{(s,a)\in\Gamma} q(s,a) \leqslant T, \quad \forall s \in \mathcal{S} : \sum_{a\in\mathcal{A}} q(s,a) - \sum_{(s',a')\in\Gamma} P(s \,|\, s',a') q(s',a') = \mathbb{I}\{s = s_0\} \right\}.$$

The first constraint ensures the expected hitting time is bounded by T, while the second is a flow constraint ensuring the vector corresponds to the occupancy measure of a policy. The regret bounds of OMD will hold against any fixed comparator policy as long as T is large enough such that $\Delta(T)$ contains the occupancy measure of the optimal policy, i.e. $q_{\pi^\star} \in \Delta(T)$ or $T \geqslant T_\star$ where we denote by $T_\star = T^{\pi^\star}(s_0)$ the expected hitting time of $\pi^\star$. OMD with a strictly convex differentiable regulariser ψ and step-size η selects occupancy measures computed through the update

$$q_1 = \operatorname*{arg\,min}_{q\in\Delta(T)} \psi(q), \qquad q_{k+1} = \operatorname*{arg\,min}_{q\in\Delta(T)} \Big\{ \eta \cdot \langle q, c_k \rangle + D_\psi(q, q_k) \Big\}, \tag{6.1}$$

where $D_\psi(x,y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x - y\rangle$ is the Bregman divergence with respect to ψ. This update can be computed efficiently for all the regularisers we will discuss (see Section 6.7.1). As discussed in the previous section, we can easily recover via normalization the corresponding policy π_{q_k} that will be executed by the learner.

If the regulariser satisfies for some $\alpha > 0$, any $q \in \mathbb{R}^{\Gamma}$ and all $k \geqslant 1$:

$$\nabla\psi(q) \in [\nabla\psi(q_k), \nabla\psi(q_k) - \eta c_k] \implies \nabla^2\psi(q) \succeq \alpha \nabla^2\psi(q_k), \tag{6.2}$$

(this is satisfied by many common regularisers), then a standard result (see e.g. Theorem 6 in Bubeck et al. (2018), Theorem 5.5 in Bubeck and Cesa-Bianchi (2012)) gives the following general regret bound for OMD:

$$\sum_{k=1}^{K} \langle q_k - q_{\pi^\star}, c_k \rangle \leqslant \underbrace{\frac{\psi(q_{\pi^\star}) - \psi(q_1)}{\eta}}_{\text{Penalty}} + \underbrace{\frac{\eta}{2\alpha} \sum_{k=1}^{K} \|c_k\|^2_{\nabla^2\psi(q_k)^{-1}}}_{\text{Stability}} \tag{6.3}$$

where $\|q\|_A^2 = \sum_{s,a,s',a'} q(s,a) A((s,a),(s',a')) q(s',a')$ for a matrix $A \in \mathbb{R}^{\Gamma\times\Gamma}$.

Various regret bounds can be obtained by instantiating the above with different regularisers. In particular, Chen et al. (2021c) use the negative-entropy to obtain a $\mathcal{O}\left(\sqrt{DKT_\star \log SAT_\star}\right)$ regret bound, in which the dimension of the problem given by the size of the state-action space SA appears as $\sqrt{\log SA}$. While in the worst-case this is unimprovable (we show this in Theorem 6.3.1), we consider *sparse* SSP problems where M, the maximum number of state-action pairs with non-zero cost in an episode, can be much smaller than the dimension SA. *The focus of this chapter is to understand the behaviour of the minimax regret for general sparse problems beyond the worst-case where* $M = SA$. In the next section, we begin by showing that OMD with negative entropy regularisation used by Chen et al. (2021c) is not well suited to this problem.

6.3 Failure of Negative Entropy Regularization

In the general non-sparse setting, Chen et al. (2021c) use the negative entropy to achieve a regret of $\mathcal{O}\big(\sqrt{DKT_\star \log SAT_\star}\big)$, which in the non-sparse setting has optimal dependence on SA (as we show later in Theorem 6.4.2). Despite this success, the negative entropy fails to benefit from sparsity in its dependence on S, as shown by the result below. As we will see in Section 6.4, this establishes the negative entropy as a sub-optimal choice for sparse SSP problems.

Theorem 6.3.1. *For any $S \geqslant 6$, there exists an SSP instance with a fixed horizon of 3, sparsity level $M = 3$, an action space of size $A = 2$ and state space of size S such that the regret of OMD (6.1) with negative-entropy regularization and any step-size $\eta > 0$ after K episodes is* $\mathbb{E}\left[R_K\right] = \Omega\big(\min\big\{\sqrt{K \log S}, K\big\}\big)$.

This result shows that despite the SSP instance being sparse ($M = 3$), the regret of OMD with negative entropy regularization nevertheless scales as $\sqrt{\log S}$, which is the same dependence on S as in the non-sparse setting. For sparse problems, the negative entropy provides no improvement on the regret with respect to S. This highlights that existing approaches and regularisers are inadequate to appropriately exploit sparse problems and motivates considering alternate regularisers specifically designed for the geometry of sparse problems, as we do in the next section.

Proof. We begin by describing the SSP instance, then give some intuition and finally provide the detailed computations of the regret.

SSP Instance: Fix K even, $S \geqslant 6$, $A = 2$ and $N = S - 5$. Consider the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, s_0, g)$ illustrated in Figure 6.1 where where $\mathcal{S} = \{s_0, s_0^L, s_g^L, s_0^R, s_g^R, s_1^R, ..., s_N^R\}$ and $\mathcal{A} = \{a_1, a_2\}$.

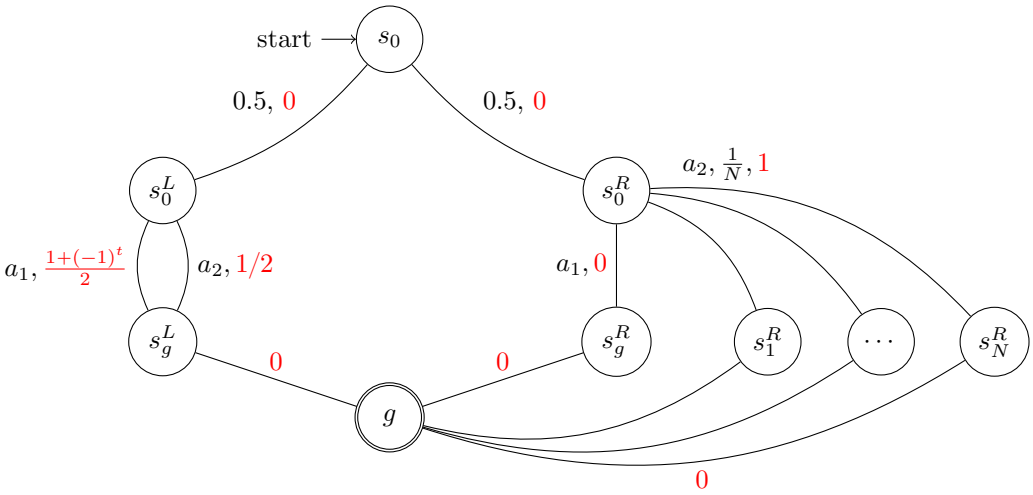

Figure 6.1: Diagram illustrating MDP construction for the proof of Theorem 6.3.1. When an action is not specified for an edge, then both actions give the same transition and cost. If an edge has a number in black, it is a transition probability; if it does not then the transition is deterministic. The costs are given in red. The formal description of the MDP is given below.

The transitions and costs (in each episode k) are defined as:

- s_0: $p(s_0^L|s_0, a) = p(s_0^R|s_0, a) = 1/2$ and $c_k(s_0, a) = 0$ for all $a \in \mathcal{A}$.
- s_0^L: $p(s_g^L|s_0^L, a) = 1$ for all $a \in \mathcal{A}$ and $c_k(s_0^L, a_1) = \frac{1+(-1)^k}{2}$, $c_k(s_0^L, a_2) = 1/2$.
- s_0^R: $p(s_g^R|s_0^R, a_1) = 1$, $p(s_i^R|s_0^R, a_2) = 1/N$ and $c_k(s_0^R, a_1) = 0$, $c_k(s_0^R, a_2) = 1$.
- s_i^R: $p(g|s_i^R, a) = 1$ and $c_k(s_i^R, a) = 0$ for all $a \in \mathcal{A}$.
- s_g^L: $p(g|s_g^L, a) = 1$ and $c_k(s_g^L, a) = 0$ for all $a \in \mathcal{A}$.
- s_g^R: $p(g|s_g^R, a) = 1$ and $c_k(s_g^R, a) = 0$ for all $a \in \mathcal{A}$.

This SSP instance has a fixed horizon of 3: all policies have a hitting time of exactly 3 (the states s_g^L and s_g^R are added to guarantee this). As a result we have that $T_\star = D = 3$. Also note that there are at most 3 state-action pairs that have non-zero cost, therefore the sparsity level $M = 3$.

Intuition: The key idea is to reduce the right part of the SSP instance to an experts problem with 2 actions and a skewed initial distribution over the actions. The initial occupancy measure played by OMD in (6.1) is $q_1 = \arg\min_{q \in \Delta(T)} \psi(q)$. For most regularisers, including the negative entropy, this encourages q_1 to be uniform across the state-action space while maintaining the constraints on the flow and expected hitting time. Since we consider a fixed-horizon MDP, only the flow constraint is relevant.

Consider the right part of the SSP instance described above. Since $s_1^R, ..., s_N^R$ constitute a large majority of the states (especially for large N), $\psi(q)$ is mainly affected by the values of q in these states. In order to minimize $\psi(q)$, q_1 needs to ensure the expected number of visits to these N states is sufficiently high. However since for any q and any $i \geqslant 1$, $q(s_i^R) = \frac{1}{N} q(s_0^R, a_2)$, for $q_1(s_i^R)$ to be sufficiently large then $q_1(s_0, a_2)$ needs to be much larger (by a factor of N). This results in q_1 being heavily skewed towards the sub-optimal a_2 in s_0^R. For the negative entropy, this gives specifically $q_1(s_0^R, a_1) \approx \frac{1}{\sqrt{N}}$. A larger step-size η could be used to compensate for this skewed initial occupancy measure and achieve low regret but this would then lead to large regret on the left part of the MDP. The trade-off between the two parts give the desired $\sqrt{\log N}$ dependence.

The necessity for a large step-size to compensate the skewed occupancy measure comes from the negative entropy stretching euclidean distance near the boundary of the space in such a way that two nearby points in terms of euclidean distance can be arbitrarily far in terms of negative entropy. This property does not generalize to all regularisers and in fact provides insights for designing a regulariser to appropriately handle sparsity. In particular, the regulariser we consider in the next section does not suffer from the same issue because the stretching of euclidean distance is finite since its gradient does not diverge at the boundary (i.e. as $q(s, a) \to 0$) unlike the negative entropy.

Detailed Computations: From Appendix B.1 in Rosenberg and Mansour (2021) (we can ignore the optimization over λ because we are in a fixed horizon setting), the update of OMD with negative entropy for any $k \geqslant 0$ can be computed by solving a convex optimization problem:

$$q_{k+1}(s,a) = q_k(s,a)e^{B_k^{v_{k+1}}(s,a)}, \quad \text{where} \quad B_k^v(s,a) = v(s) - \eta c_k(s,a) - \sum_{s' \in \mathcal{S}} p(s'|s,a)v(s'),$$
$$v_{k+1} = \arg\min_v \mathcal{D}_k(v),$$
$$\mathcal{D}_k(v) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q_k(s,a)e^{B_k^v(s,a)} - v(s_0),$$

with $q_0(s,a) = 1$ and $c_0(s,a) = 0$. This allows us to compute exactly the points played by the algorithm on the SSP instance described above, and in turn compute the regret, from which the result will follow. We leave the tedious computations to Section 6.7.2 and give the answers below for the two parts of the MDP separately:

- Left part of the MDP:

$$\begin{aligned} k \text{ even} \implies & \quad q_k(s_0^L, a_1) = \frac{0.5}{1 + e^{-0.5\eta}}, & q_k(s_0^L, a_2) = \frac{0.5}{1 + e^{0.5\eta}} \\ k \text{ odd} \implies & \quad q_k(s_0^L, a_1) = 0.25, & q_k(s_0^L, a_2) = 0.25 \end{aligned}$$

  Hence the expected costs suffered on the left part of the MDP are:

$$\begin{aligned} \sum_{k=1}^{K} \Big\{ q_k(s_0^L, a_1)c_k(s_0^L, a_1) + q_k(s_0^L, a_2)c_k(s_0^L, a_2) \Big\} &= 0.25K + 0.5 \sum_{t=1}^{K/2} \Big\{ \frac{0.5}{1 + e^{-0.5\eta}} - 0.25 \Big\} \\ &= 0.25K + 0.5 \frac{K}{2} \frac{0.5 - 0.25 - 0.25e^{-0.5\eta}}{1 + e^{-0.5\eta}} \\ &= 0.25K + \frac{K}{16} \cdot \frac{1 - e^{-0.5\eta}}{1 + e^{-0.5\eta}} \\ &= 0.25K + \frac{K}{16} \cdot \min\Big\{ \frac{\eta}{5}, \frac{1}{2} \Big\}. \end{aligned} \tag{6.4}$$

- Right part of the MDP:

$$q_k(s_0^R, a_2) = \frac{0.5\sqrt{N}}{\sqrt{N} + e^{0.5\eta(k-1)}}.$$

  Hence the expected costs suffered on the right part of the MDP are:

$$\sum_{k=1}^{K} q_k(s_0^R, a_2)c_k(s_0^R, a_2) = \sum_{k=1}^{K} \frac{0.5\sqrt{N}}{\sqrt{N} + e^{0.5k\eta}}$$

$$
\begin{aligned}
&\geqslant \int_1^{K+1} \frac{0.5\sqrt{N}}{\sqrt{N}+e^{0.5\eta x}}dx = 0.5K - \int_1^{K+1} \frac{0.5e^{0.5\eta x}}{\sqrt{N}+e^{0.5\eta x}}dx \\
&= 0.5K - \Big[\frac{1}{\eta}\log(\sqrt{N}+e^{0.5\eta x})\Big]_1^{K+1} \\
&= 0.5K - \frac{1}{\eta}\log(\sqrt{N}+e^{0.5\eta(K+1)}) + \frac{1}{\eta}\log(\sqrt{N}+e^{0.5\eta}) \\
&\geqslant 0.5K - \frac{1}{\eta}\log(2e^{0.5\eta(K+1)}) + \frac{1}{\eta}\log\sqrt{N} \qquad \text{assuming } \sqrt{N} \leqslant e^{0.5\eta(K+1)} \\
&= 0.5K - 0.5(K+1) - \frac{1}{\eta}\log 2 + \frac{1}{2\eta}\log N \\
&= -0.5 + \frac{1}{2\eta}\log\frac{N}{4}.
\end{aligned}
$$

If $\sqrt{N} > e^{0.5\eta(K+1)}$, then we have

$$
\begin{aligned}
0.5K - \frac{1}{\eta}\log(\sqrt{N}+e^{0.5\eta(K+1)}) + \frac{1}{\eta}\log(\sqrt{N}+e^{0.5\eta}) &= 0.5K + \frac{1}{\eta}\log\Big(\frac{\sqrt{N}+e^{0.5\eta}}{\sqrt{N}+e^{0.5\eta(K+1)}}\Big) \\
&\geqslant 0.5K + \frac{1}{\eta}\log\Big(\frac{e^{0.5\eta(K+1)}+e^{0.5\eta}}{2e^{0.5\eta(K+1)}}\Big) \\
&\geqslant 0.5K + \frac{1}{\eta}\log\Big(\frac{1+e^{-0.5\eta K}}{2}\Big) \\
&\geqslant 0.25K,
\end{aligned}
$$

using that $\frac{1+e^{-0.5Kx}}{2} \geqslant e^{-0.25Kx}$ since $cosh(x) \geqslant 1$. So we have

$$
\sum_{k=1}^{K} q_k(s_0^R, a_2)c_k(s_0^R, a_2) \geqslant \min\Big\{-0.5 + \frac{1}{2\eta}\log\frac{N}{4}, 0.25K\Big\}. \tag{6.5}
$$

Combining the costs from the left part in (6.4) and from the right part in (6.5), we have:

$$
\begin{aligned}
\sum_{k=1}^{K}\langle q_k, c_k\rangle &= \sum_{k=1}^{K}\Big\{q_k(s_0^L, a_1)c_k(s_0^L, a_1) + q_k(s_0^L, a_2)c_k(s_0^L, a_2)\Big\} + \sum_{k=1}^{K}\Big\{q_k(s_0^R, a_2)c_k(s_0^R, a_2)\Big\} \\
&\geqslant 0.25K + \frac{K}{16}\cdot\min\Big\{\frac{\eta}{5}, \frac{1}{2}\Big\} + \min\Big\{-0.5 + \frac{1}{2\eta}\log\frac{N}{4}, 0.25K\Big\}.
\end{aligned}
$$

Regret lower-bound: consider $q_\star$ defined as follows:

- $q_\star(s_0, a) = 1/2$ for all $a \in \mathcal{A}$
- $q_\star(s_0^L, a_2) = 1/2$, $q_\star(s_0^L, a_1) = 0$
- $q_\star(s_g^L, a) = 1/4$ for all $a \in \mathcal{A}$
- $q_\star(s_0^R, a_1) = 1/2$, $q_\star(s_0^R, a_2) = 0$, $q_\star(s_i^R, a) = 0$
- $q_\star(s_g^R, a) = 1/4$ for all $a \in \mathcal{A}$

It is straightforward to check that $q_\star$ satisfies the flow constraints and is an occupancy measure. We obtain

$$\begin{aligned}
&\sum_{k=1}^{K}\langle q_\star, c_k\rangle = \sum_{k=1}^{K}\Big\{q_\star(s_0^L, a_2)\cdot 0.5\Big\} = 0.25K \\
\implies R_K &\geqslant \sum_{k=1}^{K}\langle q_k - q_\star, c_k\rangle \geqslant \frac{K}{16}\cdot\min\Big\{\frac{\eta}{5}, \frac{1}{2}\Big\} + \min\Big\{-0.5 + \frac{1}{2\eta}\log\frac{N}{4}, 0.25K\Big\} \\
&\geqslant \min\Big\{\frac{1}{2}\sqrt{\frac{1}{10}K\log\frac{N}{4}}, \frac{K}{32}\Big\} - 0.5.
\end{aligned}$$

Recalling that $N = S - 5$, we have $R_K = \Omega\big(\min\big\{\sqrt{K\log S}, K\big\}\big)$ for an MDP where the sparsity level is $M = 3$, concluding the proof. □

6.4 The Benefits of Sparsity

In this section, we show that it is possible to achieve a regret bound of order $\mathcal{O}\big(\sqrt{DKT_\star\log(MT_\star)}\big)$, where M is the maximum number of non-zero entries in the cost. This is our main result and together with the lower bound in Theorem 6.4.2 establishes that the sparsity level M acts as a measure of effective dimension instead of the state-action space size SA for SSP with full-information feedback.

In the previous section, we showed and discussed that the negative entropy, the regulariser used in OMD by existing methods, is inadequate to handle sparse SSP problems. Motivated by this failure, we consider alternate regularisers. However, identifying a suitable regulariser poses two key challenges. Firstly, it must work for SSP and the associated technical complexities compared to other simpler online learning problems. In particular, it needs to match the dependence in terms of the other non-sparsity-related quantities appearing in the regret of the negative entropy (i.e. $D, T_\star, K$). Second, it must explicitly leverage sparsity to improve performance. We propose the following family of regularisers parameterised by $p > 1$:

$$\psi_p(q) = p\cdot\left(-1 + \|q\|_{1+1/p}^{1+1/p}\right) = p\cdot\left(-1 + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}|q(s,a)|^{1+1/p}\right). \tag{6.6}$$

As $p \to \infty$, the regulariser in (6.6) converges to the negative entropy. On the other hand, as $p \to 1$, the regulariser converges to the squared Euclidean norm that enforces much weaker regularization on sparse points. Therefore, ψ_p allows smooth interpolation between dense and

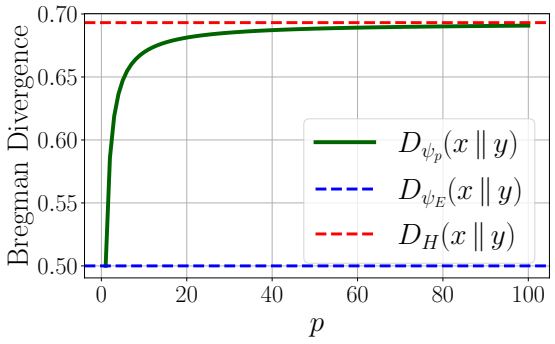

Figure 6.2: Bregman Divergence between a deterministic distribution $x = [0, 1]$ and the uniform distribution $y = [1/2, 1/2]$ for our regulariser ψ_p, squared Euclidean norm ψ_E and negative entropy H for increasing values of p.

sparse regimes via the tunable parameter p (see Figure 6.2 for a comparison). In particular, ψ_p for small p induces an OMD update that is able to easily move to and away from the boundary of the space, making it robust to the skewed initial occupancy measure on the SSP instance which caused the failure of the negative entropy in Section 6.3. The parameter p also controls a trade-off between the stability and penalty term in (6.3), which ultimately will enable the removal of the dependence on SA.

Versions of this family of regularisers can be found in the convex optimization literature (Juditsky and Nemirovski, 2010; Nesterov and Nemirovski, 2013). As far as we are aware, its use with OMD is novel. A regulariser involving an r-norm with $r \in (1, 2]$ has been used but the norm is squared rather than to the power of r (see e.g. Section 6.7 in Orabona (2019)). Our regulariser is also similar in flavor to the Tsallis-entropy in the sense that it converges to the negative entropy in the limit of its parameter.

We note that OMD with the above regulariser can be implemented efficiently for any p: the projection step of OMD over $\Delta(T)$ can be written as a convex optimization problem as in Rosenberg and Mansour (2021), which can be solved efficiently.

We can now turn to our main result, which establishes regret bounds that scale with the sparsity level M for OMD with the regulariser in (6.6) when M is known.

Theorem 6.4.1. *Consider OMD with ψ_p as regulariser. If $T > e$ is such that $q_{\pi^\star} \in \Delta(T)$, $\eta = \sqrt{\frac{pT^{1+1/p}}{KDM^{1/p}}}$, $p = \log(TM)$, then $\mathbb{E}[R_K] \leqslant \mathcal{O}\left(\sqrt{DKT\log(MT)}\right)$ for $K > \frac{T\log TM}{4D}$.*

Proof. We first show that ψ_p satisfies the condition (6.2) with $\alpha = 1$, allowing us to use the regret bound in (6.3). First, implicitly assuming that ψ_p is defined on $\mathbb{R}^{\Gamma}_{\geqslant 0}$ we have

$$\begin{aligned}
&\psi_p(q) = p \cdot \Big(-1 + \|q\|_{1+1/p}^{1+1/p}\Big) = p \cdot \Big(-1 + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} |q(s,a)|^{1+1/p}\Big) \\
\implies &\frac{\partial \psi_p(q)}{\partial q(s,a)} = (p+1)q(s,a)^{1/p} \\
\implies &\frac{\partial^2 \psi_p(q)}{\partial q(s,a)^2} = \Big(1 + \frac{1}{p}\Big) q(s,a)^{1/p-1} \\
\implies &\nabla \psi_p(q) = (p+1)q^{1/p}, \qquad \nabla^2 \psi_p(q) = \operatorname{diag}\Big(\frac{p+1}{p} q^{1/p-1}\Big)
\end{aligned}$$

From this, we can show ψ_p satisfies the condition (6.2) with $\alpha = 1$:

$$\begin{aligned}
\nabla\psi_p(q) \in \Big[\nabla\psi_p(q_k), \nabla\psi_p(q_k) - \eta c_k\Big] &\implies q^{1/p}(s,a) \leqslant q_k^{1/p}(s,a) \\
&\implies \frac{1}{q(s,a)} \geqslant \frac{1}{q_k(s,a)} \\
&\implies \frac{1}{q^{1-1/p}(s,a)} \geqslant \frac{1}{q_k^{1-1/p}(s,a)} \\
&\implies \nabla^2\psi_p(q) \succeq \nabla^2\psi_p(q_k).
\end{aligned}$$

$$\text{Eq (6.3)} \implies \sum_{k=1}^{K} \langle q_k - q_{\pi^\star}, c_k\rangle \leqslant \underbrace{\frac{\psi_p(q_{\pi^\star}) - \psi_p(q_1)}{\eta}}_{\text{Penalty}} + \underbrace{\frac{\eta}{2\alpha}\sum_{k=1}^{K} \|c_k\|^2_{\nabla^2\psi_p(q_k)^{-1}}}_{\text{Stability}}$$

Using that $q_{\pi_\star} \in \Delta(T)$, we can bound the Penalty term:

$$\psi_p(q_{\pi_\star}) - \psi_p(q_1) = p\left(\|q_{\pi_\star}\|_{1+1/p}^{1+1/p} - \|q_1\|_{1+1/p}^{1+1/p}\right) \leqslant p \cdot \|q_{\pi_\star}\|_1^{1+1/p} \leqslant p \cdot T^{1+1/p} \tag{6.7}$$

Using that $\nabla^2\psi_p(q)^{-1} = \operatorname{diag}\Big(\frac{p}{p+1} q^{1-1/p}\Big)$ and $c_k(s,a)^2 \leqslant c_k(s,a)$, we get

$$\begin{aligned}
\|c_k\|^2_{\nabla^2\psi_p(q_k)^{-1}} &\leqslant \frac{p}{p+1}\sum_{s,a} c_k(s,a) q_k(s,a)^{1-1/p} \leqslant \|c_k\|_1 \sum_{s,a} \frac{c_k(s,a)}{\|c_k\|_1} q_k(s,a)^{1-1/p} \\
&\leqslant \|c_k\|_1 \Big(\sum_{s,a} \frac{c_k(s,a)}{\|c_k\|_1} q_k(s,a)\Big)^{1-1/p} \qquad (6.8) \\
&= \|c_k\|_1^{1/p} \langle c_k, q_k\rangle^{1-1/p} \leqslant M^{1/p} \max\big\{1, \langle c_k, q_k\rangle\big\} \leqslant M^{1/p}\big(1 + \langle c_k, q_k\rangle\big),
\end{aligned}$$

where the key step (6.8) uses Jensen's inequality on the concave function $x^{1-1/p}$ ($p > 1$) and

probability distribution $c_k/\|c_k\|_1$. Plugging this into the Stability term and combining with (6.7):

$$\begin{aligned}
&\sum_{k=1}^{K}\langle q_k - q_{\pi^\star}, c_k\rangle \leqslant \frac{pT^{1+1/p}}{\eta} + \frac{\eta M^{1/p}K}{2} + \frac{\eta M^{1/p}}{2}\sum_{k=1}^{K}\langle q_k, c_k\rangle \\
\implies &\sum_{k=1}^{K}\langle q_k - q_{\pi^\star}, c_k\rangle \leqslant \frac{1}{1-\frac{\eta M^{1/p}}{2}}\left(\frac{pT^{1+1/p}}{\eta} + \frac{\eta M^{1/p}K}{2} + \frac{\eta M^{1/p}}{2}\sum_{k=1}^{K}\langle q_{\pi_\star}, c_k\rangle\right) \\
\implies &\sum_{k=1}^{K}\langle q_k - q_{\pi^\star}, c_k\rangle \leqslant \frac{2pT^{1+1/p}}{\eta} + 2\eta M^{1/p}DK,
\end{aligned}$$

where the last step uses that $\sum_{k=1}^{K}\langle q_{\pi_\star}, c_k\rangle \leqslant \sum_{k=1}^{K}\langle q_{\pi_f}, c_k\rangle \leqslant K\|q_{\pi_f}\|_1\|c_k\|_\infty \leqslant DK$, $D \geqslant 1$ and $\eta \leqslant 4M^{-1/p} \iff 1-\eta M^{1/p}/2 \geqslant 1/2$. Tuning $\eta = \sqrt{\frac{pT^{1+1/p}}{KDM^{1/p}}}$ and $p = \log(TM)$ gives the result. Note $\eta \leqslant 4M^{-1/p}$ for $K > \frac{pT^{1+1/p}M^{1/p}}{16D} = \frac{eT\log(TM)}{16D}$. □

Provided we can suitably select $T \approx T_\star$ (see remark below), this result establishes that sparsity does lead to an improvement in the minimax regret. In Section 6.4.1, we show that the dependence on M and SA is optimal, ruling out polynomial improvements from sparsity such as in the experts setting Kwon and Perchet (2016). This highlights that M acts as the effective dimension of the problem instead of SA. In particular, if the sparsity level M is constant, then we obtain a dimension-independent regret of $\mathcal{O}(\sqrt{DKT\log T})$.

Parameter-Free Algorithm. The above result assumes knowledge of the sparsity level M and the expected hitting time of the optimal policy $T_\star$. By adapting techniques from Chen et al. (2021c) for the hitting time (run parallel OMD over grid of guesses for $T_\star$ and a meta-learning algorithm on top) and from Kwon and Perchet (2016) for the sparsity level (run OMD with a current guess for the sparsity level, when/if it is violated, restart with a new guess), it is possible to obtain a parameter free algorithm that achieves the same bound as Theorem 6.4.1 (up to double-logarithmic factors) without knowledge of M and $T_\star$ (see Theorem 4.4 and Appendix C.2 in Johnson et al. (2025b) for how the algorithm can be extended to achieve this).

Soft-Sparsity Bound. Although we express the bound in terms of the sparsity level M, it can be seen that the analysis above holds more generally if M is instead an upper bound on the ℓ_1 norm of the costs: $M = \max_k\|c_k\|_1$. This relaxation allows our result to cover "softly sparse" cost structures and aligns with the notion of first-order bounds commonly studied in the online learning literature (Neu, 2015; Wei and Luo, 2018; Wagenmaker et al., 2022).

Follow-The-Regularised-Leader. In Chapter 4, we argued that FTRL is usually preferred over OMD due to some theoretical advantages. In this SSP setting however, the OMD analysis is non-standard and requires modifications compared to standard OLO/OCO. It is unclear how to adapt the FTRL analysis in a similar way, and as such the success of OMD for SSP could be used as an argument in favour of OMD over FTRL (though of course there is no evidence that FTRL does not work).

Comparison to Prior Work We now compare our results directly to prior work:

- Rosenberg and Mansour (2021) analyse OMD with negative entropy and obtain

$$\mathbb{E}\left[R_K\right] \leqslant \mathcal{O}\left(\frac{D}{c_{\min}}\sqrt{K \log \frac{DSA}{c_{\min}}}\right), \quad \eta = \sqrt{\frac{3\log(DSA/c_{\min})}{K}},$$

  where $c_{\min} = \min_k \min_{(s,a)\in\mathcal{S}\times\mathcal{A}} c_k(s,a) > 0$. This requires strictly positive costs, meaning their method cannot exploit sparsity ($c_{\min} = 0$ would break the bound). For episodes with zero costs, they propose running OMD on artificially perturbed costs and achieve $\mathbb{E}\left[R_K\right] \leqslant \tilde{\mathcal{O}}(K^{3/4}\sqrt{DT_\star})$.

  They circumvent the need to know $T_\star$ by upper-bounding it with $D/c_{\min}$, which explains the appearance of $c_{\min}$ in their bounds. More generally, if $T^\star$ was known then for $T > T_\star$, their guarantee becomes

$$\mathbb{E}\left[R_K\right] \leqslant \mathcal{O}\Big(T\sqrt{K\log(SAT)}\Big), \quad \eta = \sqrt{\frac{3\log(SAT)}{K}}.$$

  As discussed below, Chen et al. (2021c) provide a meta-learning procedure that achieves tighter bounds without knowledge of $T_\star$.

- Chen et al. (2021c) also consider OMD with negative entropy and for known $T^\star$ obtain a tighter bound for $T > T_\star$:

$$\mathbb{E}\left[R_K\right] \leqslant \mathcal{O}\Big(\sqrt{DTK\log(SAT)}\Big), \quad \eta = \sqrt{\frac{T\log(SAT)}{DK}}.$$

  Their step-size and bound are essentially the same as ours, with the main difference being that SA appears in place of the sparsity level M. Consequently, our approach matches their guarantees in the worst-case ($M = SA$) while providing improvements when $M \ll SA$,

thereby effectively exploiting sparsity in high-dimensional SSP problems.

To handle unknown $T_\star$, they introduce a meta-learning algorithm that runs multiple OMD instances with different guesses for $T_\star$ and competes against the best, recovering an overall regret bound of the same order as when $T_\star$ is known. As discussed in the remark on parameter-free algorithms, the same technique works with our regulariser.

Both works additionally provide high-probability guarantees, though via more complex procedures. The analysis of Chen et al. (2021c) can again be adapted to our setting to yield analogous high-probability bounds that scale with M.

6.4.1 Lower bound

In this section, we provide a general lower bound for sparse SSP problems.

Theorem 6.4.2. *For any $D, T_\star, K, S, A$ with $T^\star \geqslant D \geqslant 3\log S$, $S(A-1) \geqslant 400$, $K \geqslant \frac{800T^\star}{D}\log M$ and $M \geqslant 101$, there exists an SSP instance with stochastic M-sparse costs, S states and A actions such that its diameter is D, the expected hitting time of the optimal policy is $T^\star$, and the expected regret with respect to the randomness of the losses for any learner after K episodes is $\mathbb{E}[R_K] \geqslant \Omega\big(\sqrt{KT^\star D\log M}\big)$.*

For general M (> 100), the lower bound matches the upper bound established in Theorem 6.4.1 in its dependence on M, characterizing the minimax regret for general sparse problems (up to a $\log T_\star$ term). For $M = SA$, our result gives a $\Omega\big(\sqrt{KT^\star D\log SA}\big)$ lower bound improving on the $\Omega\big(\sqrt{KT^\star D}\big)$ lower bound of Chen et al. (2021c). In particular, this establishes the optimal dependence on the size of the state-action space SA in the minimax regret for the general non-sparse SSP problem.

Proof. The proof is inspired by an SSP instance from Chen et al. (2021c) combined with a probabilistic costs construction, which then requires some non-trivial arguments to extend to the sparse SSP problem.

SSP Instance: We describe the SSP instance with stochastic costs. Fix the following

- $B = \lceil \frac{\log S/2}{\log 2} \rceil - 2$
- $N = 2^{B+1} \geqslant 2^{\frac{\log S/2}{\log 2} - 1} = \frac{S}{4}$
- $L = \min\Big\{M - 1, N \cdot (A-1)\Big\} \geqslant \frac{M}{8}$
- $D' = D - B - 2$
- $T' = T_\star - B - 1$

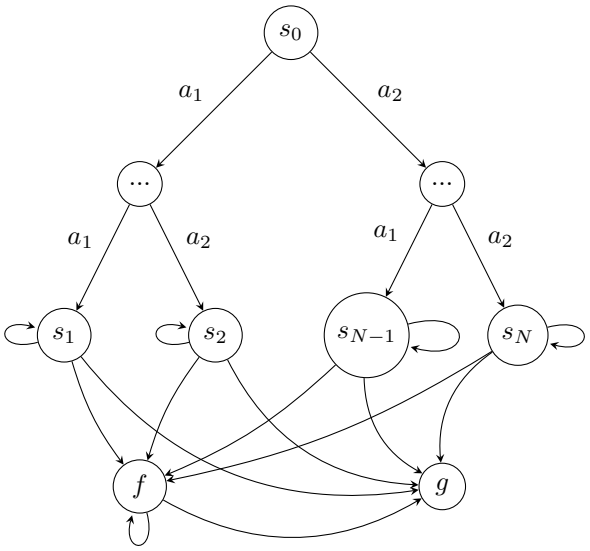

Figure 6.3: Diagram illustrating MDP construction for the proof of Theorem 6.4.2.

Consider the MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, s_0, g)$ illustrated in Figure 6.3. The first part of the states are represented by a binary tree of depth $B+2$ and allow us to formerly consider the N states at the bottom of the tree that matter, while avoiding an assumption on the existence of a state with $A \approx S$ actions as was done in prior work Chen et al. (2021c). Each non-leaf node corresponds to a state with two actions transitioning (deterministically) to the left or right child respectively (and incurring 0 cost). The total number of nodes in the tree is

$$\sum_{i=0}^{B+1} 2^i = 2^{B+2} - 1 \leqslant 2^{\frac{\log S/2}{\log 2} + 1} - 1 = 2\frac{S}{2} - 1 = S - 1.$$

The total number of leaf nodes is $N = 2^{B+1} \geqslant \frac{S}{4}$. Denote the set of states corresponding to the leaf nodes by $\mathcal{S}_\ell = \big\{s_1, ..., s_N\big\}$. The root node is s_0. There is also one additional state denoted by f (recall that the number of states in the tree is $\leqslant S - 1$). We consider the same action set across each state: $\mathcal{A} = \big\{a_1, ..., a_{A-1}, a_f\big\}$. In the states of the binary tree where we

have only described two actions, we can consider the other actions to remain in the same state deterministically with 0 cost. The remaining transitions and costs are defined as follows:

- For $s_i \in S_\ell$: , and $a_j \in \mathcal{A}$
  - if $j = f$, then $p(f|s_i, a_f) = 1$ and $c_k(s_i, a_f) = 0$.
  - if $j \in \{1, ..., A-1\}$ and $j+(A-1)\cdot(i-1) \leqslant L$, then $p(g|s_i, a_j) = \frac{1}{T'}$, $p(s_i|s_i, a_j) = 1-\frac{1}{T'}$ and the cost is an independent sample from a Bernoulli distribution at each episode k: $c_k(s_i, a_j) \sim \text{Ber}\left(\frac{D'}{2T'}\right)$.
  - if $j \in \{1, ..., A-1\}$ and $j + (A-1)\cdot(i-1) > L$, then a_j is the same as a_f, i.e. $p(f|s_i, a_j) = 1$ and $c_k(s_i, a_j) = 0$.
- For state f,
  - $p(g|f, a_f) = \frac{1}{D'}$, $p(f|f, a_f) = 1 - \frac{1}{D'}$ and $c_k(f, a_f) = 1$.
  - for all $a_j \in \mathcal{A}\backslash\{a_f\}$, $p(f|f, a_j) = 1$ and $c_k(f, a_j) = 0$.

Denote the above distribution for c_k by $\mathcal{D}$. In each episode there are at most $L+1 \leqslant M$ non-zero costs, ensuring the condition on sparsity is respected.

Intuition: The MDP construction is essentially a reduction to a non-sparse experts problem with $\mathcal{O}(M)$ actions. First, there is a reduction to an experts problem with $NA = \mathcal{O}(SA)$ actions through the leaves of the tree. Then within these, there are $\mathcal{O}(M)$ good actions, while the remaining are bad. The good actions suffer small costs in expectation and can lead directly to the goal-state. The bad actions are zero-cost but all lead to the same unique bad state f, where only one action leads to the goal-state and suffers high cost. This allows a big proportion of the actions to be bad while still guaranteeing sparsity and forcing the learner to only consider the $\mathcal{O}(M)$ good actions, completing the reduction to the non-sparse experts problem with $\mathcal{O}(M)$ actions.

However, we cannot directly apply lower bounds for the experts problem because we need to take into account for hitting-time related quantities such as D and $T_\star$. We use a similar approach to the experts lower bounds by sampling the costs i.i.d. from a Bernoulli distribution, however with the scaled parameter D'/T'. The regret in this stochastic environment can then be expressed as the maximum of asymmetric zero-mean i.i.d. random walks, capturing how much better the

optimal policy can be by choosing the best action after the i.i.d. Bernoulli costs have been sampled for all episodes. The result will then follow from the below technical result on the expectation of this maximum. It is an extension of the lower bound of Orabona and Pal (2015) to asymmetric zero-mean random-walks. The proof is quite technical and can be found in Section 6.7.3.

Theorem 6.4.3. *Fix $p \in [\frac{1}{2}, 1 - \frac{1}{n}]$. Consider random walks $Z_i^{(n)} = \sum_{t=1}^n X_t^i$, where*

$$X_t^i = \begin{cases} -p, & \text{w.p. } 1-p \\ 1-p, & \text{w.p. } p. \end{cases}$$

If $n \geqslant 200 \frac{p}{1-p} \log d$ (also ensures that $p \leqslant 1 - \frac{1}{n}$) and $d \geqslant 100$. Then,

$$\mathbb{E}\big[\max_{1 \leqslant i \leqslant d} Z_i^n\big] \geqslant 0.02\sqrt{np(1-p)\log d} - 1.5.$$

Detailed Computations: For $i \in \{1, ..., N\}$, let $\mathcal{A}_i$ correspond to the actions in state $s_i \in \mathcal{S}_L$ which can transition directly to g and $\mathcal{A} \backslash \mathcal{A}_i$ corresponds to the actions which deterministically transition to f (e.g. if $(A-1) \cdot i \leqslant L$, then $\mathcal{A}_i = \{a_1, ..., a_{A-1}\}$). For any proper policy π independent of the stochastically generated costs in episode k, we have

$$\begin{aligned}
\mathbb{E}_{c_k \sim \mathcal{D}}\big[J_k^\pi(s_i)\big] &= \mathbb{E}_{c_k \sim \mathcal{D}}\Big[\sum_{a \in \mathcal{A}_i} \pi(a|s_i)\Big(c_k(s_i,a) + \Big(1 - \frac{1}{T'}\Big)J_k^\pi(s_i)\Big) + J_k^\pi(f)\sum_{a \notin \mathcal{A}_i}\pi(a|s_i)\Big] \\
&= \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\Big(\mathbb{E}_{c_k \sim \mathcal{D}}\big[c_k(s_i,a)\big] + \Big(1 - \frac{1}{T'}\Big)\mathbb{E}_{c_k \sim \mathcal{D}}\big[J_k^\pi(s_i)\big]\Big) + D' \cdot \sum_{a \notin \mathcal{A}_i}^{A} \pi(a|s_i) \\
&= \sum_{a \in \mathcal{A}_i} \pi(a|s_i)\Big(\frac{D'}{2T'} + \Big(1 - \frac{1}{T'}\Big)\mathbb{E}_{c_k \sim \mathcal{D}}\big[J_k^\pi(s_i)\big]\Big) + D' \cdot \Big(1 - \sum_{a \in \mathcal{A}_i}\pi(a|s_i)\Big) \\
\implies \mathbb{E}_{c_k \sim \mathcal{D}}\big[J_k^\pi(s_i)\big]\Big(1 - \Big(1 - \frac{1}{T'}\Big)\sum_{a \in \mathcal{A}_i}\pi(a|s_i)\Big) &= \frac{D'}{2T'}\sum_{a \in \mathcal{A}_i}\pi(a|s_i) + D' \cdot \Big(1 - \sum_{a \in \mathcal{A}_i}\pi(a|s_i)\Big) \\
\implies \mathbb{E}_{c_k \sim \mathcal{D}}\big[J_k^\pi(s_i)\big] &= \frac{\frac{D'}{2T'}\sum_{a \in \mathcal{A}_i}\pi(a|s_i) + D' \cdot \Big(1 - \sum_{a \in \mathcal{A}_i}\pi(a|s_i)\Big)}{1 - \Big(1 - \frac{1}{T'}\Big)\sum_{a \in \mathcal{A}_i}\pi(a|s_i)} \\
&= \frac{D'}{2} \cdot \frac{\frac{1}{T'}\sum_{a \in \mathcal{A}_i}\pi(a|s_i) + 2\Big(1 - \sum_{a \in \mathcal{A}_i}\pi(a|s_i)\Big)}{\frac{1}{T'}\sum_{a \in \mathcal{A}_i}\pi(a|s_i) + \Big(1 - \sum_{a \in \mathcal{A}_i}\pi(a|s_i)\Big)} \\
&\geqslant \frac{D'}{2}.
\end{aligned}$$

The optimal policy $\pi^\star$ is the policy that takes actions in the binary tree to reach state $s_{i^\star}$ and then $\pi^\star(a_{j^\star}|s_{i^\star}) = 1$ for $i^\star, j^\star = \arg\min_{i,j:j+(A-1)\cdot i \leqslant L} \sum_{k=1}^K c_k(s_i, a_j)$. We have $J_k^{\pi^\star}(s_0) = J_k^{\pi^\star}(s_{i^\star})$

and for any $k \geqslant 1$

$$\begin{aligned} J_k^{\pi^\star}(s_{i^\star}) &= c_k(s_{i^\star}, a_{j^\star}) + \Big(1 - \frac{1}{T'}\Big) J_k^{\pi^\star}(s_{i^\star}) \\ \implies J_k^{\pi^\star}(s_0) &= T' c_k(s_{i^\star}, a_{j^\star}) \\ \implies \sum_{k=1}^K J_k^{\pi^\star}(s_0) &= T' \sum_{k=1}^K c_k(s_{i^\star}, a_{j^\star}) = T' \min_{i,j: j+(A-1)\cdot i \leqslant L} \sum_{k=1}^K c_k(s_i, a_j). \end{aligned}$$

Hence,

$$\begin{aligned} \mathbb{E}_{c_1,\dots,c_K \overset{\text{iid}}{\sim} \mathcal{D}}\big[R_K\big] &\geqslant \frac{D'}{2} \cdot K - T' \cdot \mathbb{E}_{c_1,\dots,c_K \overset{\text{iid}}{\sim} \mathcal{D}}\Big[\min_{i,j: j+(A-1)\cdot i \leqslant L} \sum_{k=1}^K c_k(s_i, a_j)\Big] \\ &= T' \cdot \Big(\frac{D'}{2T'} \cdot K - \mathbb{E}_{c_1,\dots,c_K \overset{\text{iid}}{\sim} \mathcal{D}}\Big[\min_{i,j: j+(A-1)\cdot i \leqslant L} \sum_{k=1}^K c_k(s_i, a_j)\Big]\Big) \\ &= T' \cdot \mathbb{E}_{c_1,\dots,c_K \overset{\text{iid}}{\sim} \mathcal{D}}\Big[\max_{i,j: j+(A-1)\cdot i \leqslant L} \sum_{k=1}^K \Big(\frac{D'}{2T'} - c_k(s_i, a_j)\Big)\Big] \end{aligned}$$

We now apply Theorem 6.4.3 with $p = 1 - \frac{D'}{2T'} \geqslant \frac{1}{2}$, $d = L \geqslant 100$ (since $S(A-1) \geqslant 400$ and $M \geqslant 101$) and $n = K \geqslant \frac{800 T_\star}{D} \log M \geqslant \frac{400 T'}{D'} \log M \geqslant 200 \frac{p}{1-p} \log d$. We obtain:

$$\begin{aligned} \sup_{c_1,\dots,c_K} \mathbb{E}\big[R_K\big] \geqslant \mathbb{E}_{c_1,\dots,c_K \overset{\text{iid}}{\sim} \mathcal{D}}\big[R_K\big] &\geqslant 0.02 T' \sqrt{K\Big(1 - \frac{D'}{2T'}\Big) \cdot \frac{D'}{2T'} \cdot \log L} - 1.5 T' \\ &= \Omega\Big(\sqrt{K T_\star D \log M}\Big), \end{aligned}$$

since $L \geqslant M/8$. Note that since $T_\star \geqslant D$, the hitting-time of the fast-policy is $D' + B + 2 = D$ and the hitting time of the optimal is $T' + B + 1 = T_\star$, as required. This concludes the proof. □

6.4.2 Comparison to Kwon and Perchet (2016)

In the experts setting, we have $S = 1$ and the goal state is reached immediately from the initial state. In this case, the result of Kwon and Perchet (2016) is as follows.

Proposition 6.4.4. *Consider Online Mirror Descent with regulariser $\psi(x) = \langle x, \log x \rangle$ (negative entropy) and step size*

$$\eta = \log\Big(1 + \sqrt{2A \log A/(MK)}\Big).$$

Then the expected regret satisfies

$$\mathbb{E}\left[R_K\right] \leqslant \sqrt{\frac{2MK \log A}{A}} + \log A.$$

They also show that this bound is asymptotically optimal. Compared to the $\sqrt{K \log A}$ optimal regret in the non-sparse experts setting, this yields an improvement factor of M/A, which constitutes a polynomial improvement as A grows relative to M. In particular, when M is constant, the improvement is linear in $1/A$. This gain stems from the structure of the experts problem: when there are many actions but only a few incur non-zero loss at each round, the adversary has limited ability to force the learner to accumulate large loss or regret.

SSP is Fundamentally Harder. The inability to obtain a similar polynomial improvement in sparse stochastic shortest path (SSP) problems highlights that SSPs are fundamentally harder than the experts setting. Intuitively, this difficulty arises because the adversary can induce losses on many nominally zero-cost actions by forcing transitions to a single state with high cost. In this way, the cost of a few non-zero actions can be effectively propagated across many actions via the transition dynamics. This phenomenon is illustrated by the construction in the proof of Theorem 6.4.2.

Recovering the Experts Result via Theorem 6.4.1. Theorem 6.4.1 does not recover the M/A polynomial improvement in the special case of the experts setting when $S = 1$. However, such an improvement can be recovered using a regret bound of a slightly different form that involves the hitting time of the uniform policy. Consider an alternative notion of *state-level sparsity*:

$$M' := \max_k \max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \mathbb{I}\{c_k(s,a) > 0\} \leqslant A.$$

From Theorem 4 of Zhao et al. (2022), we obtain the following cumulative loss bound for a variant of OMD with negative entropy regularisation:

$$\mathbb{E}\left[R_K\right] \leqslant \widetilde{\mathcal{O}}\left(\sqrt{T_\star \sum_{k=1}^K J_k^{\pi^\star}}\right).$$

Using state-level sparsity, we can further bound

$$\begin{aligned}
\sum_{k=1}^K J_k^{\pi^\star} &= \min_\pi \sum_{k=1}^K \mathbb{E}\left[\sum_{t=1}^{I_\pi(s_0)} c_k(s^t, a^t) \,\middle|\, P, \pi, s^1 = s_0\right] \\
&\leqslant \sum_{k=1}^K \mathbb{E}\left[\sum_{t=1}^{I_{\pi_u}(s_0)} c_k(s^t, a^t) \,\middle|\, P, \pi_u, s^1 = s_0\right] \\
&= \sum_{k=1}^K \mathbb{E}\left[\sum_{t=1}^{I_{\pi_u}(s_0)} \frac{1}{A} \sum_{a \in \mathcal{A}} c_k(s^t, a) \,\middle|\, P, \pi_u, s^1 = s_0\right]
\end{aligned}$$

$$\leqslant \frac{KM'}{A}\,\mathbb{E}\big[I_{\pi_u}(s_0) \mid P, \pi_u, s^1 = s_0\big]$$
$$= \frac{KM'}{A}\,T^u(s_0),$$

where π_u denotes the uniform policy ($\pi_u(a \mid s) = 1/A$) and $T^u(s_0)$ is its corresponding hitting time. This yields the regret bound

$$\mathbb{E}\left[R_K\right] \leqslant \widetilde{\mathcal{O}}\left(\sqrt{KT_\star \frac{M'}{A} T^u(s_0)}\right).$$

Since $M' \leqslant M$, this bound can be expressed in terms of the original sparsity parameter M whenever $M \leqslant A$, giving

$$\mathbb{E}\left[R_K\right] \leqslant \widetilde{\mathcal{O}}\left(\sqrt{KT_\star \frac{M}{A} T^u(s_0)}\right).$$

This result shows that polynomial improvements from sparsity are possible when state-level sparsity is present or when $M < A$. However, this comes at the cost of an additional factor $T^u(s_0)$, which in the worst case can offset the polynomial gain. Understanding structural conditions of the MDP under which this factor remains small is an interesting direction for future work.

In the experts setting ($S = 1$), we have $M = M'$ and $T^u(s_0) = 1$, and the above bound recovers the M/A improvement of Kwon and Perchet (2016). This further illustrates that the requirement to reach a goal state in SSPs introduces a fundamental distinction from the experts setting, even when costs are sparse. It also provides some further insights into the performance of OMD with negative entropy regularisation and that in particular issues arise when there is at least one state with non-sparse costs even though most other states may have sparse costs.

6.5 Related Works

Regret minimisation for SSP problems under full-information feedback was initiated by a line of work studying stochastic costs Tarbouriech et al. (2020); Rosenberg et al. (2020); Tarbouriech et al. (2021); Cohen et al. (2021); Chen et al. (2021b, 2022a); Jafarnia-Jahromi et al. (2023). In the adversarial setting, it was first studied by Rosenberg and Mansour (2021) in the known transition case. Their bounds were later improved by Chen et al. (2021c) (see Section 6.4 for a detailed discussion of the bounds of both of these works). There have since been many extensions: Chen and Luo (2021) consider the unknown transition setting, Zhao et al. (2022) establish dynamic regret bounds, Chen et al. (2022b) consider a policy optimisation approach in

the unknown transition setting.

Regret minimisation for SSP problems under bandit feedback where only the costs of the visited state-action pairs in an episode are revealed to the learner has also been studied both in the stochastic Chen et al. (2022b) and adversarial settings Chen et al. (2021c); Chen and Luo (2021); Chen et al. (2022b). In the adversarial known transition setting, the minimax regret is of the order $\sqrt{KDT_\star SA}$ (ignoring log terms) Chen et al. (2021c). It is an interesting future direction to study the sparse SSP problem with bandit feedback and understand if the regret scales with $\sqrt{M}$ instead of $\sqrt{SA}$, in which case M would play the same role of effective dimension as in the setting we consider.

Regret minimisation with sparse costs was studied in the classical online learning setting by Kwon and Perchet (2016) ($S = 1$); see Section 6.4.2 for a detailed discussion. The minimax regret goes from $\mathcal{O}(\sqrt{K \log A})$ to $\mathcal{O}(\sqrt{KMA^{-1} \log A})$ under full-information feedback (experts problem). For rewards instead of costs, it goes from $\mathcal{O}(\sqrt{K \log A})$ to $\mathcal{O}(\sqrt{K \log M})$, which matches the benefits of sparsity we establish for the SSP problem with costs. Note that we restrict our focus to costs since it is unclear how to interpret rewards within the SSP framework. Under bandit feedback, the sparse minimax regret goes from $\widetilde{\mathcal{O}}(\sqrt{KA})$ to $\widetilde{\mathcal{O}}(\sqrt{KM})$ for both rewards and costs Kwon and Perchet (2016); Bubeck et al. (2018). The above minimax regrets can also be achieved by sparse-agnostic methods Kwon and Perchet (2016); Tsuchiya et al. (2023). Finally, sparsity was also considered in the case of stochastic losses by Kwon et al. (2017).

6.6 Conclusion

In this chapter, we studied the SSP problem under sparse adversarial costs, known transitions and full-information feedback and characterised the sparse minimax regret achieved by Online Mirror Descent. In particular, we have shown that sparsity is a low-dimensional structure that can be leveraged to efficiently solve high-dimensional SSP problems and for which gradient-based methods are optimal.

Our results open up many further directions of research. We established the benefits of sparsity under known transition as limited to logarithmic, however, there could be structural properties of an MDP that could break this logarithmic limit and achieve polynomial benefits. Moreover,

we have limited our focus to the adversarial full-information feedback setting, but the study of sparse SSP problems in other settings, such as partial feedback, stochastic environments, or structured decision problems remains unexplored.

6.7 Missing Details

6.7.1 Efficient Implementation of OMD using our Regulariser

In this section, we describe how the OMD update with our regulariser from Section 6.4 defined in (6.6) can be computed efficiently. This closely follows Appendix B.1 of Rosenberg and Mansour (2021), who provide a similar description for the negative entropy.

Recall the regulariser $\psi_p(q) = p\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} q(s,a)^{1+1/p} - p$ for $q \in \mathbb{R}^{\Gamma}_{\geqslant 0}$. We have

$$\nabla\psi_p(q) = (p+1)\cdot q(s,a)^{1/p}.$$

The Bregman divergence is defined as:

$$\begin{aligned}
D_{\psi_p}(q,q') &= \psi_p(q) - \psi_p(q') - \langle \nabla\psi_p(q'), q - q'\rangle \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{p\cdot q(s,a)^{1+1/p} - p\cdot q'(s,a)^{1+1/p}\Big\} \\
&\quad - (p+1)\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q'(s,a)^{1/p}q(s,a) - q'(s,a)^{1+1/p}\Big\} \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q'(s,a)^{1+1/p} + q(s,a)\cdot\big[p\cdot q(s,a)^{1/p} - (p+1)\cdot q'(s,a)^{1/p}\big]\Big\}.
\end{aligned}$$

Recall that OMD with the above regulariser computes the occupancy measures as follows - see (6.1):

$$q_1 = \operatorname*{arg\,min}_{q\in\Delta(T)}\psi_p(q), \qquad q_{k+1} = \operatorname*{arg\,min}_{q\in\Delta(T)}\Big\{\eta\cdot\langle q, c_k\rangle + D_{\psi_p}(q,q_k)\Big\}.$$

As shown in Orabona (2019) (Theorem 6.15), each of these steps can be split into an unconstrained minimization step, and a projection step. Thus, q_1 can be computed as follows

$$\begin{aligned}
q_1' &= \operatorname*{arg\,min}_{q\in\mathbb{R}^{\Gamma}_{\geqslant 0}}\psi_p(q) \\
q_1 &= \operatorname*{arg\,min}_{q\in\Delta(T)} D_{\psi_p}(q,q_1'),
\end{aligned}$$

where q'_1 has a closed-from solution $q'_1(s,a) = 1$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$. Similarly, q_{k+1} is computed as follows for every $k = 1, ..., K-1$:

$$q'_{k+1} = \arg\min_{q \in \mathbb{R}^{\Gamma}_{\geqslant 0}} \Big\{ \eta \cdot \langle q, c_k \rangle + D_{\psi_p}(q, q_k) \Big\}$$
$$q_{k+1} = \arg\min_{q \in \Delta(T)} D_{\psi_p}(q, q'_{k+1}),$$

where again q'_{k+1} has a closed-from solution $q'_{k+1}(s,a) = \Big[q_k(s,a)^{1/p} - \frac{\eta}{p+1} c_k(s,a) \Big]_+^p$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$ (follows from straightforwardly differentiating above objective and setting to 0 and accounting for the non-negativity of occupancy measures) - we use notation $a_+ = \max\big\{0, a\big\}$. For the projection step, we start by formulating it as a constrained convex optimization problem:

$$\min_{q \in \mathbb{R}^{\Gamma}} D_{\psi_p}(q, q'_{k+1}) \quad \text{s.t.} \quad \sum_{a \in \mathcal{A}} q(s,a) - \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} P(s'|s,a) q(s',a') = \mathbb{I}\big\{s = s_0\big\} \qquad \forall s \in \mathcal{S}$$
$$\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s,a) \leqslant T$$
$$q(s,a) \geqslant 0 \qquad \forall (s,a) \in \mathcal{S} \times \mathcal{A}.$$

The problem can be solved by considering the Lagrangian with Lagrange multipliers λ and $\big\{v(s)\big\}_{s \in \mathcal{S}}$:

$$\mathcal{L}(q, \lambda, v) = D_{\psi_p}(q, q'_{k+1}) + \lambda \Big(\sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s,a) - T \Big) + \sum_s v(s) \Big(\sum_{s', a'} P(s|s', a') q(s', a') + \mathbb{I}\big\{s = s_0\big\} - \sum_a q(s,a) \Big)$$
$$= D_{\psi_p}(q, q'_{k+1}) + \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q(s,a) \Big(\lambda + \sum_{s' \in \mathcal{S}} P(s'|s,a) v(s') - v(s) \Big) + v(s_0) - \lambda T,$$

Differentiating the Lagrangian with respect to any $q(s,a)$ and setting to 0, we get

$$\frac{\partial \mathcal{L}(q, \lambda, v)}{\partial q(s,a)} = \nabla \psi_p(q)(s,a) - \nabla \psi_p(q'_{k+1})(s,a) + \lambda + \sum_{s' \in \mathcal{S}} P(s'|s,a) v(s') - v(s)$$
$$= (p+1) q(s,a)^{1/p} - (p+1) q'_{k+1}(s,a)^{1/p} + \lambda + \sum_{s' \in \mathcal{S}} P(s'|s,a) v(s') - v(s) = 0.$$
$$\implies q_{k+1}(s,a) = \Big[q'_{k+1}(s,a)^{1/p} - \frac{\lambda + \sum_{s' \in \mathcal{S}} P(s'|s,a) v(s') - v(s)}{p+1} \Big]_+^p.$$

This formula is also valid for $k = 0$ by setting $c_0(s,a) = 0$ and $q_0(s,a) = 1$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

To compute the value of λ and v at the optimum, we write the dual problem $\mathcal{D}(\lambda, v) =$

$\min_q \mathcal{L}(q, \lambda, v)$ by substituting q_{k+1} back into $\mathcal{L}$:

$$\mathcal{D}(\lambda, v) = D_{\psi_p}(q_{k+1}, q'_{k+1}) + \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} q_{k+1}(s,a)\Big(\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)\Big) + v(s_0) - \lambda T.$$

Recall that $q'_{k+1}(s,a) = \Big[q_k(s,a)^{1/p} - \frac{\eta}{p+1}c_k(s,a)\Big]_+^p$, so (ignoring terms independent of λ, v, e.g. $q'_{k+1}(s,a)$):

$$\begin{aligned}
D_{\psi_p}(q_{k+1}, q'_{k+1}) &= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q'_{k+1}(s,a)^{1+1/p} + q_{k+1}(s,a)\cdot\big[pq_{k+1}(s,a)^{1/p} - (p+1)q'_{k+1}(s,a)^{1/p}\big]\Big\} \\
&\propto \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q_{k+1}(s,a)\cdot\Big[p\Big(q'_{k+1}(s,a)^{1/p} - \frac{\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)}{p+1}\Big)_+ \\
&\qquad - (p+1)q'_{k+1}(s,a)^{1/p}\Big]\Big\} \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q_{k+1}(s,a)\cdot\Big[p\Big(q'_{k+1}(s,a)^{1/p} - \frac{\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)}{p+1}\Big) \\
&\qquad - (p+1)q'_{k+1}(s,a)^{1/p}\Big]\Big\} \quad \text{since if } q_{k+1}(s,a) = 0, \text{ then the whole term is } 0 \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q_{k+1}(s,a)\cdot\Big[-q'_{k+1}(s,a)^{1/p} - \frac{p}{p+1}\big(\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)\big)\Big]\Big\} \\
\implies \mathcal{D}(\lambda, v) &\propto \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q_{k+1}(s,a)\cdot\Big[-q'_{k+1}(s,a)^{1/p} - \frac{p}{p+1}\big(\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)\big) \\
&\qquad + \big(\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)\big)\Big]\Big\} + v(s_0) - \lambda T \\
&= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big\{q_{k+1}(s,a)\cdot\Big[-q'_{k+1}(s,a)^{1/p} + \frac{\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)}{p+1}\Big]\Big\} + v(s_0) - \lambda T \\
&= -\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} q_{k+1}(s,a)^{1+1/p} + v(s_0) - \lambda T \\
&= -\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big[q'_{k+1}(s,a)^{1/p} - \frac{\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)}{p+1}\Big]_+^{1+p} + v(s_0) - \lambda T.
\end{aligned}$$

Maximizing the dual gives λ and v or equivalently, we can minimize the negation of the dual:

$$\lambda_{k+1}, v_{k+1} = \operatorname*{arg\,min}_{\lambda\geqslant 0, v} \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\Big[q'_{k+1}(s,a)^{1/p} - \frac{\lambda + \sum_{s'\in\mathcal{S}} P(s'|s,a)v(s') - v(s)}{p+1}\Big]_+^{1+p} - v(s_0) + \lambda T.$$

This is a convex optimization problem subject only to non-negativity constraints, and can be efficiently solved using iterative methods (e.g. gradient descent).

6.7.2 Additional Computations for the Proof of Theorem 6.3.1

In this section, we provide the details of the compututation of the occupancy measures played by OMD with the negative entropy on the SSP instance described in the Proof of Theorem 6.3.1

for all episodes, using the convex optimization problem:

$$q_{k+1}(s,a) = q_k(s,a)e^{B_k^{v_{k+1}}(s,a)}, \quad \text{where} \quad B_k^v(s,a) = v(s) - \eta c_k(s,a) - \sum_{s' \in \mathcal{S}} p(s'|s,a)v(s'),$$
$$v_{k+1} = \arg\min_v \mathcal{D}_k(v),$$
$$\mathcal{D}_k(v) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q_k(s,a)e^{B_k^v(s,a)} - v(s_0),$$

with $q_0(s,a) = 1$ and $c_0(s,a) = 0$.

We begin by computing expressions for $B_k^v(s,a)$ in each state:

- $B_k^v(s_0,a) = v(s_0) - \frac{1}{2}v(s_0^L) - \frac{1}{2}v(s_0^R)$ for all $a \in \mathcal{A}$
- $B_k^v(s_0^L,a) = v(s_0^L) - \eta c_k(s_0^L,a) - v(s_g^L)$ for all $a \in \mathcal{A}$
- $B_k^v(s_g^L) = v(s_g^L)$
- $B_k^v(s_0^R,a_1) = v(s_0^R) - v(s_g^R)$
- $B_k^v(s_0^R,a_2) = v(s_0^R) - \eta c_k(s_0^R,a_2) - \frac{1}{N}\sum_{i=1}^N v(s_i^R) = v(s_0^R) - \eta c_k(s_0^R,a_2) - v(s_1^R)$ since by symmetry $v(s_i^R) = v(s_1^R)$ for all $i \geqslant 1$ and any v solving the convex optimization problem specified in the OMD update.
- $B_k^v(s_i^R,a) = v(s_i^R) = v(s_1^R)$ for all $a \in \mathcal{A}$
- $B_k^v(s_g^R,a) = v(s_g^R)$ for all $a \in \mathcal{A}$

Plugging these into the optimization problem, we obtain (recall the notation $q(s) = \sum_{a \in \mathcal{A}} q(s,a)$):

$$\begin{aligned}
v_{k+1} = \arg\min_v \quad & \mathcal{D}_k(v) = \arg\min_v \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} q_k(s,a)e^{B_k^v(s,a)} - v(s_0) \\
= \arg\min_v \quad & q_k(s_0,a_1)e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} + q_k(s_0,a_2)e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} \\
& + q_k(s_0^L,a_1)e^{v(s_0^L)-\eta c_k(s_0^L,a_1)-v(s_g^L)} + q_k(s_0^L,a_2)e^{v(s_0^L)-\eta c_k(s_0^L,a_2)-v(s_g^L)} \\
& + q_k(s_g^L,a_1)e^{v(s_g^L)} + q_k(s_g^L,a_2)e^{v(s_g^L)} \\
& + q_k(s_0^R,a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R,a_2)e^{v(s_0^R)-\eta c_k(s_0^R,a_2)-v(s_1^R)} \\
& + \sum_{i=1}^N \left\{ q_k(s_i^R,a_1)e^{v(s_1^R)} + q_k(s_i^R,a_2)e^{v(s_1^R)} \right\} \\
& + q_k(s_g^R,a_1)e^{v(s_g^R)} + q_k(s_g^R,a_2)e^{v(s_g^R)} \\
& - v(s_0)
\end{aligned}$$

$$
\begin{aligned}
= \arg\min_{v} \quad & q_k(s_0)e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} \\
& + q_k(s_0^L,a_1)e^{v(s_0^L)-\eta c_k(s_0^L,a_1)-v(s_g^L)} + q_k(s_0^L,a_2)e^{v(s_0^L)-\eta c_k(s_0^L,a_2)-v(s_g^L)} \\
& + q_k(s_g^L)e^{v(s_g^L)} \\
& + q_k(s_0^R,a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R,a_2)e^{v(s_0^R)-\eta c_k(s_0^R,a_2)-v(s_1^R)} \\
& + \sum_{i=1}^{N}\left\{q_k(s_i^R)e^{v(s_1^R)}\right\} \\
& + q_k(s_g^R)e^{v(s_g^R)} \\
& - v(s_0).
\end{aligned}
$$

This being a convex optimization problem, it can be solved by differentiating and setting to 0:

$$
\begin{aligned}
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0)} &= q_k(s_0)e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} - 1 = 0 \\
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} &= -0.5q_k(s_0)e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} + q_k(s_0^L,a_1)e^{v(s_0^L)-\eta c_k(s_0^L,a_1)-v(s_g^L)} + q_k(s_0^L,a_2)e^{v(s_0^L)-\eta c_k(s_0^L,a_2)-v(s_g^L)} \\
&= -0.5 + q_k(s_0^L,a_1)e^{v(s_0^L)-\eta c_k(s_0^L,a_1)-v(s_g^L)} + q_k(s_0^L,a_2)e^{v(s_0^L)-\eta c_k(s_0^L,a_2)-v(s_g^L)} = 0 \\
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^L)} &= -q_k(s_0^L,a_1)e^{v(s_0^L)-\eta c_k(s_0^L,a_1)-v(s_g^L)} - q_k(s_0^L,a_2)e^{v(s_0^L)-\eta c_k(s_0^L,a_2)-v(s_g^L)} + q_k(s_g^L)e^{v(s_g^L)} \\
&= q_k(s_g^L)e^{v(s_g^L)} - 0.5 = 0 \\
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} &= -0.5q_k(s_0)e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} + q_k(s_0^R,a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R,a_2)e^{v(s_0^R)-\eta c_k(s_0^R,a_2)-v(s_1^R)} \\
&= -0.5 + q_k(s_0^R,a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R,a_2)e^{v(s_0^R)-\eta c_k(s_0^R,a_2)-v(s_1^R)} = 0 \\
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} &= -q_k(s_0^R,a_2)e^{v(s_0^R)-\eta c_k(s_0^R,a_2)-v(s_1^R)} + e^{v(s_1^R)}\sum_{i=1}^{N} q_k(s_i^R) = 0 \\
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} &= -q_k(s_0^R,a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_g^R)e^{v(s_g^R)} = 0.
\end{aligned}
$$

Let's look specifically at the case $k = 0$ ($q_0(s,a) = 1, q_0(s) = 2, c_0(s,a) = 0$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$). For the left part of the MDP we have:

$$
\begin{aligned}
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} = 0 \implies 2e^{v(s_0^L)-v(s_g^L)} = 0.5 \implies e^{v(s_0^L)} = 0.25e^{v(s_g^L)} \\
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^L)} = 0 \implies e^{v(s_g^L)} = 0.25 \implies e^{v(s_0^L)} = 0.25^2 \\
&\qquad\qquad \implies q_1(s_0^L,a) = e^{B_0^v(s_0^L,a)} = \frac{0.25^2}{0.25} = 0.25, \text{ for all } a \in \mathcal{A}.
\end{aligned}
$$

For the right part of the MDP, we have:

$$\begin{aligned}
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} = 0 \implies e^{v(s_0^R)-v(s_g^R)} + e^{v(s_0^R)-v(s_1^R)} = 0.5 \implies e^{v(s_0^R)} = \frac{0.5}{e^{-v(s_g^R)} + e^{-v(s_1^R)}} \\
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} = 0 \implies e^{v(s_0^R)-v(s_1^R)} = 2Ne^{v(s_1^R)} \implies e^{v(s_1^R)} = \frac{1}{\sqrt{2N}} e^{0.5v(s_0^R)} \\
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} = 0 \implies e^{v(s_0^R)-v(s_g^R)} = 2e^{v(s_g^R)} \implies e^{v(s_g^R)} = \frac{1}{\sqrt{2}} e^{0.5v(s_0^R)} \\
&\qquad \implies e^{v(s_0^R)} = \frac{0.5}{\sqrt{2}e^{-0.5v(s_0^R)} + \sqrt{2N}e^{-0.5v(s_0^R)}} \\
&\qquad \implies e^{0.5v(s_0^R)} = \frac{0.5}{\sqrt{2} + \sqrt{2N}} \\
&\qquad \implies e^{v(s_0^R)} = \frac{0.25}{(\sqrt{2} + \sqrt{2N})^2} \\
&\qquad \implies q_1(s_0^R, a_1) = e^{B_0^v(s_0^R, a_1)} = e^{v(s_0^R)-v(s_g^R)} = \sqrt{2}e^{0.5v(s_0^R)} = \frac{0.5}{1+\sqrt{N}} \\
&\qquad\quad q_1(s_0^R, a_2) = e^{B_0^v(s_0^R, a_2)} = e^{v(s_0^R)-v(s_1^R)} = \sqrt{2N}e^{0.5v(s_0^R)} = \frac{0.5\sqrt{N}}{1+\sqrt{N}} \\
&\qquad\quad q_1(s_1^R, a) = e^{B_0^v(s_1^R, a)} = e^{v(s_1^R)} = \frac{1}{\sqrt{2N}} e^{0.5v(s_0^R)} = \frac{0.25}{\sqrt{N}(1+\sqrt{N})} \\
&\qquad\quad q_1(s_g^R, a) = e^{B_0^v(s_g^R, a)} = e^{v(s_g^R)} = \frac{1}{\sqrt{2}} e^{0.5v(s_0^R)} = \frac{0.25}{1+\sqrt{N}}.
\end{aligned}$$

Let's now look at general $k \geqslant 1$: Since q_k is an occupancy measure, it satisfies the properties of the dynamics of the MDP (see the definition of $\Delta(T)$ in Section 6.1) and we have that for any $s \in \mathcal{S}$: $\sum_{a\in\mathcal{A}} q_k(s,a) = \sum_{s'\in\mathcal{S}} \sum_{a'\in\mathcal{A}} p(s|s',a')q_k(s',a') + \mathbb{I}\{s = s_0\}$. In particular, this gives

- $s = s_0$: $q_k(s_0) = 1$
- $s = s_0^L$: $q_k(s_0^L) = 0.5q_k(s_0) = 0.5$
- $s = s_g^L$: $q_k(s_g^L) = q_k(s_0^L) = 0.5$
- $s = s_0^R$: $q_k(s_0^R) = 0.5q_k(s_0) = 0.5$
- $s = s_g^R$: $q_k(s_g^R) = q_k(s_0^R, a_1)$
- $s = s_1^R$: $q_k(s_1^R) = \frac{1}{N} q_k(s_0^R, a_2)$

This leads to the following simplifications in the derivatives of $\mathcal{D}_k(v)$:

$$\begin{aligned}
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0)} = e^{v(s_0)-0.5v(s_0^L)-0.5v(s_0^R)} - 1 = 0 \\
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} = -0.5 + q_k(s_0^L, a_1)e^{v(s_0^L)-\eta c_k(s_0^L,a_1)-v(s_g^L)} + (1/2 - q_k(s_0^L, a_1))e^{v(s_0^L)-\eta c_k(s_0^L,a_2)-v(s_g^L)} = 0
\end{aligned}$$

$$\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^L)} = 0.5e^{v(s_g^L)} - 0.5 = 0 \implies e^{v(s_g^L)} = 1$$
$$\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} = -0.5 + q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} + (1/2 - q_k(s_0^R, a_1))e^{v(s_0^R)-\eta-v(s_1^R)} = 0$$
$$\frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} = -q_k(s_0^R, a_2)e^{v(s_0^R)-\eta-v(s_1^R)} + q_k(s_0^R, a_2)e^{v(s_1^R)} = 0 \implies e^{v(s_0^R)} = e^{\eta+2v(s_1^R)}$$
$$\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} = -q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} + q_k(s_0^R, a_1)e^{v(s_g^R)} = 0 \implies e^{v(s_0^R)} = e^{2v(s_g^R)}.$$

Left part of the MDP:

$$\begin{aligned}
\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^L)} = 0 &\implies q_k(s_0^L, a_1)e^{v(s_0^L)-\eta\frac{1+(-1)^k}{2}} + (0.5 - q_k(s_0^L, a_1))e^{v(s_0^L)-0.5\eta} = 0.5 \\
&\implies e^{v(s_0^L)} = \frac{0.5}{q_k(s_0^L, a_1)e^{-\eta\frac{1+(-1)^k}{2}} + (0.5 - q_k(s_0^L, a_1))e^{-0.5\eta}} \\
k+1 = 2 &\implies e^{v(s_0^L)} = \frac{0.5}{0.25 + 0.25e^{-0.5\eta}} \\
&\implies q_2(s_0^L, a_1) = q_1(s_0^L, a_1)e^{B_1^v(s_0^L, a_1)} = 0.25e^{v(s_0^L)-v(s_g^L)} = \frac{0.5}{1 + e^{-0.5\eta}} \\
k+1 = 3 &\implies e^{v(s_0^L)} = \frac{0.5}{q_1(s_0^L, a_1)e^{-\eta} + (0.5 - q_1(s_0^L, a_1))e^{-0.5\eta}} \\
&\implies e^{v(s_0^L)} = \frac{0.5}{\frac{0.5}{1+e^{-0.5\eta}}e^{-\eta} + \frac{0.5e^{-0.5\eta}}{1+e^{-0.5\eta}}e^{-0.5\eta}} = 0.25\frac{e^{\eta}}{q_2(s_0^L, a_1)} \\
&\implies q_3(s_0^L, a_1) = q_2(s_0^L, a_1)e^{v(s_0^L)-\eta-v(s_g^L)} = 0.25 \\
k \text{ even} &\implies q_k(s_0^L, a_1) = \frac{0.5}{1 + e^{-0.5\eta}} \\
k \text{ odd} &\implies q_k(s_0^L, a_1) = 0.25,
\end{aligned}$$

where the last two lines follow by a straightforward induction.

Right part of the MDP:

$$\begin{aligned}
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_0^R)} = 0 \implies e^{v(s_0^R)} = \frac{0.5}{q_k(s_0^R, a_1)e^{-v(s_g^R)} + (1/2 - q_k(s_0^R, a_1))e^{-\eta-v(s_1^R)}} \\
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_1^R)} = 0 \implies v(s_0^R) = \eta + 2v(s_1^R) \\
&\frac{\partial \mathcal{D}_k(v)}{\partial v(s_g^R)} = 0 \implies v(s_0^R) = v(s_g^R) \\
&\implies e^{v(s_0^R)} = \frac{0.5}{q_k(s_0^R, a_1)e^{-0.5v(s_0^R)} + (1/2 - q_k(s_0^R, a_1))e^{-\eta-0.5v(s_0^R)+0.5\eta}} \\
&\implies e^{0.5v(s_0^R)} = \frac{0.5}{q_k(s_0^R, a_1) + (1/2 - q_k(s_0^R, a_1))e^{-0.5\eta}} = \frac{0.5}{q_k(s_0^R, a_1) + q_k(s_0^R, a_2)e^{-0.5\eta}} \\
&\implies q_{k+1}(s_0^R, a_1) = q_k(s_0^R, a_1)e^{B_k^v(s_0^R, a_1)} = q_k(s_0^R, a_1)e^{v(s_0^R)-v(s_g^R)} = q_k(s_0^R, a_1)e^{0.5v(s_0^R)} \\
&\implies q_{k+1}(s_0^R, a_1) = 0.5\frac{q_k(s_0^R, a_1)}{q_k(s_0^R, a_1) + (1/2 - q_k(s_0^R, a_1))e^{-0.5\eta}}
\end{aligned}$$

$$\implies \frac{q_{k+1}(s_0^R, a_1)}{q_{k+1}(s_0^R, a_2)} = \frac{q_{k+1}(s_0^R, a_1)}{(1/2 - q_{k+1}(s_0^R, a_1))} = \frac{q_k(s_0^R, a_1)}{(1/2 - q_k(s_0^R, a_1))e^{-0.5\eta}} = e^{0.5\eta}\frac{q_k(s_0^R, a_1)}{q_k(s_0^R, a_2)}$$
$$\implies \frac{q_{k+1}(s_0^R, a_1)}{q_{k+1}(s_0^R, a_2)} = e^{0.5k\eta}\frac{q_1(s_0^R, a_1)}{q_1(s_0^R, a_2)} = e^{0.5k\eta}\frac{0.5}{0.5\sqrt{N}} = \frac{1}{\sqrt{N}}e^{0.5k\eta}$$
$$\implies q_{k+1}(s_0^R, a_2) = \frac{0.5}{1 + \frac{1}{\sqrt{N}}e^{0.5k\eta}} = \frac{0.5\sqrt{N}}{\sqrt{N} + e^{0.5\eta k}}.$$

6.7.3 Lower Bound on the Maximum of Asymmetric Zero-Mean Random Walks

We extend the lower bound of Orabona and Pal (2015) to asymmetric zero-mean random-walks. We consider $p \geqslant 1/2$ because it simplifies the proof below (lower-bounding ψ by 1 and upper-bounding C in proof below) and is what we need in the proof of Theorem 6.4.2 (we use $p = 1 - D/2T^\star$). We re-state the theorem here.

Theorem 6.7.1. *Fix $p \in [\frac{1}{2}, 1 - \frac{1}{n}]$. Consider random walks $Z_i^{(n)} = \sum_{t=1}^n X_t^i$, where*

$$X_t^i = \begin{cases} -p, & w.p.\ 1-p \\ 1-p, & w.p.\ p. \end{cases}$$

If $n \geqslant 200\frac{p}{1-p}\log d$ (also ensures that $p \leqslant 1 - \frac{1}{n}$) and $d \geqslant 100$. Then,

$$\mathbb{E}\big[\max_{1 \leqslant i \leqslant d} Z_i^n\big] \geqslant 0.02\sqrt{np(1-p)\log d} - 1.5.$$

Proof. We follow the same lines as Orabona and Pal (2015) who show a special case of the result for $p = 1/2$. We generalize it to $p > 1/2$.

Consider $Z^{(n)} = \sum_{t=1}^n X_t$, a random-walk of length n, then $B_n = Z^{(n)} + pn \sim B(n, p)$, Binomial distribution with parameters n and p.

1st Part of the Proof:

The 1st part of the proof is all about providing a lower bound on $\mathbb{P}\big(B_n \geqslant pn + t - 1\big)$ in (6.10) for any $t \in [1, np + 1]$.

Lemma 6.7.2 (Generalized version of Lemma 4 of Orabona and Pal (2015), Theorem 2 of McKay (1989))**.** *Let n, k be integers satisfying $n \geqslant 1$ and $pn \leqslant k \leqslant n$. Define $x = \frac{k - pn}{\sqrt{p(1-p)n}}$.*

Then $B_n \sim B(n,p)$ satisfies

$$\mathbb{P}\big(B_n \geqslant k\big) \geqslant \sqrt{n}\binom{n-1}{k-1} p^{k-1/2}(1-p)^{n-k+1/2} \cdot \frac{1-\Phi(x)}{\phi(x)},$$

where $\phi(x)$ is the PDF of a standard Normal and $\Phi(x)$ is the CDF.

Denote $D(p,q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$ as the KL-divergence between two Bernoullis.

Lemma 6.7.3 (Generalized version of Theorem 5 of Orabona and Pal (2015))**.** *Let n, k be integers satisfying $n \geqslant 1$, and $np \leqslant k \leqslant n$. Define $x = \frac{k-pn}{\sqrt{p(1-p)n}}$. Then $B_n \sim B(n,p)$ satisfies*

$$\mathbb{P}\big(B_n \geqslant k\big) \geqslant \frac{\exp\Big(-nD\big(\frac{k}{n},p\big)\Big)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{1-\Phi(x)}{\phi(x)}.$$

Proof. For $k = n$, we verify the statement of the theorem directly. The left hand side is $\mathbb{P}\big(B_n \geqslant n\big) = p^n$. The right hand side is smaller because $\exp\Big\{-nD\Big(1,p\Big)\Big\} = p^n$ and for $x = \sqrt{n\frac{1-p}{p}} > 0$, we have $\frac{1-\Phi(x)}{\phi(x)} \leqslant \sqrt{2}$ (see e.g. Section 3.3 in Gasull and Utzet (2014)).

For $np \leqslant k < n$, we first bound the binomial coefficient $\binom{n}{k}$. Stirling's formula for the factorial Robbins (1955) gives for any $n \geqslant 1$,

$$\sqrt{2\pi n}\Big(\frac{n}{e}\Big)^n < n! < e^{1/12}\sqrt{2\pi n}\Big(\frac{n}{e}\Big)^n.$$

Since $0 < np \leqslant k \leqslant n-1$, we can use this approximation for k, n and $n-k$ and obtain

$$\begin{aligned}
\binom{n}{k} &= \frac{n!}{k!(n-k)!} \\
&> \frac{n^n e^{-n}\sqrt{2\pi n}}{(e^{1/12}k^k e^{-k}\sqrt{2\pi k}) \cdot (e^{1/12}(n-k)^{n-k}e^{-(n-k)}\sqrt{2\pi(n-k)})} \\
&= \frac{1}{e^{1/6}\sqrt{2\pi}}\Big(\frac{n}{k}\Big)^k\Big(\frac{n}{n-k}\Big)^{n-k}\sqrt{\frac{n}{k(n-k)}} \\
&= \frac{1}{e^{1/6}\sqrt{2\pi}}\frac{1}{p^k(1-p)^{n-k}}\exp\Big\{-nD\Big(\frac{k}{n},p\Big)\Big\}\sqrt{\frac{n}{k(n-k)}},
\end{aligned}$$

since

$$\begin{aligned}
D\Big(\frac{k}{n},p\Big) &= \frac{k}{n}\log\Big(\frac{k}{np}\Big) + \Big(1-\frac{k}{n}\Big)\log\Big(\frac{1-k/n}{1-p}\Big) = -\frac{k}{n}\log\Big(\frac{np}{k}\Big) - \frac{n-k}{n}\log\Big(\frac{n(1-p)}{n-k}\Big) \\
\implies \exp\Big\{-nD\Big(\frac{k}{n},p\Big)\Big\} &= \Big(\frac{np}{k}\Big)^k \cdot \Big(\frac{n(1-p)}{n-k}\Big)^{n-k} = p^k(1-p)^{n-k}\Big(\frac{n}{k}\Big)^k\Big(\frac{n}{n-k}\Big)^{n-k}.
\end{aligned}$$

Since $k \geqslant 1$, we can write the binomial coefficient as $\binom{n-1}{k-1} = \frac{k}{n}\binom{n}{k}$. By Lemma 6.7.2, we have

$$\begin{aligned}
\mathbb{P}\big(B_n \geqslant k\big) &\geqslant \sqrt{n}\binom{n-1}{k-1}p^{k-1/2}(1-p)^{n-k+1/2} \cdot \frac{1-\Phi(x)}{\phi(x)} \\
&= \sqrt{n}\frac{k}{n}\binom{n}{k}p^{k-1/2}(1-p)^{n-k+1/2} \cdot \frac{1-\Phi(x)}{\phi(x)} \\
&\geqslant \frac{1}{e^{1/6}\sqrt{2\pi}}\frac{k}{\sqrt{n}}\sqrt{\frac{n}{k(n-k)}}\frac{p^{k-1/2}(1-p)^{n-k+1/2}}{p^k(1-p)^{n-k}}\exp\Big\{-nD\Big(\frac{k}{n},p\Big)\Big\} \cdot \frac{1-\Phi(x)}{\phi(x)} \\
&= \frac{1}{e^{1/6}\sqrt{2\pi}}\sqrt{\frac{k}{n-k}} \cdot \sqrt{\frac{1-p}{p}}\exp\Big\{-nD\Big(\frac{k}{n},p\Big)\Big\} \cdot \frac{1-\Phi(x)}{\phi(x)}.
\end{aligned}$$

The result follows from $\sqrt{\frac{k}{n-k}} \geqslant \sqrt{\frac{np}{n-np}} = \sqrt{\frac{p}{1-p}}$ for $np \leqslant k \leqslant n-1$. □

For $k = pn + xn$, the 2nd-order Taylor approximation of $u(x) = D\big(\frac{k}{n},p\big) = D\big(p+x,p\big)$ around 0 is $\frac{x^2}{2p(1-p)}$. We define $\psi : \big[-p, 1-p\big] \to \mathbb{R}$ as the ratio of the divergence and the approximation:

$$\psi(x) = D(p+x,p) \cdot \frac{2p(1-p)}{x^2}.$$

In particular, we have that $1 \leqslant \psi(x) \leqslant \frac{p(1-p)}{(x+p)(1-p-x)}$ for $x \in [0, 1-p]$. This can be shown using Taylor's theorem on $u(x)$: for some $z \in [0,x]$,

$$\begin{aligned}
D(p+x,p) = \frac{u^{(2)}(z)}{2}x^2 = \Big(\frac{1}{z+p} + \frac{1}{1-p-z}\Big)\frac{x^2}{2} = \frac{x^2}{2(z+p)(1-p-z)} \\
\implies \frac{x^2}{2p(1-p)} \leqslant D(p+x,x) \leqslant \frac{x^2}{2(x+p)(1-p-x)},
\end{aligned} \tag{6.9}$$

since $\frac{1}{(x+p)(1-p-x)}$ is increasing on $[0, 1-p)$.

Let $t \in [1, np+1]$ be a real number. By Lemma 6.7.3 and Lemma 1 in Orabona and Pal (2015) (also Mill's ratio for standard Gaussian Boyd (1959)), we have

$$\begin{aligned}
\mathbb{P}\big(B_n \geqslant pn + t - 1\big) &= \mathbb{P}\big(B_n \geqslant \lceil pn + t - 1\rceil\big) \\
&\geqslant \frac{\exp\Big(-nD\big(\frac{\lceil pn+t-1\rceil}{n},p\big)\Big)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{\pi}{\pi\frac{\lceil pn+t-1\rceil - np}{\sqrt{p(1-p)n}} + \sqrt{2\pi}} \\
&\geqslant \frac{\exp\Big(-nD\big(\frac{pn+t}{n},p\big)\Big)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{\pi}{\pi\frac{pn+t-np}{\sqrt{p(1-p)n}} + \sqrt{2\pi}} \\
&= \frac{\exp\Big(-nD\big(p+\frac{t}{n},p\big)\Big)}{e^{1/6}\sqrt{2\pi}} \cdot \frac{\pi}{\pi\frac{t}{\sqrt{p(1-p)n}} + \sqrt{2\pi}} \\
&= e^{-1/6}\exp\Big(-\frac{1}{2p(1-p)}\psi\big(\frac{t}{n}\big) \cdot \frac{t^2}{n}\Big) \cdot \frac{1}{\frac{\sqrt{2\pi}t}{\sqrt{np(1-p)}} + 2}.
\end{aligned} \tag{6.10}$$

2nd Part of the Proof:

We can now turn to the actual proof of the result. Define the event A equal to the case that at least one of the $Z_i^{(n)}$ is greater or equal to $C\sqrt{np(1-p)\log d} - 1$. We will show this event / threshold controls the expectation of the maximum. First, we define C and provide some upper and lower bounds for it. Denote by $f(d) = \sqrt{2 - \frac{2\log\log d}{\log d}}$, then

$$C = C(d,n) = \frac{1}{\sqrt{\psi\Big(\sqrt{\frac{2p(1-p)\log d}{n}}\Big)}}\sqrt{2 - \frac{2\log\log d}{\log d}} = \frac{1}{\sqrt{\psi\Big(\sqrt{\frac{2p(1-p)\log d}{n}}\Big)}} f(d). \tag{6.11}$$

We bound the two factors separately:

- $z = \sqrt{\frac{2p(1-p)\log d}{n}} \in [0, \frac{1}{10}(1-p)]$ for $n \geqslant 200\frac{p}{1-p}\log d$ and so

$$1 \leqslant \psi(z) \leqslant \frac{p(1-p)}{(z+p)(1-p-z)} \leqslant \frac{1-p}{(1-p-\frac{1-p}{10})} = \frac{10}{9}. \tag{6.12}$$

- The function $f(d)$ is as in Orabona and Pal (2015): decreasing on $(1, e^e]$, increasing on $[e^e, +\infty)$, and $\lim_{d\to\infty} f(d) = \sqrt{2}$. Therefore for all $d \in [5, \infty)$,

$$1.12 \leqslant f(e^e) \leqslant f(d) \leqslant \max\big\{f(5), \sqrt{2}\big\} = \sqrt{2}$$

This gives for $n \geqslant 200\frac{p}{1-p}\log d$,

$$1 \leqslant \frac{1.12}{\sqrt{10/9}} \leqslant C(d,n) \leqslant \sqrt{2} \tag{6.13}$$

Since $p \geqslant 1/2$, if $n \geqslant 200\frac{p}{1-p}\log d$, then $n > \frac{200}{p(1-p)\log d}$ (if $d \geqslant 8$) and $n \geqslant 200\frac{1-p}{p}\log d$. The above implies:

$$1 < C\sqrt{np(1-p)\log d} \leqslant np \leqslant np + 1. \tag{6.14}$$

Finally, we bound the quantity of interest:

$$\begin{aligned}
\mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n\Big] &= \mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n | A\Big]\cdot\mathbb{P}(A) + \mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n | A^C\Big]\cdot\big(1-\mathbb{P}(A)\big) \\
&\geqslant \mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n | A\Big]\cdot\mathbb{P}(A) + \mathbb{E}\Big[Z_1^{(n)} | A^C\Big]\cdot\big(1-\mathbb{P}(A)\big) \\
&= \mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n | A\Big]\cdot\mathbb{P}(A) + \mathbb{E}\Big[Z_1^{(n)} | Z_1^{(n)} < C\sqrt{np(1-p)\log d} - 1\Big]\cdot\big(1-\mathbb{P}(A)\big)
\end{aligned}$$

$$\geqslant \mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n | A\Big]\cdot\mathbb{P}(A) + \mathbb{E}\Big[Z_1^{(n)} | Z_1^{(n)}\leqslant 0\Big]\cdot\big(1-\mathbb{P}(A)\big) \quad \text{by (6.14)}$$

$$\geqslant \big(C\sqrt{np(1-p)\log d}-1\big)\cdot\mathbb{P}(A) + \mathbb{E}\Big[Z_1^{(n)} | Z_1^{(n)}\leqslant 0\Big]\cdot\big(1-\mathbb{P}(A)\big). \tag{6.15}$$

First, we lower bound $\mathbb{E}\Big[Z_1^{(n)} | Z_1^{(n)}\leqslant 0\Big]$. Let $\beta = \frac{1}{1-\sqrt{\frac{2n}{\pi\lfloor n(1-p)\rfloor(n-\lfloor n(1-p)\rfloor)}}}$. For $n\geqslant \frac{200}{p(1-p)}\log d$, we have $n\geqslant \frac{205}{\pi p(1-p)} \geqslant \frac{200+\pi p}{\pi(1-p)p}$ and $\beta\leqslant\frac{10}{9}$.

Then Lemma 2.2 in Pelekis and Ramon (2016) combined with Lemma 8 in Doerr (2018) give that for $Y_n\sim B(n,1-p)$:

$$\mathbb{E}\big[Y_n | Y_n\geqslant n(1-p)\big] < n(1-p)+\beta\sqrt{np(1-p)} < n(1-p)+\frac{10}{9}\sqrt{np(1-p)}.$$

Since $B_n = Z^{(n)}+pn\sim B(n,p)$ can be written as $n-Y_n$, we have:

$$\begin{aligned}
\mathbb{E}\Big[Z_1^{(n)} | Z_1^{(n)}\leqslant 0\Big] &= \mathbb{E}\Big[B_n | B_n\leqslant np\Big] - np \\
&= \mathbb{E}\Big[n-Y_n | n-Y_n\leqslant np\Big] - np \\
&= n - \mathbb{E}\Big[Y_n | Y_n\geqslant n(1-p)\Big] - np \\
&\geqslant n - n(1-p) - \frac{10}{9}\sqrt{np(1-p)} - np \\
&= -\frac{10}{9}\sqrt{np(1-p)}.
\end{aligned} \tag{6.16}$$

Next, we lower-bound $\mathbb{P}(A)$:

$$\begin{aligned}
\mathbb{P}(A) &= 1-\mathbb{P}(A^C) \\
&= 1-\Big(\mathbb{P}\Big[Z_1^{(n)} < C\sqrt{np(1-p)\log d}-1\Big]\Big)^d \\
&= 1-\Big(\mathbb{P}\Big[B_n < np+C\sqrt{np(1-p)\log d}-1\Big]\Big)^d \\
&= 1-\Big(1-\mathbb{P}\Big[B_n \geqslant np+C\sqrt{np(1-p)\log d}-1\Big]\Big)^d \\
&\geqslant 1-\exp\Big(-d\cdot\mathbb{P}\Big[B_n\geqslant np+C\sqrt{np(1-p)\log d}-1\Big]\Big) \quad \text{since } 1-x\leqslant e^{-x} \\
&\geqslant 1-\exp\left(-d\cdot\frac{e^{-1/6}\exp\Big(-\frac{1}{2p(1-p)}\psi\big(\frac{C\sqrt{np(1-p)\log d}}{n}\big)\cdot\frac{C^2np(1-p)\log d}{n}\Big)}{\frac{\sqrt{2\pi}C\sqrt{np(1-p)\log d}}{\sqrt{np(1-p)}}+2}\right) \quad \text{using (6.10) and (6.14)} \\
&= 1-\exp\left(-d\cdot\frac{e^{-1/6}\exp\Big(-\frac{1}{2}\psi\big(C\sqrt{p(1-p)\log d/n}\big)\cdot C^2\log d\Big)}{\sqrt{2\pi}C\sqrt{\log d}+2}\right) \\
&= 1-\exp\left(-\frac{e^{-1/6}d^{1-\frac{C^2}{2}\psi\left(C\sqrt{p(1-p)\log d/n}\right)}}{\sqrt{2\pi}C\sqrt{\log d}+2}\right)
\end{aligned}$$

$$\geqslant 1 - \exp\left(-\frac{e^{-1/6} d^{1-\frac{C^2}{2}\psi\left(\sqrt{\frac{2p(1-p)\log d}{n}}\right)}}{2\sqrt{\pi \log d} + 2}\right) \text{ by (6.13) .}$$

We now use that $d^{1-\frac{C^2}{2}\psi\left(\sqrt{2p(1-p)\log d/n}\right)} = \log d$ by the definition of C in (6.11). Hence, we obtain:

$$\mathbb{P}(A) \geqslant 1 - \exp\left(-\frac{e^{-1/6}\log d}{2\sqrt{\pi \log d} + 2}\right) = 1 - g(d), \quad \text{for } g(d) = \exp\left(-\frac{e^{-1/6}\log d}{2\sqrt{\pi \log d} + 2}\right). \tag{6.17}$$

Putting everything together: we plug (6.16) and (6.17) into (6.15) to get

$$\begin{aligned}
\mathbb{E}\Big[\max_{1\leqslant i\leqslant d} Z_i^n\Big] &\geqslant \big(C\sqrt{np(1-p)\log d} - 1\big)\cdot\big(1 - g(d)\big) - \frac{10}{9} g(d)\sqrt{np(1-p)} \\
&= \frac{f(d)\cdot(1-g(d))}{\sqrt{\psi\Big(\sqrt{\frac{2p(1-p)\log d}{n}}\Big)}}\big(\sqrt{np(1-p)\log d} - 2\big) - \frac{10}{9} g(d)\sqrt{np(1-p)} \quad \text{using (6.11)} \\
&\geqslant \frac{f(d)\big(1-g(d)\big)}{\sqrt{10/9}}\big(\sqrt{np(1-p)\log d} - 2\big) - \frac{10}{9} g(d)\sqrt{np(1-p)} \quad \text{using (6.12)} \\
&= \sqrt{np(1-p)\log d}\cdot\left(\frac{f(d)\big(1-g(d)\big)}{\sqrt{10/9}} - \frac{10}{9}\cdot\frac{g(d)}{\sqrt{\log(d)}}\right) - 2\frac{f(d)\big(1-g(d)\big)}{\sqrt{10/9}} \\
&\geqslant 0.02\sqrt{np(1-p)\log d} - 1.5,
\end{aligned}$$

for $d \geqslant 100$ (we also used that $\sqrt{np(1-p)\log d} > 2$ in the 2nd inequality). This gives the result. □

Part III

Gradient-Based Methods in Reinforcement Learning

Chapter 7

Does Stochastic Gradient Really Succeed for Bandits?

We study the regret properties of the *Stochastic Gradient Bandit* (SGB) algorithm (Sutton and Barto, 2018, Chapter 2.8) for the multi-armed bandit problem (see Chapter 3). SGB is an instance of the softmax policy gradient algorithm in the bandit setting and can be viewed as a canonical building block of policy gradient methods, based on optimising a parameterised policy with respect to an objective (see Chapter 3; (Sutton et al., 1999; Konda and Tsitsiklis, 1999; Kakade, 2001)). Understanding its behaviour in the bandit setting provides insight into the strengths and limitations of gradient-based reinforcement learning algorithms more broadly and the results we provide can serve as a stepping stone towards understanding policy gradient methods in more general setting.

Even in the simple single-state bandit setting, the theoretical guarantees of policy gradient methods in terms of regret remain weak. In this stochastic multi-armed bandit setting, it is known that regret scaling with $\log T$ is achievable using classical algorithms such as UCB (Auer et al., 2002) or Thompson Sampling (Agrawal and Goyal, 2012). Unlike these classical bandit algorithms, SGB does not rely on explicit exploration mechanisms or confidence bounds. Instead, its behaviour is governed by a single learning-rate parameter η, whose role is crucial in balancing exploration and exploitation. Recently, Mei et al. (2023) established a regret upper bound of $\mathcal{O}(\log(T))$ for SGB with a *small* constant learning rate η satisfying $\eta \lesssim \Delta^2/K^{3/2}$ where K is the number of arms and Δ is the smallest sub-optimality gap between the expected reward of the

Setting	Performance Criteria	Main Theorem(s)	Publication Source
Two-Armed Bandit	Regret Minimisation	Theorem 7.3.3	Baudry et al. (2025)

Table 7.1: Chapter overview.

arms. In a follow-up work, Mei et al. (2024) further proved that SGB asymptotically converges to a globally optimal policy for *any* constant learning rate. While asymptotic convergence is a desirable property, it does not provide meaningful regret guarantees. This motivates further investigation of the regret properties of SGB without the small rate constraint. For the $K = 2$ armed problem, Baudry et al. (2025) show that logarithmic regret is achievable for a larger range of step-size $\eta \lesssim \Delta$. What remains is to understand the regret of SGB when $\eta > \Delta$, which is the focus of this chapter.

Contributions. We prove that the regret of SGB with $K = 2$ can be **polynomial** on some instances if $\eta \gtrsim \Delta$. Hence, we characterize the regret of SGB in two-armed bandits, except for a critical regime $\eta \approx \Delta$. These results establish Δ as the critical threshold for η between logarithmic and polynomial results and imply that knowledge of the gap Δ is both necessary and sufficient to tune the learning rate so as to guarantee logarithmic regret and optimal performance.

Roadmap. In Section 7.1, we define the problem setting. In Section 7.2, we describe the SGB algorithm along with some of its basic properties. In Section 7.3, we present and discuss the main result of this chapter on the polynomial regret of SGB when the step-size is above a certain threshold. In Section 7.4, we give an overview of the proof of this main result. In Section 7.5, we conclude and discuss future-work.

Bibliographic remark: This chapter is based on Section 2.2 of Baudry et al. (2025). The work was done in collaboration with Dorian Baudry; Dorian Baudry is the first author and my contributions involve the lower bounds, in particular Theorem 2 in Section 2.2. It is licensed under a Creative Commons Attribution 4.0 International License.

7.1 Problem Setting

A K-armed bandit is a sequential decision-making problem where, at each time $t \in [T]$, a learner chooses an action $A_t \in [K]$ and receives a reward r_t drawn independently at random from the fixed distribution ν_{A_t}. The objective is to select a policy π, which at each step t maps past

observations $\mathcal{H}_{t-1} = (A_s, r_s)_{s\in[t-1]}$ to a *sampling probability* $p^\pi_{k,t} = \mathbb{P}_\pi(A_t = k \mid \mathcal{H}_{t-1})$ for each arm $k \in [K]$, in order to maximize the expected cumulative reward $\mathbb{E}_\pi[\sum_{t=1}^T r_t]$ when actions are chosen according to π. This is equivalent to minimizing the *regret*, defined by

$$R_T^\pi := \mathbb{E}_\pi\left[\sum_{t=1}^T \sum_{k=1}^K \Delta_k \mathbb{I}(A_t = k)\right], \quad \text{with } \forall k,\ \Delta_k = \max_{j\in[K]} \mu_j - \mu_k \text{ and } \mu_k = \mathbb{E}_{r\sim\nu_k}[r]\,, \tag{7.1}$$

$$\text{or equivalently} \quad R_T^\pi = \mathbb{E}_\pi\left[\sum_{t=1}^T \sum_{k=1}^K p^\pi_{k,t}\Delta_k\right], \tag{7.2}$$

where for convenience we assume that arm 1 is the unique optimal arm, i.e., $\mu_1 > \max_{j\in\{2,\dots,K\}} \mu_j$. We further define the *minimum gap* $\Delta := \min_{k:\Delta_k>0} \Delta_k$. Finally, we denote by $\mathcal{F}$ the family of reward distributions supported on $[-1, 1]$, and assume that $\nu_k \in \mathcal{F}$ for all $k \in [K]$.

Notation. We use the shorthand notation $\mathbb{E}_t[\cdot]$ for the conditional expectation $\mathbb{E}[\cdot|\mathcal{H}_{t-1}]$.

7.2 The Stochastic Gradient Bandit policy (SGB)

Following Sutton and Barto (2018, Section 2.8), we define `SGB` as a randomized policy, with sampling probabilities at time $t \geqslant 1$ given by the softmax transform of some parameters $(\theta_{k,t})_{k\in[K]} \in \mathbb{R}^K$,

$$\forall t \in [T],\ \forall k \in [K]: \quad p^{SGB}_{k,t} = \frac{\exp(\theta_{k,t})}{\sum_{k'\in[K]} \exp(\theta_{k',t})}\,. \tag{7.3}$$

In the following, we drop the superscript for simplicity of notation. Given a fixed learning rate η, the parameter of each arm $k \in [K]$ is initialized at $\theta_{k,1} = 0$, and is updated as follows,

$$\forall t \geqslant 1:\ \theta_{k,t+1} = \theta_{k,t} + \eta \cdot \delta\theta_{k,t},\ \text{with}\ \ \delta\theta_{k,t} := (1 - p_{k,t}) \cdot r_t \cdot \mathbb{I}(A_t = k) - p_{k,t} \cdot r_t \cdot \mathbb{I}(A_t \neq k)\,. \tag{7.4}$$

SGB as Stochastic Gradient Ascent. From Equation (7.4), we derive the following fundamental property of SGB

$$\forall k \in [K], t \geqslant 1:\ \mathbb{E}_t[\delta\theta_{k,t}] = p_{k,t}(1 - p_{k,t})\mu_k - p_{k,t}\sum_{j\neq k} p_{j,t}\mu_j = p_{k,t} \cdot (\mathbb{E}_t[\Delta_{A_t}] - \Delta_k)\,. \tag{7.5}$$

This equation shows that a parameter $\theta_{k,t}$ is expected to increase only if the gap of arm k is smaller than $\mathbb{E}_t[\Delta_{A_t}] = \sum_{j=1}^K p_{j,t}\Delta_j$, the instantaneous regret of SGB. It also allows to verify that SGB indeed performs a stochastic gradient ascent in order to maximize the *value* of the

policy at time t, $V_t := \sum_{k=1}^K p_{k,t} \cdot \mu_k$, since $(\delta\theta_{k,t})_{k\in[K]}$ is an unbiased estimate of $\nabla_\theta V_t$:

$$\begin{aligned}
\forall k \in [K], t \geqslant 1: \quad \frac{\partial V_t}{\partial \theta_{k,t}} &= \sum_{j=1}^K \frac{\partial p_{j,t}}{\partial \theta_{k,t}} \mu_j \\
&= \frac{\partial p_{k,t}}{\partial \theta_{k,t}} \mu_k + \sum_{j=1, j\neq k}^K \frac{\partial p_{j,t}}{\partial \theta_{k,t}} \mu_j \\
&= \frac{e^{\theta_{k,t}} \sum_{k'\in[K]} e^{\theta_{k',t}} - e^{2\theta_{k,t}}}{\left(\sum_{k'\in[K]} e^{\theta_{k',t}}\right)^2} \mu_k - \sum_{j=1, j\neq k}^K \frac{e^{\theta_{k,t}+\theta_{j,t}}}{\left(\sum_{k'\in[K]} e^{\theta_{k',t}}\right)^2} \mu_j \\
&= p_{k,t}(1 - p_{k,t})\mu_k - p_{k,t} \sum_{j\neq k} p_{j,t}\mu_j \\
&= p_{k,t} \cdot (\mathbb{E}_t[\Delta_{A_t}] - \Delta_k) \\
&= \mathbb{E}_t[\delta\theta_{k,t}].
\end{aligned}$$

SGB as Softmax Policy Gradient: Section 3.2.2 of Chapter 3 presented policy gradient methods for reinforcement learning. The multi-armed bandit is a special case of a tabular MDP with a single state (or equivalently with discount factor $\gamma = 0$). SGB corresponds to an inexact instance of the policy gradient algorithm with softmax parameterisation. Indeed, the policy gradient for the softmax parameterisation in (3.7) corresponds to (7.4) when simplified to the multi-armed bandit setting. It is inexact since it uses an unbiased estimate of the policy gradient.

Some Additional Properties. From (7.4), we can verify that

$$\sum_{k=1}^K \delta\theta_{k,t} = r_t \cdot \left((1 - p_{A_t}) - \sum_{j\neq A_t} p_{j,t} \right) = 0,$$

and as a consequence we obtain the following relation

$$\forall t \geqslant 1, \ \sum_{k=1}^K \delta\theta_{k,t} = 0 \quad \Longrightarrow \quad \forall t \geqslant 1, \ \sum_{k=1}^K \theta_{k,t} = 0 \ .$$

This equation shows that the growth of the parameter of the best arm, $\theta_{1,t}$, is bounded by how negative the parameters of the sub-optimal arms are. Furthermore, for $K = 2$ it holds that $\theta_{1,t} = -\theta_{2,t}$, which simplifies the expression of the sampling probability. For instance, for the sub-optimal arm 2 it holds that

$$\text{If } K = 2: \ \forall t \geqslant 1, \quad p_{2,t} = \frac{e^{-\theta_{1,t}}}{e^{-\theta_{1,t}} + e^{\theta_{1,t}}} = \frac{e^{-2\theta_{1,t}}}{1 + e^{-2\theta_{1,t}}} \leqslant e^{-2\theta_{1,t}} \ .$$

7.3 The Step-Size Boundary of Logarithmic Regret

In this section, we characterise the regret of SGB with a fixed learning rate η in the two-armed setting as a function of η.

Regret minimisation in stochastic multi-armed bandits is classically solved by algorithms such as UCB (Auer et al., 2002) and Thompson Sampling (Agrawal and Goyal, 2012), which achieve logarithmic regret in the time horizon T. Moreover, it is well known that the optimal instance-dependent regret scales as $\frac{1}{\Delta}\log T$ for bounded reward distributions (Lai and Robbins, 1985; Lattimore and Szepesvári, 2020).

Recently, Mei et al. (2023) established a regret upper bound of order $\mathcal{O}(\log T)$ for SGB with a *small* constant learning rate η satisfying $\eta \lesssim \Delta^2/K^{3/2}$. More precisely, they showed that if

$$\eta \leqslant \frac{\Delta^2}{40K^{3/2}},$$

then for any bandit instance $\nu \in \mathcal{F}^K$,

$$R_T^{\mathrm{SGB}} \leqslant \frac{2}{\mathbb{E}[c^2]} \cdot \frac{\log T}{\eta} + 1, \qquad \text{where } c \coloneqq \inf_{t \geqslant 1} p_{1,t}.$$

This upper-bound on the regret scales at least as $\Omega\big(\frac{1}{\Delta^2}\log T\big)$. In the two-armed case, this dependence on Δ is sub-optimal and stems from the fact that the admissible learning rate scales as Δ^2, rather than Δ.

For the two-armed bandit problem ($K = 2$), Baudry et al. (2025) showed that logarithmic regret is achievable for a substantially larger range of learning rates, namely $\eta \lesssim \Delta$, which allows one to recover the optimal instance-dependent regret scaling. We restate their result below.

Theorem 7.3.1 (Regret upper bound for $K = 2$, Theorem 1 of Baudry et al. (2025))**.** *For a two-armed bandit instance $\nu \in \mathcal{F}^2$ with gap $\Delta \in (0, 1)$, the regret of SGB tuned with a learning rate η satisfying $\eta C_\eta < \Delta$ is upper bounded by*

$$R_T^{\mathrm{SGB}} \leqslant \frac{\log(1 + 4\eta\Delta T)}{2\eta} + \frac{\Delta}{2\eta(\Delta - \eta C_\eta)},$$

where

$$C_\eta := 2\sum_{n=0}^{+\infty} \frac{(2\eta)^n}{(n+2)!} \leqslant e^{2\eta}.$$

In particular, by choosing $\eta \approx \Delta$, this bound yields regret of order $\frac{1}{\Delta}\log T$, which is optimal up to constant factors.

Beyond regret guarantees, Mei et al. (2024) further showed that SGB asymptotically converges to a globally optimal policy for *any* constant learning rate.

Theorem 7.3.2 (Asymptotic convergence, Theorem 2 of Mei et al. (2024))**.** *For SGB with any constant learning rate η, we have, almost surely,*

$$p_{1,t}^{\mathrm{SGB}} \to 1 \qquad \text{as } t \to \infty,$$

where arm 1 denotes the optimal arm.

While asymptotic convergence is a desirable property, for a fixed finite horizon T it does not by itself provide meaningful regret guarantees. In particular, it is straightforward to construct examples in which SGB suffers linear regret when the learning rate is chosen too large (e.g., $\eta \approx T$), while guaranteed to converge asymptotically by the above result.

For the 2-armed bandit, the results above characterise the regret of SGB when the learning rate satisfies $\eta \lesssim \Delta$ while for $\eta > \Delta$ they guarantee asymptotic convergence, leaving the behaviour of the regret is unknown. The main contribution of this chapter is to characterise the behaviour of SGB in this regime and to show that Δ precisely marks the step-size threshold beyond which logarithmic regret is no longer achievable.

We prove that the knowledge of the minimum gap Δ is necessary to tune the fixed learning rate of SGB in order to obtain logarithmic regret. The following Theorem 7.3.3 is the main result of this chapter and shows that if η is larger than a fixed constant that depends on the gap Δ and the number of arms K, there exists a K-armed bandit problem with gap Δ for which SGB has polynomial regret. To define it, for $\mu \in [-1, 1]$ we denote respectively by $\mathrm{Rad}(\mu)$ and δ_μ the Rademacher and Dirac distributions of mean μ.

Theorem 7.3.3. *Fix $\Delta \in (0, 1)$, and consider the instance $\nu = (\nu_k)_{k\in[K]}$ with $\nu_1 = \mathrm{Rad}(\Delta)$ and*

$\nu_2 = \cdots = \nu_K = \delta_0$. *If the learning rate of SGB satisfies*

$$\eta > \lambda_\Delta := \tfrac{K-1}{K} \log\Big(1 + \tfrac{2\Delta}{1-\Delta}\Big),$$

then its regret on the instance ν *is lower bounded as follows,*

$$\forall \epsilon > 0, \qquad R_T = \widetilde{\Omega}\Big(T^{1-(1+\epsilon)\lambda_\Delta/\eta}\Big),$$

where $\widetilde{\Omega}$ *hides polylogarithmic terms in* T *and constants depending on* η *and* Δ.

We give a general version of the result but for $K = 2$, the threshold becomes

$$\eta > \frac{1}{2} \log\Big(1 + \tfrac{2\Delta}{1-\Delta}\Big) \to \Delta, \qquad \text{as } \Delta \to 0.$$

We remark that Theorems 7.3.1 and 7.3.3 do not provide explicit results about the regret of SGB when $\eta \approx \Delta$. While Baudry et al. (2025) conjecture that their could be a smooth interpolation between the logarithmic and polynomial regime near around this threshold, we note the η-gap between these results goes to 0 as $\Delta \to 0$. Hence, we fully characterize the regret of SGB in two-armed bandits. In particular, *knowing a lower bound on the gap* Δ *is necessary and sufficient to tune* η *to guarantee logarithmic regret.*

7.4 Proof of Theorem 7.3.3

Roadmap Below, we present the proof of Theorem 7.3.3, which uses several technical lemmas. These results, Lemma 7.4.1 and Lemma 7.4.2, are presented in Section 7.4.1. Furthermore, the proof of Lemma 7.4.2 is quite technical and itself supported by several more technical intermediate results, that we detail in Section 7.4.2.

Since only arm 1 yields non-zero rewards, we directly have

$$\forall t \geqslant 1,\ \theta_{1,t+1} = \theta_{1,t} + \eta(1 - p_{1,t}) \cdot r_t \cdot \mathbb{I}(A_t = 1)\ ,$$

so we denote by $\widetilde{p}_{n+1}$ the sampling probability of arm 1 after its n first pulls. The proof consists of identifying a scenario with linear regret, and lower bounding its probability as a function of η. Step 1: let $\mathcal{S}$ be the event that $\widetilde{p}_{n+1} \leqslant \frac{1}{2T}$ for some value of n that we will fix later. Then,

under $\mathcal{S}$ the probability that arm 1 is never pulled again after its n first selections is larger than $\left(1-\frac{1}{2T}\right)^T \geqslant \frac{1}{2}$, in which case the regret is larger than $T-n$. We hence first obtain that

$$R_T(\nu) \geqslant \Delta \cdot \mathbb{P}(\mathcal{S}) \cdot \frac{(T-n)_+}{2} \,.$$

Step 2: We lower bound $\mathbb{P}(\mathcal{S})$ for a well-chosen value of n, by identifying a scenario that leads (deterministically) to $\widetilde{p}_{n+1} \leqslant \frac{1}{2T}$. Let n_0, n_1 be integers to be specified later, we consider the following repartition of n pulls in two phases,

- **Phase 0 - very unlucky start:** for its first n_0 pulls, arm 1 collects only -1 rewards. We denote this event by $\mathcal{S}_0$.
- **Phase 1 - failed recovery:** the following n_1 pulls of arm 1 have an empirical mean satisfying $\widehat{\mu}_{n_1} \leqslant -\alpha\Delta$ (where α is defined below in (7.6)) and the number of $+1$s received is never more than the number of -1s throughout this phase. We denote this event by $\mathcal{S}_1$.

Since ν_1 is Rademacher of mean Δ, it is direct that $\mathbb{P}(\mathcal{S}_0) = \left[\frac{1-\Delta}{2}\right]^{n_0}$. For phase 1, let $(\widetilde{R}_t)_{t\geqslant 1}$ denote the stream of rewards collected by arm 1 (in the order they are received), and $\widehat{\mu}_{n_1} = \frac{1}{n_1}\sum_{t=n_0+1}^{n_0+n_1} \widetilde{R}_t$ be the empirical mean of the rewards during phase 1. Let $S_{n_1} = \sum_{t=n_0+1}^{n_0+n_1} \mathbb{I}\{\widetilde{R}_t = 1\}$ be the number of ones received during this phase. We first define

$$\alpha \coloneqq \frac{2}{\Delta}\left(\frac{1}{2} - \frac{\left[\frac{1-\Delta}{2}\cdot n_1\right]}{n_1}\right) \in \left[1-\frac{2}{\Delta n_1}, 1\right], \tag{7.6}$$

so that $\frac{1-\alpha\Delta}{2}$ is an integer greater than $\frac{1-\Delta}{2}\cdot n_1$ with α that is as close to 1 as possible.
From Lemma 7.4.1, we first obtain that

$$\mathbb{P}\left(\widehat{\mu}_{n_1} = -\alpha\Delta\right) = \mathbb{P}\left(S_{n_1} = \frac{1-\alpha\Delta}{2}\cdot n_1\right) \geqslant e^{-1/6}\sqrt{\frac{2}{n_1\pi}}\exp\left(-n_1\cdot\Delta\log\left(1+\frac{2\Delta}{1-\Delta}\right)\right).$$

Then, conditioned on this event, Bertrand's Ballot theorem (Bertrand, 1887; Renault, 2008) provides that the running number of -1 always exceeds the running number of $+1$ with probability

$$\frac{\left(n_1 - S_{n_1}\right) - S_{n_1}}{n_1} = \frac{n_1 - 2S_{n_1}}{n_1} = -\widehat{\mu}_{n_1} = \alpha\Delta,$$

which finally gives that the probability of $\mathcal{S}_1$ is lower bounded by

$$\mathbb{P}(\mathcal{S}_1) \geqslant \alpha\Delta\cdot e^{-1/6}\sqrt{\frac{1}{2n_1\pi}}\exp\left(-n_1\cdot\Delta\log\left(1+\frac{2\Delta}{1-\Delta}\right)\right).$$

Then, it remains to show that $\mathcal{S}_0 \cap \mathcal{S}_1 \subset \mathcal{S}$ for some values of n_0 and n_1. By Lemma 7.4.2, this is guaranteed for $n = n_0 + n_1$ by fixing

$$n_0 = \left\lceil \frac{1}{\eta} \cdot \frac{K}{K-1} \cdot \log\left(\frac{1}{\epsilon \cdot (K-1)}\right) \right\rceil \vee \left\lceil \frac{1}{\eta}\Big(\frac{K}{K-1}\Big)^2 \log\Big(\frac{K^3 n_1}{(K-1)^2}\Big) \right\rceil,$$
$$\text{and} \quad n_1 = \left\lceil \frac{K-1}{K} \cdot \frac{\log(2T/(K-1))}{\eta\Delta(1-\epsilon)} + \frac{2}{\Delta} \right\rceil.$$

By independence of the two sequences of rewards involved, it holds that $\mathbb{P}(\mathcal{S}) \geqslant \mathbb{P}(\mathcal{S}_0) \cdot \mathbb{P}(\mathcal{S}_1)$. To obtain the scaling of the result, we combine the above results and use that n_0 is tuned as a function of n_1. Hence, we can state that there exists a function C_{n_1}, with (inverse) polynomial scaling in n_1, such that

$$\mathbb{P}(\mathcal{S}) \geqslant C_{n_1} \cdot e^{-n_1 \cdot \Delta \log\left(1+\frac{2\Delta}{1-\Delta}\right)},$$

so that replacing n_1 by its value and ignoring the C_{n_1} term, which we hide with the $\gtrsim$ sign, leads to

$$\begin{aligned}
\mathbb{P}(\mathcal{S}) &\gtrsim e^{-n_1 \cdot \Delta \log\left(1+\frac{2\Delta}{1-\Delta}\right)} \\
&\geqslant e^{-\left(1+\frac{2}{\Delta}+\frac{K-1}{K}\frac{1+\varepsilon}{\eta\Delta}\log\frac{2T}{K-1}\right)\cdot\Delta\log\left(1+\frac{2\Delta}{1-\Delta}\right)} \\
&= e^{-(\Delta+2)\log\left(1+\frac{2\Delta}{1-\Delta}\right)} \cdot e^{-\frac{K-1}{K}\frac{1+\varepsilon}{\eta}\log\left(1+\frac{2\Delta}{1-\Delta}\right)\cdot\log\frac{2T}{K-1}} \\
&= e^{-(\Delta+2)\log\left(1+\frac{2\Delta}{1-\Delta}\right)} \cdot \left(\frac{2}{K-1}\right)^{-\frac{K-1}{K}\frac{1+\varepsilon}{\eta}\log\left(1+\frac{2\Delta}{1-\Delta}\right)} \cdot T^{-\frac{K-1}{K}\frac{1+\varepsilon}{\eta}\log\left(1+\frac{2\Delta}{1-\Delta}\right)}.
\end{aligned}$$

For completeness, we detail the poly-logarithmic terms coming from the expression of C_{n_1} below,

$$\begin{aligned}
C_{n_1} &:= \left(\frac{1-\Delta}{2}\right)^{n_0} \cdot \alpha\Delta \cdot e^{-1/6}\sqrt{\frac{2}{n_1\pi}} \\
&\geqslant \alpha\Delta \cdot e^{-1/6} \cdot \sqrt{\frac{2}{n_1\pi}} \cdot \left(\frac{1-\Delta}{2}\right) \cdot \left(\frac{K^3 n_1}{(K-1)^2}\right)^{\frac{1}{\eta}\left(\frac{K}{K-1}\right)^2 \log\left(\frac{1-\Delta}{2}\right)} \\
&= \alpha\Delta \cdot \left(\frac{1-\Delta}{2}\right) \cdot e^{-1/6} \cdot \sqrt{\frac{2}{\pi}} \cdot \left(\frac{K^3}{(K-1)^2}\right)^{\frac{1}{\eta}\left(\frac{K}{K-1}\right)^2 \log\left(\frac{1-\Delta}{2}\right)} \cdot n_1^{-\frac{1}{2}+\frac{1}{\eta}\left(\frac{K}{K-1}\right)^2 \log\left(\frac{1-\Delta}{2}\right)}.
\end{aligned}$$

We can further detail C_{n_1} by replacing n_1 by its value. For simplicity, we capture all the problem-dependent factors in a new constant $c_{\eta,\Delta}$, and finally obtain that

$$\exists c_{\eta,\Delta} > 0 : \quad C_{n_1} \geqslant c_{\eta,\Delta} \cdot (\log(T))^{-\frac{1}{2}+\frac{1}{\eta}\left(\frac{K}{K-1}\right)^2 \log\left(\frac{1-\Delta}{2}\right)}.$$

From a high-level perspective, the preliminary phase only serves to make $\widetilde{p}_{n_0+1}$ small enough to reduce the impact of the ordering of the rewards in the next phase (see the proof of Lemma 7.4.2). The scaling of the lower bound (ignoring log terms) comes from the choice of n_1.

Step 3: Combining the results from the first two steps, we obtain that

$$R_T(\nu) \geqslant \Delta \cdot \mathbb{P}(\mathcal{S}) \cdot \frac{(T - n_0 - n_1)_+}{2} \gtrsim T^{1 - \frac{K-1}{K}\frac{1+\varepsilon}{\eta}\log\left(1+\frac{2\Delta}{1-\Delta}\right)},$$

where the omitted constants depending on η and Δ and polylog factors in T can be recovered from the above lower bound on $\mathbb{P}(\mathcal{S})$. This concludes the proof.

We emphasize that the result should hold with random sub-optimal arms (further assuming that they are "lucky enough" throughout the trajectory), but constant rewards simplify the proof.

7.4.1 Technical Lemmas

Anti-concentration of the empirical mean of Rademacher variables

Lemma 7.4.1. *Let n be a sample size and S_n be the number of ones received from n independent pulls of a Rademacher variable of mean $\Delta \geqslant \frac{2}{n}$. Fix $\alpha = \frac{2}{\Delta}\left(\frac{1}{2} - \frac{1}{n} \cdot \left\lceil \frac{1-\Delta}{2} \cdot n \right\rceil\right) \in \left[1 - \frac{2}{\Delta n}, 1\right]$, then*

$$\mathbb{P}\left(S_n = \frac{1-\alpha\Delta}{2} \cdot n\right) \geqslant e^{-1/6}\sqrt{\frac{2}{n\pi}} \exp\left(-n \cdot \Delta \log\left(1 + \frac{2\Delta}{1-\Delta}\right)\right).$$

Proof. Using the notation $p \coloneqq \frac{1+\Delta}{2}$, we use the following standard formula,

$$\begin{aligned}
\mathbb{P}(S_n = s) &= \binom{n}{s} \cdot p^s (1-p)^{n-s} \\
&= \binom{n}{s} e^{s\log(p) + (n-s)\log(1-p)} \\
&= \binom{n}{s} e^{n\left(\frac{s}{n}\log\left(\frac{p}{s/n}\right) + \frac{n-s}{n}\log\left(\frac{1-p}{(n-s)/n}\right)\right) - nH\left(\frac{s}{n}\right)} \\
&= \binom{n}{s} e^{-nH\left(\frac{s}{n}\right)} \cdot e^{-n\mathrm{kl}\left(\frac{s}{n}, p\right)},
\end{aligned}$$

where $H(x) = -x\log x - (1-x)\log(1-x)$ is the Shannon entropy, and kl the Bernoulli KL-divergence. With $\frac{s}{n} = \frac{1-\alpha\Delta}{2}$ and $\alpha \leqslant 1$, we have

$$\mathrm{kl}\left(\frac{s}{n}, p\right) \leqslant \mathrm{kl}\left(\frac{1-\Delta}{2}, \frac{1+\Delta}{2}\right) = \frac{1-\Delta}{2}\log\frac{1-\Delta}{1+\Delta} + \frac{1+\Delta}{2}\log\frac{1+\Delta}{1-\Delta}$$

$$= \Delta \log\Big(\frac{1+\Delta}{1-\Delta}\Big) = \Delta \log\left(1 + \frac{2\Delta}{1-\Delta}\right).$$

Sterling's approximation for the factorial (Robbins, 1955) gives for any $n \geqslant 1$

$$\sqrt{2\pi n}\Big(\frac{n}{e}\Big)^n < n! < \sqrt{2\pi n}\Big(\frac{n}{e}\Big)^n e^{1/12}.$$

Thus, for $c \in (0,1)$ such that cn is integer, we obtain that

$$\begin{aligned}
\binom{n}{cn} &= \frac{n!}{(cn)!((1-c)n)!} \\
&\geqslant \frac{\sqrt{2\pi n}\cdot n^n e^{-n}}{\big(e^{1/12}\sqrt{2\pi cn}\cdot c^{cn}n^{cn}e^{-cn}\big)\cdot\big(e^{1/12}\sqrt{2\pi(1-c)n}\cdot(1-c)^{(1-c)n}n^{(1-c)n}e^{-(1-c)n}\big)} \\
&= \frac{e^{-1/6}}{\sqrt{2\pi c(1-c)n}}\cdot\Big(\frac{n}{e}\Big)^{n-cn-(1-c)n}\cdot\frac{1}{c^{cn}\cdot(1-c)^{(1-c)n}} \\
&= \frac{e^{-1/6}}{\sqrt{2\pi c(1-c)n}}\cdot e^{nH(c)}\,.
\end{aligned}$$

Remarking that $\sqrt{2c(1-c)} \leqslant 1/\sqrt{2}$ for any $c \in (0,1)$, we finally obtain with $cn = s$:

$$\mathbb{P}(S_n = s) \geqslant e^{-1/6}\sqrt{\frac{2}{n\pi}}\cdot e^{-n\mathrm{kl}\left(\frac{1-\alpha\Delta}{2},\frac{1+\Delta}{2}\right)} = e^{-1/6}\sqrt{\frac{2}{n\pi}}\exp\left(-n\cdot\Delta\log\left(1+\frac{2\Delta}{1-\Delta}\right)\right).$$

□

Sufficient conditions on n_0 and n_1 for $\mathcal{S}_0 \cap \mathcal{S}_1 \subset \mathcal{S}$

Lemma 7.4.2. *Fix any $\epsilon \in (0,1)$, and define n_0, n_1 as follows:*

$$\begin{aligned}
n_0 &= \left\lceil \frac{1}{\eta}\cdot\frac{K}{K-1}\cdot\log\left(\frac{1}{\epsilon\cdot(K-1)}\right)\right\rceil \vee \left\lceil \frac{1}{\eta}\Big(\frac{K}{K-1}\Big)^2\log\Big(\frac{K^3 n_1}{(K-1)^2}\Big)\right\rceil, \\
n_1 &= \left\lceil \frac{K-1}{K}\cdot\frac{\log(2T/(K-1))}{\eta\Delta(1-\epsilon)} + \frac{2}{\Delta}\right\rceil.
\end{aligned}$$

Then under $\mathcal{S}_0 \cap \mathcal{S}_1$, it holds that $\widetilde{p}_{n_0+n_1+1} \leqslant \frac{1}{2T}$.

Proof. Using the same notation as $\widetilde{p}_{n+1}$, let $\widetilde{\theta}_{n+1}$ be the the SGB parameter of arm 1 after its n first pulls.

First, at the end of phase 0 the policy has received n_0 rewards of -1 from arm 1. Therefore,

since $1 - \widetilde{p}_n \geqslant 1 - \frac{1}{K} = \frac{K-1}{K}$ for all $n \leqslant n_0$, we have that

$$\widetilde{\theta}_{n_0+1} = -\eta \sum_{n=1}^{n_0} (1 - \widetilde{p}_n) \leqslant -\frac{K-1}{K} \cdot \eta n_0. \tag{7.7}$$

Next, we show that the combination of this property and the events occurring under $\mathcal{S}_1$ are sufficient to prove the desired result. Indeed, we recall that under $\mathcal{S}_1$, the sequence of rewards in phase 1, $\{\widetilde{R}_t\}_{t=n_0+1}^{n_0+n_1}$, satisfy the following two conditions:

- The total number of $+1$ rewards should be less than $\frac{1-\alpha\Delta}{2} \cdot n_1$ and the total number of -1 rewards should be more than $\frac{1+\alpha\Delta}{2} \cdot n_1$; i.e. $\sum_{t=n_0+1}^{n_0+n_1} \widetilde{R}_t \leqslant -n_1\alpha\Delta$.
- There should never be more $+1$ rewards than -1 rewards observed at the current time step within the phase; i.e. $\sum_{t=n_0+1}^{n_0+k} \widetilde{R}_t \leqslant 0$ for all $k \leqslant n_1$.

The rest of the proof is based on characterizing the sequence among all sequences satisfying the above conditions that leads to the largest value for $\widetilde{\theta}_{n_0+n_1+1}$. We call this the *maximising* sequence. Then, we show that even for that sequence it holds that $\widetilde{p}_{n_0+n_1+1} \leqslant \frac{1}{2T}$, from which the result of the lemma follows. Section 7.4.2 is dedicated to this and we detail below the lemmas that lead us to this result culminating in Lemma 7.4.9.

In Lemma 7.4.6, we give a general construction for the maximising sequence that applies to reward sequences satisfying the above conditions for arbitrary initial parameter. Then, in Corollary 7.4.8, we show that under the occurrence of phase 0, the maximising sequence of rewards with initial parameter $\widetilde{\theta}_{n_0+1}$ takes a simple form: a first period alternating -1 and $+1$ rewards followed by a second period with only -1 rewards. From this simple form, we can upper-bound $\widetilde{\theta}_{n_0+n_1+1}$ for the maximising sequence (also Corollary 7.4.8). Then, in Lemma 7.4.9, we show that the proposed tuning for n_0 and n_1 is sufficient to get that the probability $\widetilde{p}_{n_0+n_1+1}$ is smaller than $\frac{1}{2T}$. □

From a high-level perspective, we highlight that the phase $\mathcal{S}_0$ serves to mitigate the impact of the order in which rewards are received in the SGB updates, by ensuring that $\widetilde{\theta}_{n_0+t}$ remains "sufficiently" negative throughout the entire phase 1 (i.e. for $t \leqslant n_1$), while the tuning $n_1 = \mathcal{O}(\log(T))$ permits that a probability of $\frac{1}{2T}$ is achieved even in the identified maximising scenario.

7.4.2 Maximizing Reward Sequence and Probability Under $\mathcal{S}_0 \cap \mathcal{S}_1$

In this subsection, our goal is to upper bound the probability $\widetilde{p}_{n_0+n_1+1}$ under the bad event $\mathcal{S}_0 \cap \mathcal{S}_1$. To do so, we construct a reward sequence that maximizes $\widetilde{p}_{n_0+n_1+1}$ among all sequences in $\mathcal{S}_0 \cap \mathcal{S}_1$. We start by recalling the two events:

- $\mathcal{S}_0$ **(phase 0):** during its first n_0 pulls, arm 1 receives only -1 rewards.
- $\mathcal{S}_1$ **(phase 1):** during the next n_1 pulls, arm 1 has an empirical mean $\widehat{\mu}_{n_1} \leqslant -\alpha\Delta$ (with α defined in (7.6)), and the cumulative number of $+1$ rewards never exceeds the cumulative number of -1 rewards.

Structure of the argument. The analysis proceeds in five steps:

1. We characterize the feasible reward trajectories under the event $\mathcal{S}_1$, formalizing which sequences of $+1$ and -1 rewards are allowed.
2. We analyze the local effect of reward ordering on the SGB parameter, identifying a critical threshold that determines whether placing a $+1$ or -1 first increases the parameter.
3. We construct a globally maximizing reward sequence by iteratively applying the local ordering principle while respecting feasibility, and prove its optimality.
4. We apply this worst-case sequence to phase 1, using the outcome of phase 0 to control the initial parameter and bound the maximal SGB parameter $\widetilde{\theta}^{M}_{n_1+1}$.
5. We convert the bound on the SGB parameter into an explicit bound on the probability $\widetilde{p}_{n_0+n_1+1}$, completing the control of the bad event.

Step 1: Constraining trajectories under $\mathcal{S}_1$ We first formalise which sequences are allowed under $\mathcal{S}_1$.

Definition 7.4.3 (Constrained trajectories)**.** *Let $\mathcal{Z} \subset \{-1, 1\}^{\mathbb{N}}$ be the set of reward trajectories that satisfy the constraints of $\mathcal{S}_1$. For a finite trajectory $S_n = \{R_1, \ldots, R_n\} \in \mathcal{Z}$, define the* feasibility indices*:*

$$I_+^{\mathcal{Z}}(S_n) = \mathbb{I}(\{R_1, \ldots, R_n, +1\} \in \mathcal{Z}),$$
$$I_-^{\mathcal{Z}}(S_n) = \mathbb{I}(\{R_1, \ldots, R_n, -1\} \in \mathcal{Z}).$$

These indicate whether adding a $+1$ or -1 reward to S_n is allowed under the constraints. Specifically, for $n < n_1$:

- $I_+^{\mathcal{Z}}(S_n) = 1$ *if the number of* $+1$ *rewards in* S_n *is less than* $\frac{1-\alpha\Delta}{2} \cdot n_1$ *and less than the number of* -1 *rewards.*
- $I_-^{\mathcal{Z}}(S_n) = 1$ *if the number of* -1 *rewards in* S_n *is less than* $\frac{1+\alpha\Delta}{2} \cdot n_1$.

Step 2: Understanding local effects of reward ordering Before constructing a full maximizing sequence, we need a local building block: how the order of two consecutive rewards affects the SGB parameter.

Lemma 7.4.4 (Two-step reward effect)**.** *Consider an arbitrary parameter θ_n at the n-th pull of arm 1 (all other arms set to $-\frac{1}{K-1}\theta_n$). Let $R_n, R_{n+1} \in \{-1, 1\}$ denote the next two rewards. Define:*

- θ_{n+2} *as the parameter after* $(R_n, R_{n+1}) = (1, -1)$
- $\widetilde{\theta}_{n+2}$ *as the parameter after* $(R_n, R_{n+1}) = (-1, 1)$.

Then:

- *If* $\theta_n > \frac{K-1}{K}\log(K-1)$, $\theta_{n+2} < \widetilde{\theta}_{n+2}$
- *If* $\theta_n = \frac{K-1}{K}\log(K-1)$, $\theta_{n+2} = \widetilde{\theta}_{n+2}$
- *If* $\theta_n < \frac{K-1}{K}\log(K-1)$, $\theta_{n+2} > \widetilde{\theta}_{n+2}$

Proof. The key idea is to compute explicitly how swapping the order of a $+1$ and -1 reward changes the parameter. We first explicitly compute the values of the parameters $\widetilde{\theta}_{n+2}$ and θ_{n+2} after the two steps:

$$\begin{aligned}
\widetilde{\theta}_{n+2} &= \theta_n - \eta(1-p_n) + \eta(1-\widetilde{p}_{n+1}) = \theta_n - \eta(1-p_n) + \eta\frac{K-1}{K-1+\exp(\frac{K}{K-1}\widetilde{\theta}_{n+1})} \\
&= \theta_n - \eta(1-p_n) + \eta\frac{K-1}{K-1+\exp\left(\frac{K}{K-1}\theta_n - \frac{K}{K-1}\eta(1-p_n)\right)}. \\
\theta_{n+2} &= \theta_n + \eta(1-p_n) - \eta(1-p_{n+1}) = \theta_n + \eta(1-p_n) - \eta\frac{K-1}{K-1+\exp(\frac{K}{K-1}\theta_{n+1})} \\
&= \theta_n + \eta(1-p_n) - \eta\frac{K-1}{K-1+\exp\left(\frac{K}{K-1}\theta_n + \frac{K}{K-1}\eta(1-p_n)\right)}.
\end{aligned}$$

We then compare the two to understand which is larger:

$$\theta_{n+2} - \widetilde{\theta}_{n+2} = 2\eta(1-p_n) - \eta\frac{K-1}{K-1+\exp\left(\frac{K}{K-1}\theta_n + \frac{K}{K-1}\eta(1-p_n)\right)} - \eta\frac{K-1}{K-1+\exp\left(\frac{K}{K-1}\theta_n - \frac{K}{K-1}\eta(1-p_n)\right)}.$$

Next, using $1-p_n = \frac{K-1}{K-1+\exp(\frac{K}{K-1}\theta_n)}$, define

$$f(x) = \frac{2\eta(K-1)}{K-1+\exp(x)} - \frac{\eta(K-1)}{K-1+\exp\left(x+\frac{K}{K-1}\frac{\eta(K-1)}{K-1+\exp(x)}\right)} - \frac{\eta(K-1)}{K-1+\exp\left(x-\frac{K}{K-1}\frac{\eta(K-1)}{K-1+\exp(x)}\right)},$$

which corresponds to the difference in parameters above as a function of $\frac{K}{K-1}\theta_n$.

Let $Y = \exp(\frac{K}{K-1}\frac{\eta(K-1)}{K-1+\exp(x)})$ and $Z = K-1$. Then $f(x) > 0$ if and only if

$$\begin{aligned}
&\frac{2}{Z+\exp(x)} > \frac{1}{Z+Y\exp(x)} + \frac{1}{Z+\frac{1}{Y}\exp(x)} \\
&\iff 2Z^2 + 2ZYe^x + \frac{2Z}{Y}e^x + 2e^{2x} > 2Z^2 + ZYe^x + \frac{Z}{Y}e^x + 2Ze^x + Ye^{2x} + \frac{1}{Y}e^{2x} \\
&\iff (Y-1)^2(Z-e^x) > 0 \\
&\iff x < \log Z = \log(K-1).
\end{aligned}$$

Similarly, $f(x) < 0$ if $x > \log(K-1)$ and $f(x) = 0$ if $x = \log(K-1)$. Finally, since $f\left(\frac{K}{K-1}\theta_n\right) = \theta_{n+2} - \widetilde{\theta}_{n+2}$, this proves the lemma. □

Remark 7.4.5. *The intuition is simple but powerful:*

- *Above the threshold, placing a -1 reward first increases the parameter more than placing a $+1$ first.*
- *Below the threshold, placing a $+1$ first is better.*
- *At the threshold, the order does not matter.*

We will repeatedly use this property to construct sequences that maximize the parameter over many pulls.

Step 3: Constructing the full maximizing sequence We now combine the two-step intuition with the feasibility constraints of $\mathcal{S}_1$. At each step, we choose the reward that is both feasible

and locally maximizes the parameter. Iterating this procedure produces a globally maximizing sequence.

Lemma 7.4.6 (Maximizing reward sequence)**.** *Fix $\widetilde{\theta}_1 \in \mathbb{R}^K$. Let $\{\widetilde{R}_t^M\}_{t=1}^{n_1} \in \mathcal{Z}$ be defined recursively by*

$$\widetilde{R}_t^M = \begin{cases} -1, & \text{if } \widetilde{\theta}_t^M > \frac{K-1}{K}\log(K-1) \text{ and } I_-^{\mathcal{Z}}(S_{t-1}^M) = 1, \text{ or } I_+^{\mathcal{Z}}(S_{t-1}^M) = 0, \\ +1, & \text{if } \widetilde{\theta}_t^M < \frac{K-1}{K}\log(K-1) \text{ and } I_+^{\mathcal{Z}}(S_{t-1}^M) = 1, \text{ or } I_-^{\mathcal{Z}}(S_{t-1}^M) = 0, \\ +1 \text{ or } -1, & \text{otherwise (i.e., } \widetilde{\theta}_t^M = \frac{K-1}{K}\log(K-1) \text{ and either choice is feasible).} \end{cases}$$

with $S_t^M = \{\widetilde{R}_1^M, \ldots, \widetilde{R}_t^M\}$. Then $\{\widetilde{R}_t^M\}_{t=1}^{n_1}$ maximizes the SGB parameter:

$$\widetilde{\theta}_{n_1+1}^M \geqslant \widetilde{\theta}_{n_1+1}$$

for any sequence $\{\widetilde{R}_t\} \in \mathcal{Z}$, both starting from $\widetilde{\theta}_1$.

Remark 7.4.7. *At each step, the construction applies the two-step effect (Lemma 7.4.4) while respecting feasibility. Because each local choice maximizes the parameter, the resulting sequence is globally optimal.*

Proof of Lemma 7.4.6. We prove the lemma by contradiction, showing that any sequence different from $\{\widetilde{R}_t^M\}_{t=1}^{n_1}$ cannot produce a larger final parameter.

Assume $\{\widetilde{R}_t^M\}_{t=1}^{n_1}$ is not maximizing, and let $\{\widetilde{R}_t^W\}_{t=1}^{n_1} \in \mathcal{Z}$ be a sequence that achieves the maximum, with corresponding parameters $\widetilde{\theta}_t^W$ and trajectories S_t^W. Denote the resulting maximal final parameter by $\widetilde{\theta}_{n_1+1}^W$.

Since $\{\widetilde{R}_t^W\}_{t=1}^{n_1} \neq \{\widetilde{R}_t^M\}_{t=1}^{n_1}$, there exists at least one index n where one of the conditions in the definition of $\widetilde{R}_n^M$ is violated. Specifically, for some n:

1. $\widetilde{R}_n^W = +1$, $\widetilde{\theta}_n^W > \frac{K-1}{K}\log(K-1)$, and $I_-^{\mathcal{Z}}(S_{n-1}^W) = 1$.
2. $\widetilde{R}_n^W = -1$, $\widetilde{\theta}_n^W < \frac{K-1}{K}\log(K-1)$, and $I_+^{\mathcal{Z}}(S_{n-1}^W) = 1$.
3. $\widetilde{R}_n^W = +1$ and $I_+^{\mathcal{Z}}(S_{n-1}^W) = 0$.
4. $\widetilde{R}_n^W = -1$ and $I_-^{\mathcal{Z}}(S_{n-1}^W) = 0$.

The last two cases cannot occur because $\{\widetilde{R}_t^W\}_{t=1}^{n_1} \in \mathcal{Z}$ is feasible by definition.

Let m be the largest index at which one of the first two violations occurs. By construction, at this point the "opposite" reward is still feasible according to the feasibility indices, so there must be at least one more reward after m; in particular, $m < n_1$. We now focus on the two cases separately, with the goal of showing that in each case we can swap the pair of rewards at positions m and $m+1$ to strictly increase the final SGB parameter, thereby contradicting the assumption that the original sequence was maximizing.

Case 1. Suppose

$$\widetilde{R}_m^W = +1, \qquad \widetilde{\theta}_m^W > \frac{K-1}{K}\log(K-1), \qquad \text{and} \qquad I_-^{\mathcal{Z}}(S_{m-1}^W) = 1.$$

First, we claim that $\widetilde{R}_{m+1}^W$ must be -1. To see why, we proceed by contradiction. Suppose instead that $\widetilde{R}_{m+1}^W = +1$. Then, because $I_-^{\mathcal{Z}}(S_m^W) = 1$ and $\widetilde{\theta}_{m+1}^W \geqslant \widetilde{\theta}_m^W > \frac{K-1}{K}\log(K-1)$, the index $m+1$ would satisfy the same violation as m. This contradicts the definition of m as the largest index at which the violation occurs, so our claim follows: $\widetilde{R}_{m+1}^W$ must indeed be -1.

Recall that in *Case 1* we are in the situation where

$$\widetilde{R}_m^W = +1, \qquad \widetilde{\theta}_m^W > \frac{K-1}{K}\log(K-1), \qquad \text{and} \qquad I_-^{\mathcal{Z}}(S_{m-1}^W) = 1.$$

We have just shown that this necessarily implies $\widetilde{R}_{m+1}^W = -1$, so that the reward pair at times $(m, m+1)$ is $(+1, -1)$ while the parameter is above the critical threshold from Lemma 7.4.4 at time m. When the parameter exceeds the threshold, Lemma 7.4.4 shows that receiving the negative reward before the positive one yields a larger net increase in the parameter. This motivates a local modification of the sequence obtained by swapping the rewards at times m and $m+1$, while keeping all other rewards unchanged. Formally, define the alternative sequence $\{\widehat{R}_n^W\}$ by

$$\widehat{R}_n^W = \begin{cases} \widetilde{R}_n^W, & n \neq m, m+1, \\ -1, & n = m, \\ +1, & n = m+1. \end{cases}$$

Let $\widehat{\theta}_n^W$ denote the SGB parameters corresponding to this modified sequence. Since $I_-^{\mathcal{Z}}(S_{m-1}^W) =$

1, both rewards are feasible at these positions, and thus the swapped sequence still belongs to $\mathcal{Z}$. By Lemma 7.4.4, reversing the order of $(+1,-1)$ to $(-1,+1)$ strictly increases the parameter after two steps, yielding

$$\widetilde{\theta}^W_{m+2} < \widehat{\theta}^W_{m+2}.$$

Since $\widehat{R}^W_n = \widetilde{R}^W_n$ for all $n > m+1$, the ordering is preserved and the final parameter satisfies

$$\widehat{\theta}^W_{n_1+1} > \widetilde{\theta}^W_{n_1+1},$$

contradicting the assumption that $\widetilde{\theta}^W_{n_1+1}$ was maximal.

Case 2. The second case,

$$\widetilde{R}^W_m = -1, \qquad \widetilde{\theta}^W_m < \frac{K-1}{K}\log(K-1), \qquad \text{and} \qquad I^{\mathcal{Z}}_+(S^W_{m-1}) = 1,$$

is handled analogously. One first shows that $\widetilde{R}^W_{m+1} = +1$, and then swaps the reward pair $(-1,+1)$ to $(+1,-1)$. Since the parameter is below the threshold, Lemma 7.4.4 again implies that this swap strictly increases the parameter after two steps, leading to the same contradiction.

Conclusion: In both cases, assuming a sequence different from $\{\widetilde{R}^M_t\}_{t=1}^{n_1}$ leads to a contradiction. Hence, $\{\widetilde{R}^M_t\}_{t=1}^{n_1}$ must be the maximizing sequence. □

Step 4: Application to phase 1. The next corollary characterizes the maximal possible value of the parameter $\widetilde{\theta}_{n_1+1}$ under the event $\mathcal{S}_1$, with the initial parameter $\widetilde{\theta}_1$ determined by the outcome of phase 0. It is a direct application of Lemma 7.4.6 to phase 1.

Recall from (7.7) that, at the end of phase 0, the SGB parameter satisfies

$$\widetilde{\theta}_{n_0+1} \leqslant -\frac{K-1}{K}\,\eta n_0.$$

In the notation of this subsection, the beginning of phase 1 corresponds to reindexing time so that $\widetilde{\theta}_{n_0+1}$ is denoted by $\widetilde{\theta}_1$. Thus, under $\mathcal{S}_0$, the initial parameter for phase 1 satisfies $\widetilde{\theta}_1 \leqslant -\frac{K-1}{K}\,\eta n_0$.

This sufficiently negative initialization ensures that throughout phase 1 the parameter remains below the threshold $\frac{K-1}{K}\log(K-1)$. Consequently, by Lemma 7.4.6, the maximizing reward

sequence under $\mathcal{S}_1$ is obtained by placing a $+1$ reward whenever feasible, subject to the constraints defining $\mathcal{S}_1$. Since $\mathcal{S}_1$ requires that the number of $+1$ rewards never exceeds the number of -1 rewards and that the empirical mean remains below $-\alpha\Delta$ (fixing the ratio number of total $+1$ and -1 rewards), the maximizing sequence consists of alternating rewards as long as permitted, followed by only -1 rewards once the number of $+1$ rewards has run out.

Corollary 7.4.8. *Assume that*

$$\widetilde{\theta}_1 \leqslant -\frac{K-1}{K} \cdot \eta n_0 \qquad \text{and} \qquad n_0 \geqslant \frac{1}{\eta}\left(\frac{K}{K-1}\right)^2 \log\!\left(\frac{K^3 n_1}{(K-1)^2}\right).$$

Then, under the event $\mathcal{S}_1$, it holds that

$$\widetilde{\theta}_{n_1+1} \;\leqslant\; \widetilde{\theta}^M_{n_1+1} \;\leqslant\; -\eta\alpha\Delta n_1\left(1 - \frac{1}{K-1} e^{-\frac{K-1}{K}\eta n_0}\right),$$

where $\widetilde{\theta}^M_{n_1+1}$ is the maximising parameter obtained from the reward sequence

$$\widetilde{R}^M_t = \begin{cases} (-1)^t, & \text{if } t \leqslant (1-\alpha\Delta)\, n_1, \\ -1, & \text{otherwise.} \end{cases}$$

The above corollary highlights the role of phase 0 in ensuring that the initial parameter $\widetilde{\theta}_1$ at the start of phase 1 is sufficiently negative. This guarantees that even when the maximizing sequence alternates $+1$ and -1 rewards early in phase 1, which can be shown to increase the parameter overall, the parameter $\widetilde{\theta}^M_t$ remains below the critical threshold $\frac{K-1}{K}\log(K-1)$ for all t. Once the sequence reaches the portion with consecutive -1 rewards, the parameter can then be driven down as intended, producing the maximal negative value consistent with the constraints of $\mathcal{S}_1$.

Proof of Corollary 7.4.8. We want to characterize the maximal possible value of $\widetilde{\theta}_{n_1+1}$ under $\mathcal{S}_1$, given that the initial parameter $\widetilde{\theta}_1$ is determined by the outcome of phase 0. The strategy is to apply Lemma 7.4.6 to identify the maximizing sequence within phase 1, and then control the effect of early reward alternation using the negativity of $\widetilde{\theta}_1$.

Part I: Identify when alternation occurs. By Definition 7.4.3, the feasibility index $I^{\mathcal{Z}}_+(S^M_{t-1}) = 0$ if either

- the number of $+1$ rewards observed up to time t reaches $\frac{1-\alpha\Delta}{2} \cdot n_1$, or
- the number of $+1$ and -1 rewards observed up to time t are equal.

We assumed that $\widetilde{\theta}_1 \leqslant -\frac{K-1}{K} \cdot \eta n_0 < 0 < \frac{K-1}{K} \log(K-1)$.

By Lemma 7.4.6, as long as $\widetilde{\theta}_t^M < \frac{K-1}{K} \log(K-1)$ and $I_+^{\mathcal{Z}}(S_{t-1}^M) = 1$, the maximizing reward is $\widetilde{R}_t^M = +1$. However, due to the second condition above, when the numbers of $+1$ and -1 rewards are equal, $I_+^{\mathcal{Z}}(S_{t-1}^M) = 0$, which forces $\widetilde{R}_t^M = -1$. This gives the alternating sequence of rewards

$$\widetilde{R}_t^M = (-1)^t$$

for the early part of phase 1 as long as $\widetilde{\theta}_t^M < \frac{K-1}{K} \log(K-1)$.

Part II: Control the parameter during alternation. During an alternating sequence of rewards, the SGB parameter tends to drift upward due to the asymmetry of the update rule

$$\widetilde{\theta}_{t+1} = \widetilde{\theta}_t + \eta(1 - \widetilde{p}_t)\, \widetilde{R}_t.$$

Indeed, receiving a $+1$ reward increases $\widetilde{\theta}_t$ and therefore increases $\widetilde{p}_{t+1}$. As a consequence, the subsequent -1 reward is multiplied by the smaller factor $(1 - \widetilde{p}_{t+1})$, so it does not fully cancel the previous increase. A similar phenomenon occurs with the order reversed. Repeating this alternation therefore leads to a net upward drift of the parameter.

We now show that if phase 0 is sufficiently long, i.e., if n_0 is large, then the initial parameter $\widetilde{\theta}_1$ is negative enough to control this effect. In particular, $\widetilde{\theta}_1$ can be chosen so that throughout the entire alternating phase the cumulative upward drift remains limited, preventing the parameter from crossing the threshold $\frac{K-1}{K} \log(K-1)$. This ensures that we remain in the case of Lemma 7.4.6 where $+1$ is a maximising during the alternation.

Intermediate claim: If

$$n_0 \geqslant \frac{1}{\eta}\Big(\frac{K}{K-1}\Big)^2 \log\Big(\frac{K^3 n_1}{(K-1)^2}\Big),$$

then for all $n \leqslant \frac{1-\alpha\Delta}{2} \cdot n_1$,

$$\widetilde{\theta}_{2n+1}^M \leqslant \frac{K-1}{K}\widetilde{\theta}_1 < 0.$$

Part II-a: Proof of intermediate claim by induction. Recall the SGB update:

$$\widetilde{\theta}_{t+1} = \widetilde{\theta}_t + \eta(1 - \widetilde{p}_t) \cdot \widetilde{R}_t^M,$$

so a $+1$ reward increases the parameter, a -1 reward decreases it.

Base case ($n = 0$): $\widetilde{\theta}_1 \leqslant 0$ implies $\widetilde{\theta}_1 \leqslant \frac{K-1}{K}\widetilde{\theta}_1$, so the claim holds.

Inductive step: Assume the claim holds for all $k \leqslant n < \frac{1-\alpha\Delta}{2} \cdot n_1$. Then the alternating sequence has continued through $t \leqslant 2(n+1)$, i.e.,

$$\widetilde{R}_t^M = (-1)^t, \quad t \leqslant 2(n+1),$$

and the parameter remains negative. Compute the parameter after $2(n+1)$ rewards:

$$\begin{aligned}
\widetilde{\theta}^M_{2(n+1)+1} &= \widetilde{\theta}_1 + \sum_{t=1}^{2(n+1)} \eta\,(1 - \widetilde{p}_t^M) \cdot \widetilde{R}_t^M \\
&= \widetilde{\theta}_1 + \eta \sum_{k=1}^{n+1} \left[(1 - \widetilde{p}^M_{2k}) - (1 - \widetilde{p}^M_{2k-1})\right] \\
&= \widetilde{\theta}_1 + \eta \sum_{k=1}^{n+1} \left[\widetilde{p}^M_{2k-1} - \widetilde{p}^M_{2k}\right]. \qquad (7.8)
\end{aligned}$$

Using the definition of $\widetilde{p}_t^M$, we bound each term:

$$\begin{aligned}
\widetilde{p}^M_{2k-1} - \widetilde{p}^M_{2k} &= \frac{1}{1 + (K-1) \cdot e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}}} - \frac{1}{1 + (K-1) \cdot e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k}}} \\
&= (K-1) \cdot \frac{e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k}} - e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}}}{(1 + (K-1)e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}})(1 + (K-1)e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k}})} \\
&= \frac{1}{K-1} \frac{(K-1)e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}}}{1 + (K-1)e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}}} \cdot \frac{(K-1)e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k}}}{1 + (K-1)e^{-\frac{K}{K-1}\widetilde{\theta}^M_{2k}}} \cdot \left(e^{\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}} - e^{\frac{K}{K-1}\widetilde{\theta}^M_{2k}}\right) \\
&= \frac{1}{K-1}(1 - \widetilde{p}^M_{2k-1}) \cdot (1 - \widetilde{p}^M_{2k}) \cdot \left(e^{\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}} - e^{\frac{K}{K-1}\widetilde{\theta}^M_{2k}}\right) \\
&\leqslant \frac{1}{K-1} \exp\Big(\frac{K}{K-1}\widetilde{\theta}^M_{2k-1}\Big) \\
&\leqslant \exp\Big(\widetilde{\theta}_1\Big),
\end{aligned}$$

using the inductive hypothesis $\widetilde{\theta}^M_{2k-1} \leqslant \frac{K-1}{K}\widetilde{\theta}_1$. Summing over k and substituting in (7.8), we get

$$\widetilde{\theta}^M_{2(n+1)+1} \leqslant \widetilde{\theta}_1 + \eta(n+1)e^{\widetilde{\theta}_1}. \qquad (7.9)$$

Finally, the choice of n_0 guarantees that

$$\begin{aligned} n_0 \geqslant \frac{1}{\eta}\Big(\frac{K}{K-1}\Big)^2 \log\Big(\frac{K^3 n_1}{(K-1)^2}\Big) &\implies \frac{K^3}{(K-1)^2} n_1 \leqslant n_0 e^{\eta n_0 (K-1)^2/K^2} \\ &\implies \eta n_1 \leqslant \frac{1}{K}\Big(n_0 \eta \frac{(K-1)^2}{K^2}\Big) e^{-\widetilde{\theta}_1} \quad \text{using } \widetilde{\theta}_1 \leqslant -\Big(\frac{K-1}{K}\Big)^2 \eta n_0 \\ &\implies \eta n_1 e^{\widetilde{\theta}_1} \leqslant -\frac{1}{K}\widetilde{\theta}_1 \quad \text{using again } \widetilde{\theta}_1 \leqslant -\Big(\frac{K-1}{K}\Big)^2 \eta n_0 \\ &\implies \eta (n+1) e^{\widetilde{\theta}_1} \leqslant -\frac{1}{K}\widetilde{\theta}_1, \end{aligned}$$

using that $n \leqslant \frac{1-\alpha\Delta}{2} \cdot n_1 < \frac{n_1}{2} \leqslant n_1 - 1$. So (7.9) becomes

$$\widetilde{\theta}^M_{2(n+1)+1} \leqslant \frac{K-1}{K}\widetilde{\theta}_1 < 0,$$

completing the induction.

We have shown that for all $n \leqslant \frac{1-\alpha\Delta}{2} \cdot n_1$,

$$\widetilde{\theta}^M_{2n+1} \leqslant \frac{K-1}{K}\widetilde{\theta}_1 < 0.$$

Part III: Complete the maximizing sequence. By the intermediate claim, the alternating sequence of rewards persists while the parameter satisfies $\widetilde{\theta}^M_t < \frac{K-1}{K}\log(K-1)$. This continues until the number of observed $+1$ rewards reaches its maximum allowable value, $\frac{1-\alpha\Delta}{2} \cdot n_1$, which occurs at time $t_0 = (1-\alpha\Delta) \cdot n_1$. At this point, no further $+1$ rewards are feasible, so $I^{\mathcal{Z}}_+(S^M_{t_0-1}) = 0$. From time $t_0 + 1$ onward, the maximizing sequence therefore assigns reward -1 at every remaining round.

Step IV: Bound the final parameter. Let

$$t_0 := (1-\alpha\Delta)\, n_1$$

denote the time at which the alternating sequence ends, i.e., the point at which the maximum allowable number $\frac{1-\alpha\Delta}{2} \cdot n_1$ of $+1$ rewards has been observed and $I^{\mathcal{Z}}_+(S^M_{t_0-1}) = 0$. From this time onward, there remain

$$n_1 - t_0 = \alpha\Delta\, n_1$$

rounds in phase 1, and the maximizing sequence assigns reward -1 in each of these rounds.

By the intermediate claim proved in Step 2, throughout the alternating phase we have

$$\widetilde{\theta}_{t_0}^M \leqslant \frac{K-1}{K}\,\widetilde{\theta}_1.$$

Recalling the bound on the initial parameter inherited from phase 0,

$$\widetilde{\theta}_1 \leqslant -\frac{K-1}{K}\eta n_0,$$

it follows that

$$\widetilde{\theta}_{t_0}^M \leqslant -\Big(\frac{K-1}{K}\Big)^2 \eta n_0.$$

We now bound the probability term appearing in the SGB update. For any $t \geqslant 0$, monotonicity of $\widetilde{\theta}_t^M$ during the all -1 phase yields

$$\widetilde{\theta}_{t_0+t}^M \leqslant \widetilde{\theta}_{t_0}^M,$$

and therefore for all $t \in \{1, ..., n_1 - t_0\}$

$$\begin{aligned}
\widetilde{p}_{t_0+t}^M &= \frac{1}{1+(K-1)e^{-\frac{K}{K-1}\widetilde{\theta}_{t_0+t}^M}} \\
&\leqslant \frac{1}{K-1}\exp\Big(\frac{K}{K-1}\widetilde{\theta}_{t_0+t}^M\Big) \\
&\leqslant \frac{1}{K-1}\exp\Big(\frac{K}{K-1}\widetilde{\theta}_{t_0}^M\Big) \\
&\leqslant \frac{1}{K-1}\exp\Big(-\frac{K-1}{K}\eta n_0\Big) =: \delta_0.
\end{aligned}$$

Hence, the final parameter satisfies

$$\begin{aligned}
\widetilde{\theta}_{n_1+1}^M &= \widetilde{\theta}_{t_0+1}^M - \eta \sum_{k=t_0+1}^{n_1} (1-\widetilde{p}_k^M) \\
&\leqslant -\eta(n_1-t_0)(1-\delta_0) \\
&= -\eta\alpha\Delta n_1(1-\delta_0),
\end{aligned}$$

which proves the corollary. □

Step 5: Final probability control. With the preceding results in place, we can now derive explicit sufficient conditions on the lengths of phases 0 and 1 that guarantee the desired small-probability event at the end of phase 1 and finishes this section.

Lemma 7.4.9. *Fix any $\epsilon \in (0,1)$, and define n_0 and n_1 by*

$$n_0 = \left\lceil \frac{1}{\eta} \cdot \frac{K}{K-1} \cdot \log\Big(\frac{1}{\epsilon(K-1)}\Big) \right\rceil \vee \left\lceil \frac{1}{\eta}\Big(\frac{K}{K-1}\Big)^2 \log\Big(\frac{K^3 n_1}{(K-1)^2}\Big) \right\rceil, \tag{7.10}$$

$$n_1 = \left\lceil \frac{K-1}{K} \cdot \frac{\log(2T/(K-1))}{\eta\Delta(1-\epsilon)} + \frac{2}{\Delta} \right\rceil. \tag{7.11}$$

Then, under the event $\mathcal{S}_0 \cap \mathcal{S}_1$, it holds that

$$\widetilde{p}_{n_0+n_1+1} \leqslant \frac{1}{2T}.$$

Proof. The proof proceeds in three steps: (i) transfer the outcome of phase 0 into a suitable initial condition for phase 1; (ii) apply the worst-case characterization from Corollary 7.4.8; (iii) convert the resulting bound on the parameter into a bound on the probability.

Part I: Initial condition at the start of phase 1. On the event $\mathcal{S}_0$, the outcome of phase 0 from (7.7) satisfies

$$\widetilde{\theta}_{n_0+1} \leqslant -\frac{K-1}{K}\,\eta n_0.$$

Throughout phase 1 we relabel $\widetilde{\theta}_{n_0+1}$ as the initial parameter $\widetilde{\theta}_1$. By construction of n_0, we have

$$n_0 \geqslant \frac{1}{\eta}\Big(\frac{K}{K-1}\Big)^2 \log\Big(\frac{K^3 n_1}{(K-1)^2}\Big),$$

so the conditions required to apply Corollary 7.4.8 are satisfied.

Part II: Worst-case evolution during phase 1. Applying Corollary 7.4.8 to the rewards in phase 1, viewing $\widetilde{\theta}_{n_0+1}$ as the initial parameter $\widetilde{\theta}_1$ of that corollary (since phase 1 starts at global time n_0+1), yields

$$\widetilde{\theta}_{n_0+n_1+1} \leqslant -\eta\alpha\Delta n_1 \left(1 - \frac{1}{K-1} e^{-\frac{K-1}{K}\eta n_0}\right).$$

The first term in the definition (7.10) of n_0 ensures that

$$\frac{1}{K-1} e^{-\frac{K-1}{K}\eta n_0} \leqslant \epsilon,$$

and therefore

$$\widetilde{\theta}_{n_0+n_1+1} \leqslant -\eta\alpha\Delta(1-\epsilon)\, n_1.$$

Next, recalling that $\alpha > 1 - \frac{2}{\Delta n_1}$ (from 7.6), we obtain

$$\widetilde{\theta}_{n_0+n_1+1} \leqslant -\eta\alpha\Delta(1-\epsilon)\, n_1 \leqslant -\eta\Delta(1-\epsilon)\, n_1 + 2\eta(1-\epsilon). \tag{7.12}$$

By the definition (7.11) of n_1, we have

$$\begin{aligned} n_1 &\geqslant \frac{K-1}{K} \cdot \frac{\log(2T/(K-1))}{\eta\Delta(1-\epsilon)} + \frac{2}{\Delta} \\ \implies \eta\Delta(1-\epsilon)n_1 &\geqslant \frac{K-1}{K} \log\left(\frac{2T}{K-1}\right) + 2\eta(1-\epsilon). \end{aligned}$$

Substituting into (7.12),

$$\begin{aligned} \widetilde{\theta}_{n_0+n_1+1} &\leqslant -\eta\Delta(1-\epsilon)n_1 + 2\eta(1-\epsilon) \\ &\leqslant -\frac{K-1}{K} \log\left(\frac{2T}{K-1}\right) - 2\eta(1-\epsilon) + 2\eta(1-\epsilon) \\ &= -\frac{K-1}{K} \log\left(\frac{2T}{K-1}\right). \end{aligned}$$

Part III: From parameter bound to probability bound. Finally, using the definition of $\widetilde{p}_t$,

$$\begin{aligned} \widetilde{p}_{n_0+n_1+1} &= \frac{e^{\widetilde{\theta}_{n_0+n_1+1}}}{e^{\widetilde{\theta}_{n_0+n_1+1}} + (K-1)e^{-\frac{1}{K-1}\widetilde{\theta}_{n_0+n_1+1}}} \\ &\leqslant \frac{1}{K-1} \exp\Big(\frac{K}{K-1}\widetilde{\theta}_{n_0+n_1+1}\Big) \\ &\leqslant \frac{1}{K-1} \exp\Big(-\log\big(\tfrac{2T}{K-1}\big)\Big) = \frac{1}{2T}. \end{aligned}$$

This concludes the proof. □

7.5 Conclusion

This chapter sheds light on the theoretical properties of SGB, a simple policy gradient algorithm for bandits, whose behaviour is governed by a learning rate η. For two-armed bandits, we show that knowledge of a lower bound on the sub-optimality gap Δ is necessary to achieve logarithmic regret. Along with the other results of Baudry et al. (2025), this establishes Δ as the critical threshold between logarithmic and polynomial regret in the time horizon for two-armed bandits. Beyond two arms, it remains to provide a similar characterisation of the regret for general K-armed bandits where the threshold is conjectured to be $2\Delta/K$ in Baudry et al. (2025) (for both

upper and lower bounds). The general bandit setting being understood, a natural next step would be to move to the RL multi-state MDP setting, which is however significantly more challenging.

Chapter 8

Optimal Convergence Rate for Exact Policy Mirror Descent in Discounted MDPs

The problem of finding an optimal policy in tabular discounted Markov Decision Processes (MDPs) was classically solved using dynamic programming approaches (see Chapter 3) such as policy iteration (PI) and value iteration (VI) (Puterman, 1994; Sutton and Barto, 2018). These methods are well understood theoretically and are guaranteed to converge linearly to the optimal policy in the tabular setting with a rate equal to the discount factor γ of the MDP (Bellman, 1957), i.e. $\|V^{\pi^\star} - V^{\pi^k}\|_\infty \leqslant C\gamma^k$ for some $C > 0$ where V^π is the discounted value of the policy in the MDP and π^k is the policy produced by PI or VI after k iterations. In this chapter, continuing our effort to better understand the performance of gradient-based methods, we consider the problem of finding an optimal policy through the lens of policy gradient methods (see Chapter 3) and aim to determine whether their guarantees can match those of classical algorithms in tabular MDPs. In particular, we view the sequential decision-making process in *policy space*, where the learner selects a policy at each iteration and observes the resulting value function, rather than acting directly in the environment. The goal is not to minimise regret over rounds, but rather to produce a *final* policy that is near-optimal.

Among the recent works focused on understanding policy-gradient methods in the tabular setting, Xiao (2022) study a general family of algorithms based on the gradient-based mirror descent

method (see Chapter 2) known as Policy Mirror Descent (PMD). PMD can be interpreted as algorithmically regularising the policy improvement step of policy iteration (PI), and can therefore be seen as a form of regularised PI, without actually modifying the original objective. A notable instance of PMD is the Natural Policy Gradient (NPG) algorithm (Kakade, 2001), which underlies many widely used methods in modern reinforcement learning (Schulman et al., 2015, 2021). This makes the study of PMD particularly interesting because it can allow us to not only better understand NPG but also discover closely related algorithms which may share similar properties while having additional desirable features as well.

Under exact gradients, linear convergence of PMD was established by Xiao (2022), though their rate depends on an instance-dependent factor that can scale poorly with the dimension of the problem, and can lead to an iteration complexity scaling with the size of the state space (see Section 8.10.3). The γ-rate of classical approaches (PI and VI) can be achieved by PMD when the MDP objective is regularised (Cen et al., 2021; Lan, 2021; Zhan et al., 2023). However, the classical approaches achieve the γ-rate without MDP-regularisation, revealing that regularisation is, in general, not necessary for algorithms to reach the γ-rate. In fact, Khodadadian et al. (2021) showed that NPG, a special case of PMD, can achieve the γ-rate, suggesting that such rates are attainable by instances of PMD without regularisation. This motivates the following questions:

Can the classical dimension-independent linear γ-rate be matched by policy-gradient algorithms in unregularised MDPs ? And what is the best rate that they can achieve ?

For PMD, our work answers the first question positively: **our results establish that in the exact gradient setting, policy mirror descent can converge with a dimension-independent linear γ-rate, even on high-dimensional problems.** We answer the second by establishing that the γ-rate is in fact the best rate achievable for PMD as well as for a more general family of algorithms (see Section 8.5). We also extend our results to the inexact setting where gradients can be estimated using a generative model and improve the sample complexity of PMD to the optimal dependence on the size of the state-action space (see contributions below). We highlight our contributions below and summarise some of them in Table 8.1.

Contributions

- We recover the γ-rate for the general family of PMD algorithms with exact policy evalua-

tions under an adaptive step-size (see the third bullet point below). In particular, Theorem 8.4.1 establishes the following bound in ℓ_∞-norm for the value V^{π^k} of the policy π^k after k iterations of PMD compared to the value $V^{\pi^\star}$ of an optimal policy $\pi^\star$,

$$\|V^{\pi^\star} - V^{\pi^k}\|_\infty \leqslant \frac{2}{1-\gamma}\gamma^k,$$

providing guarantees for any starting-state distribution. This matches the rate of VI and PI as well as the best known rates for PMD on regularised MDPs. This is also the first fully dimension-independent linear convergence result for PMD on unregularised MDPs, by which we mean that there is no dependence on the size of the state space or the action space.

- We provide a matching non-asymptotic lower-bound in Theorem 8.5.1, establishing the γ-rate as the optimal rate for PMD methods in early iterations. Our results show that a particular choice of learning rate allows PMD to reach this lower-bound exactly. Note that an asymptotic lower-bound is not possible due to the exact convergence of PI in finite iterations (see Section 8.5 for a discussion of the significance of this non-asymptotic lower-bound).

- The γ-rate for PMD in Theorem 8.4.1 relies on an adaptive step-size, where the adaptivity comes from the fact that the step-size depends on the policy at the current iteration (see Section 8.4). In Theorem 8.6.1 we show that this adaptivity is necessary for PMD to achieve the γ-rate, establishing our step-size as both sufficient and necessary.

- We establish a novel theoretical analysis that avoids the use of the performance difference lemma (Lemma 8.3.1, Kakade and Langford (2002a)). This leads to a simple analysis and avoids needing to deal with visitation distribution mismatches that lead to dimension dependence in prior work.

- By extending our analysis to the inexact setting, with a similar approach to Xiao (2022), we establish an instance-independent sample complexity of $\tilde{O}(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-8}\varepsilon^{-2})$ under a generative model, where the notation $\tilde{O}()$ hides poly-logarithmic factors, $\mathcal{S}$ is the state space of the MDP, $\mathcal{A}$ is the action space and ε is the required accuracy. This improves on the previous best known sample complexity for PMD by removing the dependence on a distribution mismatch coefficient that can scale with problem-dependent quantities such as the size of the state space and as such provides the optimal dependence on $|\mathcal{S}||\mathcal{A}|$ (Azar

	Linear γ-Rate	General Mirror Map	ℓ_∞ Bound	Dimension Independent
Khodadadian et al. (2021)	✓	×	✓	✓
Xiao (2022)	×	✓	×	×
This work	✓	✓	✓	✓

Table 8.1: Comparison of contributions with prior work that study PMD. Khodadadian et al. (2021) focus on NPG, an instance of PMD for a specific mirror map (see Section 8.1). Their analysis is fundamentally different to ours as it exploits the closed-form update of NPG. Their step-size is similar to ours, although it has a dependence on a sub-optimality gap (see Section 8.4). Linear γ-rate holds if the linear convergence is with the γ-rate, not only linear convergence. The ℓ_∞-bound is satisfied if it holds for $\|V^{\pi^\star} - V^{\pi^k}\|_\infty$. Dimension independence is satisfied when there is no instance for which the bound can scale with the size of the state or action space. We compare these works in more detail in Section 8.4.

et al., 2013). More generally, we highlight that the analysis we establish in the exact setting can easily be combined with any other scheme for estimating the Q functions (see Section 8.7), paving the way for further improvements in instance-independent sample complexity results should more efficient estimation procedures be developed.

Our contributions are primarily on establishing the optimal rate for general (not just NPG) *exact* PMD where we assume access to the true action-values of policies, and the simplicity of the analysis. The sample complexity result in the inexact setting illustrates how our analysis can be easily extended to obtain improved results for inexact PMD. These contributions advance the theoretical understanding of PMD and rule out searching for instances of PMD that may improve on PI or NPG in the exact setting.

Roadmap. In Section 8.1, we define the problem setting. In Section 8.2, we present the PMD method as an instance of mirror descent. In Section 8.3, we outline properties of PMD. In Section 8.4, we present our main result on the convergence of PMD with linear γ-rate and adaptive step-size. We show in Section 8.5 that the γ-rate is optimal and in Section 8.6 that the adaptive step-size is necessary. In Section 8.7, we extend our result to sample complexity bounds with inexact gradients. In Section 8.8, we discuss related works. In Section 8.9, we conclude and discuss future-work. Finally, in Section 8.10, we provide missing details and proofs omitted earlier in the chapter.

Bibliographic remark: This chapter is based on Johnson et al. (2023) and is my own work.
It is licensed under a Creative Commons Attribution 4.0 International License.

Setting	Performance Criteria	Main Theorem(s)	Publication Source
Tabular RL (known transitions)	Best-Policy Identification	Theorem 8.4.1, Theorem 8.5.1	Johnson et al. (2023)

Table 8.2: Chapter overview.

8.1 Problem Setting

We refer the reader to Chapter 3 for the definition of a Markov Decision Process (MDP) and associated concepts and notation.

We consider tabular discounted MDPs where the state space $\mathcal{S}$ and action space $\mathcal{A}$ are both finite ($|\mathcal{S} \times \mathcal{A}| < \infty$), the discount factor γ is in $[0, 1)$ and the reward function $r(s, a) \in [0, 1]$ is deterministic. We recall the definition of the discounted visitation-distribution for starting state s' and policy π. For all $s \in \mathcal{S}$,

$$d^{\pi}_{s'}(s) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}^{\pi}(s_t = s | s_0 = s'), \tag{8.1}$$

We also write $d^{\pi}_{\rho}(s) = \sum_{s' \in \mathcal{S}} \rho(s') d^{\pi}_{s'}(s)$.

We are interested in the planning problems (see Section 3.2), that is, finding an ε-optimal policy, i.e a policy π such that $\|V^{\pi^\star} - V^{\pi}\|_\infty < \varepsilon$. We assume the transition dynamics $p(\cdot \mid s, a)$ and reward function r of the MDP are known and that the value and action-value functions of any policy can be computed exactly via (3.3).

8.2 Exact Policy Mirror Descent

We are interested in policy gradient methods that are based on optimising a parameterised policy π_θ with respect to $V^{\pi_\theta}(\rho)$ for some $\rho \in \Delta(\mathcal{S})$. In the tabular setting, we can use the direct parameterisation of a policy π_θ, which associates a parameter to each state-action pair, i.e. we have $\pi_\theta(a|s) = \theta_{s,a}$. We will drop the subscript θ for notational convenience. The gradient of the value function with respect to this parameterisation (see Section 3.2.2) is given for each state-action pair (s,a) by

$$\frac{\partial}{\partial \pi(a|s)} V^{\pi}(\rho) = \frac{1}{1-\gamma} d^{\pi}_{\rho}(s) Q^{\pi}(s, a). \tag{8.2}$$

Mirror Descent (MD, Beck and Teboulle (2003)) carries out gradient descent in a geometry that

is non-Euclidean (see Chapter 2 - here we consider the offline version of Online Mirror Descent, which is a special case where the losses are constant across rounds). Using $-V^{\pi}(\rho)$ as the minimising objective, the proximal perspective of MD gives an update of the form

$$\pi^{k+1} = \operatorname{argmin}_{p\in\Pi}\Big\{ -\eta_k\langle \nabla V^{\pi^k}(\rho), p\rangle + D_h(p,\pi^k)\Big\} \tag{8.3}$$

where $h : \operatorname{dom} h \to \mathbb{R}$ is the mirror map (with $\Pi \subset \operatorname{dom} h$) and D_h is the Bregman divergence generated by h. We require h to be of Legendre type (Rockafellar, 1970), i.e strictly convex and essentially smooth (differentiable and $\|\nabla h(x_k)\| \to \infty$ for any sequence x_k converging to a point on the boundary of dom h) on the relative interior (rint) of dom h. The Bregman Divergence is defined as

$$D_h(\pi,\pi') = h(\pi) - h(\pi') - \langle \nabla h(\pi'), \pi - \pi'\rangle \qquad \text{for } \pi,\pi' \in \operatorname{dom} h.$$

As the objective $V^{\pi}(\rho)$ is non-convex in general (Agarwal et al., 2021), usual techniques from convex theory (Bubeck, 2015) are not applicable.

In discounted MDPs, the presence of the visitation-distribution term $d_\rho^\pi(s)$ in the gradient of the objective in (8.2) can slow down learning because it can lead to vanishingly small gradients when states are infrequently visited under the current policy π (Agarwal et al., 2021). To circumvent this issue, Policy Mirror Descent (PMD) (Lan, 2021; Shani et al., 2020; Xiao, 2022) applies a variant of update (8.3) with a weighted Bregman divergence D_h^{PMD} that matches the visitation distribution factors of the gradient $D_h^{\mathrm{PMD}}(p,\pi^k) = \sum_s d_\rho^{\pi^k}(s) D_h(p_s,\pi_s^k)$ where the mirror map h is now defined on a subset of $\mathbb{R}^{|\mathcal{A}|}$. The resulting update has for all states a factor of $d_\rho^{\pi^k}(s)$ in both terms. The minimisation can then be applied for each state individually to get the PMD update

$$\pi_s^{k+1} = \operatorname{argmin}_{p\in\Delta(\mathcal{A})}\Big\{ -\eta_k\langle Q_s^{\pi^k}, p\rangle + D_h(p,\pi_s^k)\Big\} \tag{8.4}$$

for all states s. Exact PMD iteratively applies update (8.4) for some sequence of step-sizes $\eta_k > 0$ and initial policy $\pi^0 \in \operatorname{rint}\Pi$. We call this algorithm exact because we assume access to the true state-action values Q^k. The update (8.4) of PMD considered in this chapter uses the true action-value Q^π. We will often add a superscript k to any quantity that is associated to π^k. For example, $V^k(s) = V^{\pi^k}(s)$. Similarly for $\pi^\star$ and the superscript $\star$.

PMD is a general family that covers many algorithms, specified by the choice of mirror map

h. These will inherit the guarantees of PMD, which motivates the study of the convergence guarantees of PMD beyond specific instances. Taking h to be the negative entropy yields NPG, whose theoretical properties have attracted a lot of interest (Agarwal et al., 2021; Cen et al., 2021; Khodadadian et al., 2021). With a null Bregman Divergence, PMD recovers PI. PI is generated by a constant mirror map, which is not of Legendre type but the analysis still applies so all results on PMD remain valid for PI. In fact, PMD can be viewed as a form of regularised PI since the update (8.4) converges to a PI update as $\eta_k \to \infty$, regardless of the mirror map. Beyond these, providing mirror maps that generate other Bregman Divergences will lead to different algorithms. In particular, every exponential family has a corresponding mirror map generating a unique Bregman Divergence (Banerjee et al., 2005), highlighting the generality of PMD.

In prior work, PMD has sometimes been applied to regularised MDPs (Lan, 2021) where the action-value is augmented with regularisation and is no longer the true action-value. This is a different algorithm that converges to a policy that is optimal in the regularised MDP but not in the original unregularised MDP.

The focus of this chapter is to understand how exact PMD performs in finding the optimal policy of a discounted MDP.

8.3 Properties of PMD

We begin by presenting some lemmas relevant to the analysis of PMD. First is the performance difference lemma (Kakade and Langford, 2002a), which relates the difference in values of policies to the policies themselves. The proof can be found in their paper under Lemma 6.1.

Lemma 8.3.1 (Performance Difference Lemma)**.** *For any $\pi, \pi' \in \Pi$, we have*

$$V^{\pi}(\rho) - V^{\pi'}(\rho) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_\rho^\pi}\Big[\langle Q_s^{\pi'}, \pi_s - \pi'_s\rangle\Big].$$

Next is the Three-Point Descent Lemma, which relates the improvement of the proximal gradient update compared to an arbitrary point. It originally comes from Chen and Teboulle (1993) (Lemma 3.2) where a proof can be found, though we use a slightly modified version from Xiao (2022) (Lemma 6).

Lemma 8.3.2 (Three-Point Descent Lemma, Lemma 6 in Xiao (2022))**.** *Suppose that $\mathcal{C} \subset \mathbb{R}^n$*

is a closed convex set, $\phi : \mathcal{C} \to \mathbb{R}$ is a proper, closed convex function, $D_h(\cdot,\cdot)$ is the Bregman divergence generated by a function h of Legendre type and rint dom $h \cap \mathcal{C} \neq \varnothing$. For any $x \in$ rint dom h, let

$$x^+ = \mathit{argmin}_{u\in\mathcal{C}}\{\phi(u) + D_h(u,x)\}. \tag{8.5}$$

Then $x^+ \in$ rint dom $h \cap \mathcal{C}$ and $\forall u \in \mathcal{C}$,

$$\phi(x^+) + D_h(x^+,x) \leqslant \phi(u) + D_h(u,x) - D_h(u,x^+) \tag{8.6}$$

The update (8.4) of PMD is an instance of the proximal minimisation (8.5) with $\mathcal{C} = \Delta(\mathcal{A})$, $x = \pi_s^k$ and $\phi(x) = -\eta_k\langle Q_s^k, x\rangle$. Plugging these into (8.6), Lemma 8.3.2 relates the decrease in the proximal objective of π_s^{k+1} to any other policy, i.e. $\forall p \in \Delta(\mathcal{A})$,

$$-\eta_k\langle Q_s^k, \pi_s^{k+1}\rangle + D_h(\pi_s^{k+1},\pi_s^k) \leqslant -\eta_k\langle Q_s^k, p\rangle + D_h(p,\pi_s^k) - D_h(p,\pi_s^{k+1}). \tag{8.7}$$

This equation is key to the analysis in the proof of our main result Theorem 8.4.1. In particular, it allows us to prove the following lemma regarding the monotonic improvement in action-value of PMD iterates.

Lemma 8.3.3 (Descent Property of PMD, Lemma 7 in Xiao (2022))**.** *Consider the policies produced by the iterative updates of PMD in (8.4). Then for any $k \geqslant 0$,*

$$Q^{k+1}(s,a) \geqslant Q^k(s,a), \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}.$$

Proof. Recall that the Three-Point Descent Lemma states that $\forall p \in \Delta(\mathcal{A})$,

$$-\eta_k\langle Q_s^k, \pi_s^{k+1}\rangle + D_h(\pi_s^{k+1},\pi_s^k) \leqslant -\eta_k\langle Q_s^k, p\rangle + D_h(p,\pi_s^k) - D_h(p,\pi_s^{k+1}).$$

Using this with $p = \pi_s^k$,

$$D_h(\pi_s^k,\pi_s^{k+1}) + D_h(\pi_s^{k+1},\pi_s^k) \leqslant \eta_k\langle Q_s^k, \pi_s^{k+1} - \pi_s^k\rangle$$

and since the Bregman divergences are none-negative and $\eta_k > 0$,

$$0 \leqslant \langle Q_s^k, \pi_s^{k+1} - \pi_s^k\rangle.$$

By the performance difference lemma (Lemma 8.3.1), for any $s \in \mathcal{S}$

$$V^{k+1}(s) - V^k(s) = \frac{1}{1-\gamma}\mathbb{E}_{s' \sim d_s^{k+1}}\Big[\langle Q_{s'}^k, \pi_{s'}^{k+1} - \pi_{s'}^k \rangle\Big] \geqslant 0,$$
$$\implies Q^{k+1}(s,a) - Q^k(s,a) = \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a)\Big(V^{k+1}(s') - V^k(s')\Big) \geqslant 0.$$

□

8.4 Linear Convergence of Exact PMD with γ-Rate

In this section, we present our main results on the linear convergence of exact PMD, where we recall that exact refers to the exact policy evaluation of the policies under consideration. The term linear convergence is somewhat misleading: it does not indicate a linear decrease in error, but rather a geometric one, with the error shrinking by a constant factor at each iteration.

We first introduce some relevant notation. Fix a state $s \in \mathcal{S}$ and an integer $k \geqslant 0$. Let $\mathcal{A}_s^k = \{a \in \mathcal{A} : Q^k(s,a) = \max_{a' \in \mathcal{A}} Q^k(s,a')\}$ denote the set of optimal actions in state s under policy π^k. Denote by $\widetilde{\Pi}_s^{k+1}$ the set of greedy policies w.r.t Q_s^k in state s, i.e $\widetilde{\Pi}_s^{k+1} = \Big\{p \in \Delta(\mathcal{A}) : \sum_{a \in \mathcal{A}_s^k} p(a) = 1\Big\}$. We are now ready to state our main result in the setting of exact PMD.

Theorem 8.4.1. *Let $\{c_k\}_{k \in \mathbb{Z}_{\geqslant 0}}$ be a sequence of positive reals. Consider applying iterative updates of (8.4) with $\pi^0 \in$ rint Π and step-sizes satisfying for all $k \geqslant 0$,*

$$\eta_k \geqslant \frac{1}{c_k} \max_{s \in \mathcal{S}} \Big\{ \min_{\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}} D_h(\widetilde{\pi}_s^{k+1}, \pi_s^k) \Big\}. \tag{8.8}$$

Then we have for all $k \geqslant 0$,

$$\|V^\star - V^k\|_\infty \leqslant \gamma^k \Big(\|V^\star - V^0\|_\infty + \sum_{i=1}^{k} \gamma^{-i} c_{i-1} \Big). \tag{8.9}$$

The sequence $\{c_k\}_{k \in \mathbb{Z}_{\geqslant 0}}$ plays an important role in both the step-size constraint (8.8) and the bound (8.9). In particular, different choices will lead to different guarantees:

- $c_i = c_0$ for some $c_0 > 0$ gives a step-size with a constant component. The resulting bound is

$$\|V^\star - V^k\|_\infty \leqslant \gamma^k \|V^\star - V^0\|_\infty + \frac{c_0}{1-\gamma},$$

which converges linearly up to some accuracy controlled by c_0.

- $c_i = \gamma^{i+1} c_0$ for some initial $c_0 > 0$ gives a step-size with a component that is geometrically increasing as in Xiao (2022), though at a slower rate than the one discussed in Section 8.4. The resulting bound is

$$\|V^\star - V^k\|_\infty \leqslant \gamma^k \Big(\|V^\star - V^0\|_\infty + k c_0 \Big),$$

which converges linearly with the sought-for γ-rate, though in early iterations the k factor may dominate.

- $c_i = \gamma^{2(i+1)} c_0$ for some $c_0 > 0$ gives a step-size with a geometrically increasing component. The resulting bound is

$$\|V^\star - V^k\|_\infty \leqslant \gamma^k \Big(\|V^\star - V^0\|_\infty + \frac{c_0}{1-\gamma} \Big),$$

which converges linearly with the γ-rate, and matches the bounds of PI and VI as c_0 goes to 0.

PMD cannot do better than the last bound as we will show in Theorem 8.5.1. Some simulations that demonstrate the behaviour of our step-size (8.8) and validate its benefits compared to the step-size of (Xiao, 2022) are given in Section 8.10.1.

A key component in establishing the γ-rate is avoiding a direct application of the performance difference lemma on $V^\star(\rho) - V^k(\rho)$, as was done in prior work. In particular, Xiao (2022) uses the lemma on $\mathbb{E}_{s \sim d_\rho^\star}[\langle Q_s^k, \pi_s^\star - \pi_s^{k+1} \rangle]$, which introduces a distribution mismatch coefficient in order to get a recursion. On the other hand, we extract the value sub-optimalities $V^\star(s) - V^k(s)$ and $\|V^\star - V^{k+1}\|_\infty$ directly from $\langle Q_s^k, \pi_s^\star - \pi_s^{k+1} \rangle$ in (8.11). This allows us to remove distribution mismatch factors and obtain an exact γ-rate.

Proof of Theorem 8.4.1. Fix $s \in \mathcal{S}$ and $k \geqslant 0$. From Lemma 8.3.3, we have that $\langle Q_s^k, \pi_s^{k+1} \rangle \leqslant \langle Q_s^{k+1}, \pi_s^{k+1} \rangle = V^{k+1}(s)$. This decouples the dependencies on π^k and π^{k+1} below and is one of the ingredients that allows us to bypass the performance difference lemma. Using this,

$$\begin{aligned} \langle Q_s^k, \pi_s^\star - \pi_s^{k+1} \rangle \geqslant \langle Q_s^k, \pi_s^\star \rangle - V^{k+1}(s) &= \langle Q_s^k - Q_s^\star, \pi_s^\star \rangle + \langle Q_s^\star, \pi_s^\star \rangle - V^{k+1}(s) \\ &\geqslant -\|Q_s^\star - Q_s^k\|_\infty + V^\star(s) - V^{k+1}(s), \end{aligned} \tag{8.10}$$

where the last step uses Hölder's inequality. Now we use that the difference in state-action values of different policies for the same state-action pair propagates the error to the next time-step, which is discounted by a factor of γ. Formally, for any state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\begin{aligned} Q^\star(s,a) - Q^k(s,a) &= \gamma \sum_{s'} p(s'|s,a)(V^\star(s') - V^k(s')) \\ &\leqslant \gamma \sum_{s'} p(s'|s,a)\|V^\star - V^k\|_\infty = \gamma \|V^\star - V^k\|_\infty, \end{aligned}$$

which is the same phenomenon that is responsible for the contraction of the Bellman operator. This gives $\|Q_s^\star - Q_s^k\|_\infty \leqslant \gamma \|V^\star - V^k\|_\infty$. Plugging into Equation (8.10),

$$V^\star(s) - V^{k+1}(s) - \gamma \|V^\star - V^k\|_\infty \leqslant \langle Q_s^k, \pi_s^\star - \pi_s^{k+1} \rangle.$$

The rest of the proof relies on making the right-hand side of the above arbitrarily small by taking a large enough step size. Choose any greedy policy with respect to Q_s^k, $\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}$,

$$\begin{aligned} V^\star(s) - V^{k+1}(s) - \gamma \|V^\star - V^k\|_\infty &\leqslant \langle Q_s^k, \pi_s^\star - \pi_s^{k+1} \rangle & (8.11) \\ &\leqslant \langle Q_s^k, \widetilde{\pi}_s^{k+1} - \pi_s^{k+1} \rangle & (8.12) \end{aligned}$$

where we use that $\widetilde{\pi}_s^{k+1}$ is greedy with respect to Q_s^k. We then apply the Three-Point Descent Lemma (Lemma 8.3.2) or (8.7) to $p = \widetilde{\pi}_s^{k+1}$,

$$\langle Q_s^k, \widetilde{\pi}_s^{k+1} - \pi_s^{k+1} \rangle \leqslant \frac{D(\widetilde{\pi}_s^{k+1}, \pi_s^k) - D(\widetilde{\pi}_s^{k+1}, \pi_s^{k+1}) - D(\pi_s^{k+1}, \pi_s^k)}{\eta_k} \leqslant D(\widetilde{\pi}_s^{k+1}, \pi_s^k)/\eta_k.$$

Combining with (8.12) and noting that this holds for any $\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}$, we have

$$V^\star(s) - V^{k+1}(s) - \gamma \|V^\star - V^k\|_\infty \leqslant \frac{1}{\eta_k} \min_{\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}} D(\widetilde{\pi}_s^{k+1}, \pi_s^k) \leqslant c_k$$

from the step-size condition in the statement of the theorem. Rearranging and recalling that s and k were arbitrary, we can choose s where $V^\star(s) - V^{k+1}(s)$ reaches its maximum value. We get

$$\|V^\star - V^{k+1}\|_\infty \leqslant \gamma \|V^\star - V^k\|_\infty + c_k,$$

and unravelling this recursion completes the proof. □

Comparison to Xiao (2022): Linear convergence of unregularised PMD was first established by Xiao (2022) under a geometrically increasing step-size $\eta_k = \eta_0/\gamma^k$. We discuss this step-size further and show the necessity of adaptivity to achieve the γ-rate in Section 8.6. Their rate of convergence is $1 - \frac{1}{\theta_\rho}$ where θ_ρ is an instance-dependent term defined as follows

$$\theta_\rho = \frac{1}{1-\gamma}\left\|\frac{d_\rho^\star}{\rho}\right\|_\infty,$$

where $d_\rho^\star$ is the visitation distribution defined in (8.1) under an optimal policy and ρ is the starting-state distribution to which the bound applies, i.e the bound is on $V^\star(\rho) - V^k(\rho)$. This θ_ρ is at best γ when we use ρ to be the stationary distribution of the optimal policy. However, in this case, the guarantee only applies to states on the support of this stationary distribution and provides no guarantees for other states. In general, it is unclear how θ_ρ may scale in a specific MDP. In particular, it is possible to construct an MDP where θ_ρ scales linearly with the size of the state space $|\mathcal{S}|$ (see Section 8.10.3). Though this MDP is somewhat trivial, it nonetheless illustrates how θ_ρ can easily be large leading to slow rates of convergence. It is also not straightforward to obtain convergence in individual states from the bound in Xiao (2022) due to the presence of ρ in the denominator of the mismatch coefficient in θ_ρ. In contrast, we obtain the optimal γ-rate of convergence and our result holds in ℓ_∞-norm over all states so avoids having to deal with a starting-state distribution ρ altogether.

This distribution mismatch coefficient commonly appears in convergence bounds in the literature (Kakade and Langford, 2002a; Scherrer, 2014a; Bhandari and Russo, 2019; Shani et al., 2020; Agarwal et al., 2021), both under exact and inexact policy evaluation. For many of these papers mentioned, removing it would be of great interest though often does not appear possible. Our results show that it is removable for the general family of PMD algorithms to obtain dimension-free linear convergence.

Comparison to Khodadadian et al. (2021): The γ-rate was established by Khodadadian et al. (2021) for NPG, a specific instance of PMD for which the Bregman Divergence is the KL-divergence. The bound shown in their work is similar to the one implied by our result with $c_i = \gamma^{2(i+1)}c_0$. Defining $\Delta^k(s) = \max_{a\in\mathcal{A}} Q^k(s,a) - \max_{a\notin\mathcal{A}_s^k} Q^k(s,a)$, the minimal sub-optimality gap in state s under π^k, then the step-size corresponding to their bound with the KL as Bregman

Divergence is

$$\eta_k \geqslant \max_{s, \widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}} \left\{ \left(L_k + \log |\mathcal{A}| + D(\widetilde{\pi}_s^{k+1}, \pi_s^k) \right) \frac{1}{\Delta_k(s)} \right) \right\},$$

where $L_k = Lk$ for some constant $L > 0$. This highlights the connection with our step-size condition (8.8). In particular, they both have an adaptive component that depends linearly on the Bregman divergence between the current policy and the greedy policy and a non-adaptive component on which the bound depends. An important difference is that our step-size is independent of the sub-optimality gap $\Delta_k(s)$, and will be robust to situations where this gap is small. We can construct a general family of MDPs for which we can make $\Delta_k(s)$ arbitrarily small and the step-size of Khodadadian et al. (2021) will correspondingly become arbitrarily large (see Section 8.10.4). Despite the apparent similarities with our results, their analysis is significantly different to ours as it exploits the specific closed-form update of NPG to bound the difference in value with an update of PI. Our analysis applies to PMD for a general mirror map (not just NPG) and as such does not utilize specific properties of the mirror map and does not require the analytic solution of the update to be known. Our analysis also easily extends to inexact PMD (see Section 8.7), which theirs does not.

Computing the step-size: The expression (8.8) for the step-size can be simplified. The minimum over $\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}$ can be removed by taking any $\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}$. It was stated with the minimum to have the smallest condition but that is not necessary. In fact, there will often only be one greedy action, i.e. $\mathcal{A}_s^k$ will contain a single action and there will just be one policy in $\widetilde{\Pi}_s^{k+1}$. As for the maximum over $s \in \mathcal{S}$, this condition is imposed because we are using the same step-size in all states for simplicity. If we allow different step-sizes in each state, then the step-size in state s, denoted $\eta_k(s)$ would just have to satisfy (choosing any $\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}$)

$$\eta_k(s) \geqslant \frac{1}{c_k} D_h(\widetilde{\pi}_s^{k+1}, \pi_s^k).$$

The computational complexity of the step-size is then that of the Bregman divergence between two policies. Note that the step-size (8.8) is always finite. It can be unbounded when h is of Legendre type on the relative interior of $\Delta(\mathcal{A})$ (as for the negative entropy) but then all iterates will belong to the relative interior of $\Delta(\mathcal{A})$ and the Bregman divergences are well defined so the step-size is finite.

8.5 Optimality of PMD

We have established in Theorem 8.4.1 that PMD achieves a linear γ-rate. The following result shows that this rate is in fact optimal in a worst-case sense.

Theorem 8.5.1. *Fix $n > 0$. There exists a class of MDPs with $|\mathcal{S}| = 2n + 1$ and a policy $\pi^0 \in$ rint Π such that running iterative updates of (8.4) for any positive step-size regime, we have for $k < n$:*

$$\|V^\star - V^k\|_\infty \geqslant \frac{1}{2}\gamma^k \|V^\star - V^0\|_\infty. \tag{8.13}$$

A key feature of this result is that the bound holds for $k < n$. For a fixed iteration budget, Theorem 8.5.1 implies that there exists an MDP on which PMD will not do better than the linear γ-rate for any step-size. The γ-rate for PMD that we prove in Theorem 8.4.1 is optimal in this sense.

Proof. **Proof outline.** The proof is based on an MDP in which PMD can learn the optimal action in at most one new state per iteration, starting from the left-most state of a chain of states. The argument has three main steps. First, we define a carefully designed MDP (Figure 8.1) and an initial policy that places most of its mass on a suboptimal action. Second, we show by induction that, for the first k iterations, PMD cannot increase the probability of the optimal action in states s_i with $i > k$. Finally, we use this delayed policy improvement to lower bound the suboptimality of the value function after k iterations and relate it to the initial error, yielding the γ^k lower bound.

Fix $n > 0$ and $\delta \in (0, (1-\gamma)\gamma^n)$. Consider the MDP shown in Figure 8.1. The state space is $\mathcal{S} = \{s_0, s_1, s_1', ..., s_n, s_n'\}$ and the action space is $\mathcal{A} = \{a_1, a_2\}$. There is a chain of states of length $n + 1$ with the states indexed from 0 to n. The left-most state (s_0) is absorbing with reward +1. In the other states in the chain (s_i for $i = 1, ..., n$), the agent can take action a_1 and move left (to s_{i-1}) with reward of 0, or take action a_2 and move to an additional absorbing state unique to the state it is currently in (s_i') with reward $r_i = \gamma^{i+1} + \delta$ (that the agent also receives in that state for all future time-steps). Summarising, we have for $1 \leqslant i \leqslant n$

$$\begin{aligned} &p(s_{i-1}|s_i, a_1) = 1, \quad r(s_i, a_1) = 0, \\ &p(s_i'|s_i, a_2) = 1, \quad r(s_i, a_2) = r_i = \gamma^{i+1} + \delta, \end{aligned}$$

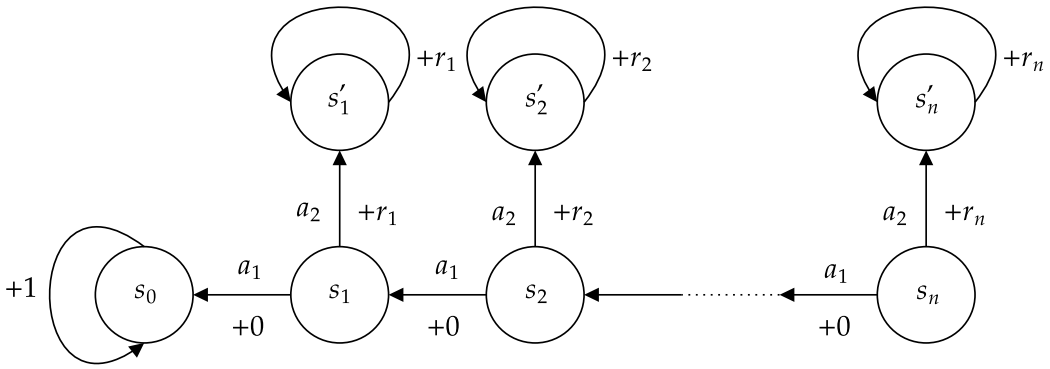

Figure 8.1: MDP used in the proof of Theorem 8.5.1

$$p(s'_i|s'_i, a) = 1, \quad r(s'_i, a) = r_i = \gamma^{i+1} + \delta \quad \forall a \in \mathcal{A}.$$

The value of δ is carefully restricted so that the optimal action in all the states of the chain is a_1. The proof consists in showing that if the agent starts with an initial policy that places most probability mass on the sub-optimal action a_2, then it has to learn that a_1 is the optimal action in the state directly to the left before it can start switching from action a_2 to a_1 in the current state. And this can at best happen one iteration at a time starting from the left-most state. In particular, we consider π^0 s.t $\pi^0(a_1|s) = \alpha$, $\pi^0(a_2|s) = 1 - \alpha$ for all states and some α s.t $0 < \alpha \leqslant \delta(1-\gamma)$. We make the following claim from which the result will follow straightforwardly.

Key idea. The crucial observation is that, due to the structure of the MDP, learning that action a_1 is optimal in state s_i requires first learning that it is optimal in s_{i-1}. Since PMD updates are monotone in the action-values, this implies that the probability mass on a_1 can only increase one state to the right per iteration. We formalise this idea in the following claim.

Claim. Fix $k < n$. The policies produced by PMD satisfy $\pi^k(a_1|s_i) \leqslant \alpha$ for $k < i \leqslant n$.

Induction strategy. We formalise the above intuition by proving that, after k iterations, the policy has not increased the probability of the optimal action a_1 beyond its initial value α in any state s_i with $i > k$. The proof is by induction on the iteration index.

Base Case ($k = 0 \to 1$). We show that after one iteration, PMD cannot increase the probability

of action a_1 in any state s_i with $i > 1$. This follows from showing that a_2 has larger action-value than a_1 in those states under the initial policy.

Formally, we want to show that $\pi^1(a_1|s_i) \leqslant \alpha$ for $i > 1$. We do this by showing that $Q^0(s_i, a_1) \leqslant Q^0(s_i, a_2)$ for $i > 1$ so that the probability of $\pi^1(a_1|s_i)$ cannot increase w.r.t $\pi^0(a_1|s_i)$, which is α (this follows from $\langle Q_s^k, \pi_s^{k+1} - \pi_s^k \rangle \geqslant 0$ for all iterations of PMD). We have:

$$
\begin{aligned}
Q^0(s_i, a_1) &= \gamma V^0(s_{i-1}) \\
&= \gamma\Big(\alpha Q^0(s_{i-1}, a_1) + (1-\alpha) Q^0(s_{i-1}, a_2)\Big) \\
&\leqslant \gamma\Big(\alpha \frac{\gamma^{i-1}}{1-\gamma} + \frac{r_{i-1}}{1-\gamma}\Big) \\
&\overset{(a)}{\leqslant} \gamma\Big(\delta(1-\gamma)\frac{\gamma^{i-1}}{1-\gamma} + \frac{\gamma^i + \delta}{1-\gamma}\Big) \\
&= \frac{\gamma^{i+1}}{1-\gamma} + \frac{\delta\gamma(1 + \gamma^{i-1} - \gamma^i)}{1-\gamma} \\
&\overset{(b)}{\leqslant} \frac{\gamma^{i+1}}{1-\gamma} + \frac{\delta}{1-\gamma} \\
&= Q^0(s_i, a_2),
\end{aligned}
$$

where we used $\alpha \leqslant \delta(1-\gamma)$ in (a) and $\gamma(1 + \gamma^{i-1} - \gamma^i) < 1$ for $\gamma \in [0, 1)$ in (b). This concludes the base case.

Inductive Step. Assuming that PMD has not increased the probability of a_1 beyond α in states s_i with $i > k$, we show that the same holds after iteration $k+1$ for all states s_i with $i > k+1$. The argument mirrors the base case, with the inductive hypothesis controlling the contribution of $Q^k(s_{i-1}, a_1)$: we show that $Q^k(s_i, a_1) \leqslant Q^k(s_i, a_2)$ for $i > k+1$ so that the probability of $\pi^{k+1}(a_1|s_i)$ cannot increase w.r.t $\pi^k(a_1|s_i)$, which is less than or equal to α by the inductive hypothesis. We have:

$$
\begin{aligned}
Q^k(s_i, a_1) &= \gamma V^k(s_{i-1}) \\
&= \gamma\Big(\pi^k(a_1|s_{i-1}) Q^k(s_{i-1}, a_1) + \pi^k(a_2|s_{i-1}) Q^k(s_{i-1}, a_2)\Big) \\
&\overset{(a)}{\leqslant} \gamma\Big(\alpha Q^k(s_{i-1}, a_1) + Q^k(s_{i-1}, a_2)\Big) \\
&\leqslant \gamma\Big(\alpha \frac{\gamma^{i-1}}{1-\gamma} + \frac{r_{i-1}}{1-\gamma}\Big) \\
&\overset{(b)}{\leqslant} \gamma\Big(\delta(1-\gamma)\frac{\gamma^{i-1}}{1-\gamma} + \frac{\gamma^i + \delta}{1-\gamma}\Big)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\gamma^{i+1}}{1-\gamma} + \frac{\delta\gamma(1+\gamma^{i-1}-\gamma^i)}{1-\gamma} \\
&\overset{(c)}{\leqslant} \frac{\gamma^{i+1}}{1-\gamma} + \frac{\delta}{1-\gamma} \\
&= Q^k(s_i, a_2),
\end{aligned}
$$

where we used in (a) that $\pi^k(a_1|s_{i-1}) \leqslant \alpha$ for $i > k+1$, which is true by the inductive hypothesis since $i-1 > k$, in (b) that $\alpha \leqslant \delta(1-\gamma)$ and in (c) that $\gamma(1+\gamma^{i-1}-\gamma^i) < 1$ for $\gamma \in [0,1)$. This concludes the proof of the claim.

From policy stagnation to value suboptimality. Having established that PMD cannot increase the probability of the optimal action beyond state s_k after k iterations, we now translate this delayed policy improvement into a lower bound on the value suboptimality at state s_{k+1}.

$$
\begin{aligned}
V^k(s_{k+1}) &= \pi^k(a_1|s_{k+1})Q^k(s_{k+1}, a_1) + \pi^k(a_2|s_{k+1})Q^k(s_{k+1}, a_2) \\
&\leqslant \alpha\frac{\gamma^{k+1}}{1-\gamma} + \frac{r_{k+1}}{1-\gamma} \\
&= \alpha\frac{\gamma^{k+1}}{1-\gamma} + \frac{\gamma^{k+2}+\delta}{1-\gamma},
\end{aligned}
$$

so

$$
\begin{aligned}
V^\star(s_{k+1}) - V^k(s_{k+1}) &\geqslant \frac{\gamma^{k+1}}{1-\gamma} - \alpha\frac{\gamma^{k+1}}{1-\gamma} - \frac{\gamma^{k+2}+\delta}{1-\gamma} \\
&= \frac{\gamma^{k+1}(1-\gamma)}{1-\gamma} - \frac{\alpha\gamma^{k+1}+\delta}{1-\gamma} \\
&\geqslant \gamma^{k+1} - \frac{\alpha+\delta}{1-\gamma} \\
&\geqslant \gamma^{k+1} - \frac{2\delta}{1-\gamma}, \qquad (8.14)
\end{aligned}
$$

where we used that $\alpha \leqslant \delta$.

Normalising by the initial error. To express the lower bound relative to the initial suboptimality, we next upper bound $\|V^\star - V^0\|_\infty$. This allows us to relate the error at iteration k directly to the starting error and obtain a clean γ^k dependence. We have

$$
\begin{aligned}
V^0(s_1) &= \alpha Q^0(s_1, a_1) + (1-\alpha)Q^0(s_1, a_2) \\
&= \alpha\frac{\gamma}{1-\gamma} + (1-\alpha)\frac{\gamma^2+\delta}{1-\gamma},
\end{aligned}
$$

so

$$\begin{aligned}
V^\star(s_1) - V^0(s_1) &= \frac{\gamma}{1-\gamma} - \alpha\frac{\gamma}{1-\gamma} - (1-\alpha)\frac{\gamma^2+\delta}{1-\gamma} \\
&= (1-\alpha)\frac{\gamma}{1-\gamma} - (1-\alpha)\frac{\gamma^2+\delta}{1-\gamma} \\
&= \frac{1-\alpha}{1-\gamma}\Big(\gamma - \gamma^2 - \delta\Big) \\
&\leqslant \frac{1-\alpha}{1-\gamma}\Big(\gamma - \gamma^2\Big) \\
&= \gamma\frac{1-\alpha}{1-\gamma}\Big(1-\gamma\Big) \\
&= \gamma(1-\alpha) \\
&\leqslant \gamma/
\end{aligned}$$

By induction, the same bound holds for all states. Above is the base case. The inductive step is as follows (assuming $V^\star(s_k) - V^0(s_k) \leqslant \gamma$),

$$\begin{aligned}
V^\star(s_{k+1}) - V^0(s_{k+1}) &= \frac{\gamma^{k+1}}{1-\gamma} - (1-\alpha)\frac{\gamma^{k+2}+\delta}{1-\gamma} - \alpha\gamma V^0(s_k) \\
&= (1-\alpha)\Big[\frac{\gamma^{k+1} - \gamma^{k+2} - \delta}{1-\gamma}\Big] + \alpha\gamma\Big[V^\star(s_k) - V^0(s_k)\Big] \\
&\leqslant (1-\alpha)\gamma^{k+1} + \alpha\gamma^2 \\
&\leqslant \gamma
\end{aligned}$$

and so

$$\|V^\star - V^0\|_\infty \leqslant \gamma,$$

which combining with (8.14) gives,

$$\begin{aligned}
V^\star(s_{k+1}) - V^k(s_{k+1}) &\geqslant \gamma^k\|V^\star - V^0\|_\infty - \frac{2\delta}{1-\gamma} \\
\implies \|V^\star - V^k\|_\infty &\geqslant \gamma^k\|V^\star - V^0\|_\infty - \frac{2\delta}{1-\gamma}.
\end{aligned}$$

Choice of parameters. Finally, we choose δ small enough to satisfy the constraints imposed by optimality of action a_1 and to ensure that the residual term does not dominate the γ^k contribution. Choosing $\delta = \frac{(1-\gamma)}{4}\gamma^n\|V^\star - V^0\|_\infty \leqslant \frac{(1-\gamma)}{4}\gamma^{n+1}$ (which is in the valid range of δ) concludes the proof since $\frac{2\delta}{1-\gamma} \leqslant \frac{1}{2}\gamma^k\|V^\star - V^0\|_\infty$. □

Lower-bound beyond $k < n$? For a fixed state-space size $|\mathcal{S}| = 2n + 1$, it is not known what lower-bound holds when $k \geqslant n$. However, PI is an instance of PMD (see Section 8.2) and thus a lower bound that scales with γ^k for all $k > 0$ cannot hold since PI converges exactly in finite-iterations (in fact with a number of iterations that scales linearly with $|\mathcal{S}|$ (Scherrer, 2013, Theorem 3)). However, an asymptotic lower-bound for PMD under arbitrary step-size will not hold even if we exclude PI from the class of considered algorithms. As the step-size tends to infinity, any PMD update recovers a PI update. This implies that general PMD can be arbitrarily close to PI's exact convergence for the same finite number of iterations. Thus, any lower-bound on the convergence of PMD must be limited to finite iterations. Since the finite iteration convergence of PI only scales linearly with $|\mathcal{S}|$ (Scherrer, 2013, Theorem 3), the number of iterations guaranteed by Theorem 8.5.1 has the same dependence on $|\mathcal{S}|$ as the number of iterations needed for exact convergence of PI. Furthermore, this restriction is mild for the high-dimensional problems we consider in this thesis where $|\mathcal{S}|$ and n are large.

To the best of our knowledge, this lower bound on the value convergence of PMD scaling with γ^k is new. We expect this result may have been known for the special case of PI, though we could not find a proof of it in the literature. The works that establish a lower bound for PI do so in the setting of exact convergence to the optimal policy (Hollanders et al., 2012), not ε-accurate convergence, and for undiscounted MDPs (Fearnley, 2010).

Rate-Optimality Theorem 8.5.1 establishes that the γ-rate is optimal in the first $|\mathcal{S}|/2$ iterations. This non-asymptotic optimality is justified by the discussion above, which highlights that a rate for PMD under arbitrary step-size can only be optimal for a finite number of iterations, and specifically up until the exact convergence of PI.

There are some results on the super-linear convergence of NPG in the literature, though these apply once you have a policy within some neighbourhood of the optimal policy or value. Cen et al. (2021) establish such a result for NPG in the regularised case, and Khodadadian et al. (2021) in the unregularised case under additional conditions. Theorem 8.5.1 does not contradict this latter result as for the MDP considered in the proof, the super-linear convergence would kick-in for iterations beyond the $k < n$ considered here.

Lower-bound for general PG methods: The lower bound of Theorem 8.5.1 in fact applies to any algorithm that at each iteration increases the probability of the current greedy action.

The greedy action is the action with highest action-value for the current policy. This covers algorithms more general than PMD and in particular, includes the vanilla PG algorithm.

8.6 Adaptive Step-Size Necessity

We have established in Theorem 8.4.1 that PMD under an adaptive step-size achieves a linear γ-rate and in Theorem 8.5.1 that this rate is optimal for PMD. We now show the adaptivity is in fact necessary to achieve the γ-rate. This strengthens the notion of optimality from the previous section - both the rate and step-size are unimprovable. The proof uses the same MDP construction as in the proof of Theorem 8.5.1 and can be found in Section 8.10.2.

Theorem 8.6.1. *Fix $n > 0$ and $\gamma > 0.2$. There exists an MDP with state-space of size $|\mathcal{S}| = 2n + 1$ and a policy $\pi^0 \in$ rint Π such that running iterative updates of NPG (PMD with h as the negative entropy) that satisfy $\|V^\star - V^k\|_\infty \leqslant \gamma^k(\|V^\star - V^0\|_\infty + \frac{1-\gamma}{8})$ requires*

$$\eta_{k_i} \geqslant KL(\tilde{\pi}_{s_i}^{k_i+1}, \pi_{s_i}^{k_i})/2\gamma^{k_i}, \qquad \tilde{\pi}_{s_i}^{k_i+1} \in \widetilde{\Pi}_s^{k+1} \tag{8.15}$$

for $i = 1, ..., n$ s.t $k_1 < k_2 < ... < k_n$ where $\{s_1, ..., s_n\}$ are distinct states of the considered MDP.

This theorem states that there are at least n distinct iterations with n distinct states where the step-size has to be bigger than a quantity depending on the Bregman divergence between the current policy and its greedy policy in the considered state in order to achieve a linear γ-rate. This is precisely the notion of adaptivity that appears in the step-size condition of Theorem 8.4.1 and Khodadadian et al. (2021). Theorem 8.6.1 shows we cannot improve on this in general and provides justification for using an adaptive step-size instead of the one from Xiao (2022). Beyond its necessity, the adaptivity of our step-size can be a strength: it is large when needed, small when not. This is illustrated by the simulations in Section 8.10.1.

Theorem 8.6.1 only holds for the particular case of the negative entropy (NPG). This limitation is due to technical reasons rather than fundamental ones. The proof relies on the closed form update of NPG and we don't have this for general mirror map. However, this result for NPG allows us to argue the necessity of the step-size for PMD in general.

8.7 Sample Complexity of Inexact PMD under a Generative Model

In this section, we study the sample complexity of PMD when the exact action-values are not available and must be estimated from data using a generative model. We proceed in three steps. First, we describe the inexact PMD (IPMD) update and the Monte-Carlo estimator used to approximate action-values. Second, we recall a concentration result controlling the accuracy of this estimator. Third, we show that PMD retains an approximate descent property under estimation error, which allows us to extend the exact convergence analysis to the inexact setting and derive a sample complexity bound.

In the previous sections, we have assumed access to the state-action values Q_s^k to carry out the PMD update. In Inexact PMD (IPMD), we replace Q_s^k with an estimate $\widehat{Q}_s^k$ giving the update

$$\pi_s^{k+1} = \operatorname{argmin}_{p\in\Delta(\mathcal{A})}\Big\{ -\eta_k\langle \widehat{Q}_s^k, p\rangle + D_h(p, \pi_s^k)\Big\}. \tag{8.16}$$

Similarly to the exact case, IPMD iteratively applies update (8.16) for some sequence of $\eta_k > 0$ and initial policy $\pi^0 \in \operatorname{rint}\Pi$, this time only assuming access to an inexact estimator of Q^k.

We consider the setting of a generative model (Kearns and Singh, 1998), in which one can sample transitions from $p(\cdot \mid s, a)$ for any state-action pair (s, a). This allows us to estimate action-values using Monte-Carlo rollouts. Following Xiao (2022); Lazaric et al. (2016), for each (s, a) and iteration k, we draw M_k independent trajectories of horizon H starting from (s, a) and following policy π^k. Using these trajectories, we define the truncated Monte-Carlo estimator as follows. For each state-action pair (s, a) and iteration k, let

$$\big\{(s_t^{(i)}, a_t^{(i)})\big\}_{t=0}^{H-1}, \qquad i = 1, \dots, M_k,$$

denote M_k independent trajectories generated by starting from $(s_0^{(i)}, a_0^{(i)}) = (s, a)$, sampling actions according to $\pi^k(\cdot \mid s_t^{(i)})$, and sampling next states according to the transition kernel $p(\cdot \mid s_t^{(i)}, a_t^{(i)})$. The truncated Monte-Carlo estimator of the action-value function is then defined as

$$\widehat{Q}^k(s, a) \;=\; \frac{1}{M_k}\sum_{i=1}^{M_k}\sum_{t=0}^{H-1}\gamma^t r\big(s_t^{(i)}, a_t^{(i)}\big). \tag{8.17}$$

We use these $\widehat{Q}^k(s,a)$ to replace $Q^k(s,a)$ in the PMD update step.

The first ingredient in the analysis is a uniform concentration bound on the Monte-Carlo estimator $\widehat{Q}^k$. The following lemma from Xiao (2022) quantifies how the estimation error depends on the rollout horizon H and the number of samples M_k.

Lemma 8.7.1 (Lemma 15 in Xiao (2022))**.** *Consider using (8.17) to estimate Q_s^k for all state-action pairs for K iterations of IPMD. Then for $\delta \in (0,1)$, if for all $k \leqslant K$,*

$$M_k \geqslant \frac{\gamma^{-2H}}{2} \log\big(\frac{2K|\mathcal{S}||\mathcal{A}|}{\delta}\big).$$

Then with probability at least $1-\delta$, we have for all $k \leqslant K$,

$$\|\widehat{Q}_s^k - Q_s^k\|_\infty \leqslant \frac{2\gamma^H}{1-\gamma}.$$

While Lemma 8.7.1 controls the estimation error, convergence of IPMD requires understanding how this error affects the monotonic improvement property of PMD. The following lemma extends the descent property of PMD (Lemma 8.3.3) to the inexact setting by explicitly quantifying the impact of action-value estimation error.

Lemma 8.7.2 (Descent Property of Inexact PMD, Lemma 11 in Xiao (2022))**.** *Consider the policies produced by the iterative updates of IPMD in (8.16). For any $k \geqslant 0$, if $\|\widehat{Q}_s^k - Q_s^k\|_\infty \leqslant \tau$, then*

$$\widehat{Q}^{k+1}(s,a) \geqslant \widehat{Q}^k(s,a) - \frac{2\tau\gamma}{1-\gamma}, \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}.$$

Proof. The argument follows the proof of the exact descent property, with an additional error term arising from the difference between $\widehat{Q}^k$ and Q^k. We control this term using the assumed uniform bound on the estimation error.

The Three-Point Descent Lemma (Lemma 8.3.2) applied to the IPMD update (8.16) gives $\forall p \in \Delta(\mathcal{A})$,

$$-\eta_k\langle \widehat{Q}_s^k, \pi_s^{k+1}\rangle + D_h(\pi_s^{k+1}, \pi_s^k) \leqslant -\eta_k\langle \widehat{Q}_s^k, p\rangle + D_h(p, \pi_s^k) - D_h(p, \pi_s^{k+1}).$$

Using this with $p = \pi_s^k$,

$$D_h(\pi_s^k, \pi_s^{k+1}) + D_h(\pi_s^{k+1}, \pi_s^k) \leqslant \eta_k\langle \widehat{Q}_s^k, \pi_s^{k+1} - \pi_s^k\rangle$$

and since the Bregman divergences are none-negative and $\eta_k > 0$,

$$0 \leqslant \langle \widehat{Q}_s^k, \pi_s^{k+1} - \pi_s^k \rangle.$$

We cannot use the above inequality directly with the performance difference lemma (Lemma 8.3.1) since $\widehat{Q}_s^k$ is not the true action-value. Instead, we have for any $s' \in \mathcal{S}$

$$\begin{aligned}
V^{k+1}(s') - V^k(s') &= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s'}^{k+1}} \Big[\langle Q_s^k, \pi_s^{k+1} - \pi_s^k \rangle \Big] \\
&= \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s'}^{k+1}} \Big[\langle Q_s^k - \widehat{Q}_s^k, \pi_s^{k+1} - \pi_s^k \rangle + \langle \widehat{Q}_s^k, \pi_s^{k+1} - \pi_s^k \rangle \Big] \\
&\geqslant \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s'}^{k+1}} \Big[- \|Q_s^k - \widehat{Q}_s^k\|_\infty \|\pi_s^{k+1} - \pi_s^k\|_1 \Big] \\
&\geqslant \frac{1}{1-\gamma} \mathbb{E}_{s \sim d_{s'}^{k+1}} \Big[- 2\tau \Big] \\
&= -\frac{2\tau}{1-\gamma}
\end{aligned}$$

Using this, we immediately have

$$Q^{k+1}(s,a) - Q^k(s,a) = \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a) \Big(V^{k+1}(s') - V^k(s') \Big) \geqslant \frac{-2\tau\gamma}{1-\gamma},$$

which concludes the proof. □

Following the same ideas as Xiao (2022), we can extend Theorem 8.4.1 to the inexact setting. The following theorem establishes a sample complexity result, which is the sufficient number of calls to the generative model to obtain an ε-optimal policy. For simplicity, we focus on the step-size following from the choice $c_i = \gamma^{2(i+1)}$.

Theorem 8.7.3. *Consider applying iterative updates of (8.16) using the Q-estimator in (8.17) given access to a generative model with $\pi^0 \in \operatorname{rint} \Pi$ and step-sizes satisfying for all $k \geqslant 0$ (with the definitions of $\mathcal{A}_s^k$ and $\widetilde{\Pi}_s^{k+1}$ suitably adjusted with $\widehat{Q}_s^k$ instead of Q_s^k),*

$$\eta_k \geqslant \max_{s \in \mathcal{S}} \Big\{ \min_{\widetilde{\pi}_s^{k+1} \in \widetilde{\Pi}_s^{k+1}} \frac{D(\widetilde{\pi}_s^{k+1}, \pi_s^k)}{\gamma^{2k+1}} \Big\}.$$

Fix $\varepsilon > 0$. For any $\delta \in (0,1)$, suppose the following are satisfied for all $k \geqslant 0$,

$$K > \frac{1}{1-\gamma} \log \frac{4}{(1-\gamma)\varepsilon}, \quad H \geqslant \frac{1}{1-\gamma} \log \frac{16}{(1-\gamma)^3 \varepsilon} \quad \text{and} \quad M_k = M \geqslant \frac{\gamma^{-2H}}{2} \log \frac{2K|\mathcal{S}||\mathcal{A}|}{\delta}.$$

Then with probability at least $1-\delta$, $\|V^\star - V^k\|_\infty \leqslant \gamma^k\Big(\|V^\star - V^0\|_\infty + \frac{1}{1-\gamma}\Big) + \frac{8}{(1-\gamma)^3}\gamma^H < \varepsilon$. *Choosing* K, H *and* M *to be tight to their lower-bounds, the corresponding sample complexity is* $\tilde{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^8\varepsilon^2}\Big)$, *where the notation* $\tilde{O}()$ *hides poly-logarithmic factors.*

Proof. This proof is similar to that of Xiao (2022) (Theorem 14). It is also similar in structure to the proof of Theorem 8.4.1.

Fix a state $s \in \mathcal{S}$ and an integer $k \geqslant 0$. We first assume that the action-value estimates are uniformly τ-accurate, i.e.,

$$\|Q^k - \widehat{Q}^k\|_\infty \leqslant \tau$$

for all $k \geqslant 0$. With this assumption, we have from the Descent Property of IPMD (Lemma 8.7.2),

$$Q^{k+1}(s,a) \geqslant Q^k(s,a) - \frac{2\gamma\tau}{1-\gamma}, \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A}.$$

Now proceeding in a similar way to the proof of Theorem 8.4.1,

$$\begin{aligned}
\langle \widehat{Q}_s^k, \pi_s^\star - \pi_s^{k+1}\rangle &= \langle Q_s^k, \pi_s^\star - \pi_s^{k+1}\rangle + \langle \widehat{Q}_s^k - Q_s^k, \pi_s^\star - \pi_s^{k+1}\rangle \\
&\geqslant \langle Q_s^k, \pi_s^\star\rangle - \langle Q_s^k, \pi_s^{k+1}\rangle - \|\widehat{Q}_s^k - Q_s^k\|_\infty \|\pi_s^\star - \pi_s^{k+1}\|_1 \\
&\geqslant \langle Q_s^k, \pi_s^\star\rangle - \langle Q_s^{k+1}, \pi_s^{k+1}\rangle - \frac{2\gamma\tau}{1-\gamma} - 2\tau \\
&\geqslant \langle Q_s^k, \pi_s^\star\rangle - V^{k+1}(s) - \frac{4\gamma\tau}{1-\gamma} \\
&= \langle Q_s^k - Q_s^\star, \pi_s^\star\rangle + V^\star(s) - V^{k+1}(s) - \frac{4\gamma\tau}{1-\gamma} \\
&\geqslant -\|Q_s^\star - Q_s^k\|_\infty + V^\star(s) - V^{k+1}(s) - \frac{4\gamma\tau}{1-\gamma} \\
&\geqslant -\gamma\|V^\star - V^k\|_\infty + V^\star(s) - V^{k+1}(s) - \frac{4\gamma\tau}{1-\gamma}.
\end{aligned}$$

Now again proceeding exactly as to the proof of Theorem 8.4.1 with this extra τ-term using the step-size condition ($c_k = \gamma^{2k+1}$), we end up with

$$\|V^\star - V^{k+1}\|_\infty \leqslant \gamma\|V^\star - V^k\|_\infty + \gamma^{2k+1} + \frac{4\gamma\tau}{1-\gamma}.$$

Unravelling this recursion yields

$$\begin{aligned}\|V^\star - V^k\|_\infty &\leqslant \gamma^k\Big(\|V^\star - V^0\|_\infty + \sum_{i=1}^{k}\gamma^{-i}\gamma^{2(i-1)+1}\Big) + \frac{4\gamma\tau}{1-\gamma}\sum_{i=0}^{k-1}\gamma^i \\ &\leqslant \gamma^k\Big(\|V^\star - V^0\|_\infty + \frac{1}{1-\gamma}\Big) + \frac{4\gamma\tau}{(1-\gamma)^2}.\end{aligned}$$

Now using the properties of the estimator (8.17) in Lemma 8.7.1, we have with probability $1-\delta$ for all $0 \leqslant k \leqslant K$,

$$\tau = \frac{2\gamma^H}{1-\gamma},$$

giving

$$\begin{aligned}\|V^\star - V^k\|_\infty &\leqslant \gamma^k\Big(\|V^\star - V^0\|_\infty + \frac{1}{1-\gamma}\Big) + \frac{8\gamma^H}{(1-\gamma)^3} \\ &\leqslant \frac{2}{1-\gamma}\gamma^k + \frac{8\gamma^H}{(1-\gamma)^3}.\end{aligned}$$

This establishes the first bound. Now

$$\begin{aligned}K &> \frac{1}{1-\gamma}\log\frac{4}{(1-\gamma)\varepsilon} \implies \frac{2}{1-\gamma}\gamma^K \leqslant \varepsilon/2, \\ H &\geqslant \frac{1}{1-\gamma}\log\frac{16}{(1-\gamma)^3\varepsilon} \implies \frac{8\gamma^H}{(1-\gamma)^3} \leqslant \varepsilon/2\end{aligned}$$

giving

$$\|V^\star - V^k\|_\infty \leqslant \varepsilon/2 + \varepsilon/2 = \varepsilon$$

as required. In terms of M, we have

$$\begin{aligned}M &\geqslant \frac{\gamma^{-2H}}{2}\log\frac{2K|\mathcal{S}||\mathcal{A}|}{\delta} \\ &\geqslant \frac{1}{2}\Big(\frac{16}{(1-\gamma)^3\varepsilon}\Big)^2\log\frac{2K|\mathcal{S}||\mathcal{A}|}{\delta} \\ &= \frac{16^2}{2(1-\gamma)^6\varepsilon^2}\log\frac{2K|\mathcal{S}||\mathcal{A}|}{\delta}\end{aligned}$$

and the corresponding number of calls to the sampling model, i.e. the sample complexity is (what we have shown above is actually a lower bound but can choose K, H, M so that it is of

the following order),

$$|\mathcal{S}| \cdot |\mathcal{A}| \cdot K \cdot H \cdot M = \tilde{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^8\varepsilon^2}\Big),$$

where the notation $\tilde{O}()$ hides poly-logarithmic factors. This completes the proof. □

The sample complexity established by Xiao (2022) (Theorem 16) under a generative model and the same Q-estimator is $\tilde{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^8\varepsilon^2}\Big\|\frac{d_\rho^\star}{\rho}\Big\|_\infty^3\Big)$. In their work, Xiao (2022) stresses the interest in reducing the dependence on both $1/(1-\gamma)$ and the distribution mismatch coefficient in order to scale the PMD guarantees to more relevant settings such as function approximation. Theorem 8.7.3 partially resolves this matter by removing the dependence on the distribution mismatch coefficient, which may scale with the size of the state space. This makes the result dimension-optimal, which is crucial when scaling the results to large or infinite state or action spaces. The dependence on $1/(1-\gamma)$ remains distant from the $1/(1-\gamma)^3$ lower-bound of Azar et al. (2013) (see Section 8.8). Whether this can be reached by specific instances of PMD remains open, though using a more suitable Q-estimator than (8.17) with our step size regime and analysis, which extends to arbitrary Q-estimators, could bring the sample complexity closer to this.

Beyond a generative model: Of more practical interest is a sample-complexity result where one does not have access to a generative model but rather must learn along a single-trajectory. This is however a much more difficult problem and requires assumptions on the irreducibility of the MDP. We have kept the focus on the generative model since even in this simpler setting, the sample-complexity results are still far from optimal.

8.8 Related Works

8.8.1 Convergence Rates for Exact PMD

We first consider the setting where exact policy evaluation is assumed. In this setting, several earlier works have sub-linear convergence results for PMD (Geist et al., 2019; Shani et al., 2020) and NPG specifically (Agarwal et al., 2021; Abbasi-Yadkori et al., 2019), though these have since been improved to linear convergence results as discussed below. Note that the work of (Abbasi-Yadkori et al., 2019) refers to their algorithm as Politex, which shares the same update as NPG.

A line of work has considered PG methods applied to regularised MDPs. In this setting, linear

convergence has been established for NPG with entropy regularisation (Cen et al., 2021), PMD with strongly-convex regularisers (Lan, 2021) and PMD with convex non-smooth regularisers (Zhan et al., 2023). The rates of convergence are either exactly γ or can be made arbitrarily close to γ by letting the step-size go to infinity.

In the setting of unregularised MDPs, which is the focus of this chapter, linear convergence of the special case of NPG was established (Bhandari and Russo, 2021; Khodadadian et al., 2021) under an adaptive step-size similar to ours that depends on the current policy at each step. The bound of Bhandari and Russo (2021) has an additive asymptotic error-floor that can be made arbitrarily small by making the step-size larger, while for a similar step-size Khodadadian et al. (2021) does not have this term. The analysis of Khodadadian et al. (2021) relies on a link between NPG and PI and consists of bounding the difference in value between iterates of both methods. Bhandari and Russo (2021) also establish linear convergence for a number of algorithms including PMD, although it is in the idealised setting of choosing the step size at each iteration that leads to the largest increase in value. This step-size choice will make PMD at least as good as PI since arbitrarily large step-sizes can be chosen and PMD with an infinite step-size converges to a PI update. Since PI converges linearly, so will PMD. This does not establish linear convergence of PMD for step-sizes with a closed-form expression.

Linear convergence for unregularised general PMD was recently established by Xiao (2022) under a geometrically increasing step-size. In general, their rate is instance-dependent and may scale with problem dependent quantities such as the size of the state space. For general starting-state distributions, this same instance-dependent rate was established by Li et al. (2023b) for a variant of PMD which augments the update with an added regularisation term. We focus our comparison on the work of Xiao (2022) rather than this work as the guarantees are equivalent in both but Li et al. (2023b) do not directly study PMD and have a more complicated algorithm.

A summary of our results compared to those of Khodadadian et al. (2021) and Xiao (2022) is presented in Table 8.1 and discussed in more detail in Section 8.4. In terms of optimality, Khodadadian et al. (2021) provide a lower-bound for constant step-size NPG, though it only applies to MDPs with a single-state, which can be solved in a single iteration with exact policy evaluation as the step-size goes to infinity (for which the lower-bound goes to 0). We provide a lower-bound in Theorem 8.5.1 that applies to PMD with arbitrary step-size on an MDP with any

finite state space. To the best of our knowledge, prior to the work in this chapter no lower-bound has been established in this general setting.

8.8.2 Sample Complexity of Inexact PMD

Sample complexity in the inexact policy evaluation setting refers to the number of samples needed to guarantee an ε-optimal policy is returned when we no longer have access to the Q-values exactly. We here give an outline of results, typically established in high-probability, under a generative model that we formally present in Section 8.7. The lower bound on the sample complexity in this setting was shown to be of $\tilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\right)$ by Azar et al. (2013). This lower-bound can be reached by model-based approches (Agarwal et al., 2020; Li et al., 2020) and model-free approaches (Sidford et al., 2018; Wainwright, 2019a).

The sample-complexity for PG methods has been recently studied in (Yuan et al., 2022b). Under a generative model, some works have considered PMD or NPG under various types of regularisation (Cen et al., 2021; Lan, 2021; Zhan et al., 2023). We focus on unregularised methods, for which results for PMD or its instances on tabular MDPs under a generative model are limited. There are works that obtain sample complexity results for NPG (Agarwal et al., 2021; Liu et al., 2020) and for PMD (Shani et al., 2020) though they do not attain the optimal ε-dependence of $\mathcal{O}(\varepsilon^{-2})$. Lazaric et al. (2016) show that a variant of PI, a special case of PMD, achieves the optimal ε-dependence of $\mathcal{O}(\varepsilon^{-2})$. More recently, Xiao (2022) show that the general family of PMD methods match the $\mathcal{O}(\varepsilon^{-2})$ sample complexity with a factor of $(1-\gamma)^{-8}$. Our result for the inexact setting shares the same dependence on ε and $1-\gamma$ as Xiao (2022) but removes an instance-dependent quantity which can depend on the size of the state space. Further comparison to the result in (Xiao, 2022) is given in Section 8.7. Beyond tabular MDPs and generative models, Chen and Theja Maguluri (2022) study NPG under linear function approximation and off-policy sampling, though their results imply worse sample complexities when restricted to tabular MDPs under a generative model. PMD under linear function approximation (Yuan et al., 2023; Alfano and Rebeschini, 2022a) and general function approximation (Alfano et al., 2023a) have also been studied and results similar to Xiao (2022) were obtained in those settings.

8.8.3 Recent Works

Since Johnson et al. (2023), several works have further explored PMD. Most of them have focused on extensions beyond the tabular setting to linear and more general function approximation

(Yuan et al., 2022a; Alfano and Rebeschini, 2022b; Alfano et al., 2023b; Xiong et al., 2024; Xu et al., 2024). Inexact PMD has been studied in the tabular setting but under weaker sampling models than the generative model, such as trajectory-based or temporal sampling (Li et al., 2025). Finally,Protopapas and Barakat (2024) study a lookahead variant of PMD that embeds multiple value-iteration steps within each update. This gives faster convergence in the exact setting and improves the sample complexity to $\mathcal{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^7\varepsilon^2}\Big)$ shaving off a factor of $1/(1-\gamma)$ compared to our sample complexity but still far from the optimal $\mathcal{O}\Big(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\varepsilon^2}\Big)$.

8.9 Conclusion

In this chapter, we have shown that the general family of exact policy mirror descent algorithms in tabular MDPs under an adaptive step-size achieve the same dimension-free linear γ-rate of convergence of classical algorithms such as policy iteration, improving the understanding of gradient-based methods in high-dimensional reinforcement learning.

We provide matching lower-bounds that establish this rate as optimal for PMD and the adaptive step-size as necessary. We exploit a new approach to study the convergence of PMD, for which avoiding direct use of the performance difference lemma is a key element. Though the focus of our work is on the exact policy evaluation setting, the analysis naturally extends to the inexact setting, given access to an estimator of the action-value of a policy. We provide a result for a simple estimator under a generative model that improves upon the best-known sample complexity, although it still does not match the lower bound. Our method is general and applies to any estimator, meaning our result could be improved by a better estimator. Exploiting further algorithmic properties of specific instances of PMD in the inexact setting may be needed to bridge the gap to the optimal sample complexity, and determine if PMD can match the lower bound in the inexact setting.

8.10 Missing Details

8.10.1 Simulations

In this Section, we present simulations (see Figures 1-4 below) made by running NPG (PMD with negative-entropy mirror map, see (Khodadadian et al., 2021)) on the hard MDP used to prove Theorem 8.5.1. We use $n = 25$ so $|\mathcal{S}| = 51$ and $|\mathcal{A}| = 2$. We use $\delta = (1-\gamma)\gamma^n/100$, $\gamma = 0.99$,

$\pi^0(a_1|s) = \alpha$ (note that a_1 is the optimal action in each state). In each of the plots, we compare the performance of NPG using our step-size (8.8) from Theorem 8.4.1 (denoted **Adaptive** in the plots and by ADA in the discussion below) with $c_k = \gamma^{2k}$ and $c_0 = 1$ and the step-size of (Xiao, 2022) $\eta_k = \eta_0/\gamma^k$ (denoted by INC in the discussion below) for a fixed $\eta_0 = 1$. In each of the plots, the left plot is $\|V^\star - V^k\|_\infty$ against iterations and the right plot is η_k against iterations. In all plots, the green curve represents the γ-rate. In particular it is the curve of $y = \gamma^x$.

Discussion With fixed initial step-size $\eta_0 = 1$ for INC, the convergence of NPG with INC can be made arbitrarily slow in early iterations by choosing α $(= \pi^0(a_1|s))$ closer to 0 (Figures 1-3). The convergence of NPG with ADA is unaffected by small values of α because it adapts to them.

These observations agree with the theoretical results we established. Our adaptive step-size (8.8) achieves the γ-rate (Theorem 8.4.1) and this is optimal for iterations k up until $n = 25$ (Theorem 8.5.1). The adaptivity in the step-size is necessary to achieve the γ-rate in these early iterations (Theorem 8.6.1).

In later iterations (Figure 1: iterations $\approx$ 100. Figure 2: iterations $\approx$ 200), INC has faster convergence than ADA. In these later iterations, acting greedily with respect to the action value of the current policy gives the optimal policy so INC converges faster because the step-size is larger (see Figure 1-2 right plot). This suggests combining ADA and INC by taking the maximum of both (Figure 4), yielding faster convergence than both. This is encouraging for future-work on super-linear convergence (as discussed in (Xiao, 2022)) beyond the optimal γ-rate valid up to a finite iteration (see Section 8.5).

8.10.2 Proof of Theorem 8.6.1

Overview. We prove the theorem by exhibiting an MDP on which achieving a γ-linear contraction of the value error at every iteration forces the NPG step-size to exceed a certain threshold at a sequence of distinct iterations. The construction relies on the same hard instance used in the proof of Theorem 8.5.1, where information about the optimal action propagates from left to right along a chain of states. The argument proceeds in three steps:

1. We first control the initial value error and show it is dominated by the error at the left-most state.

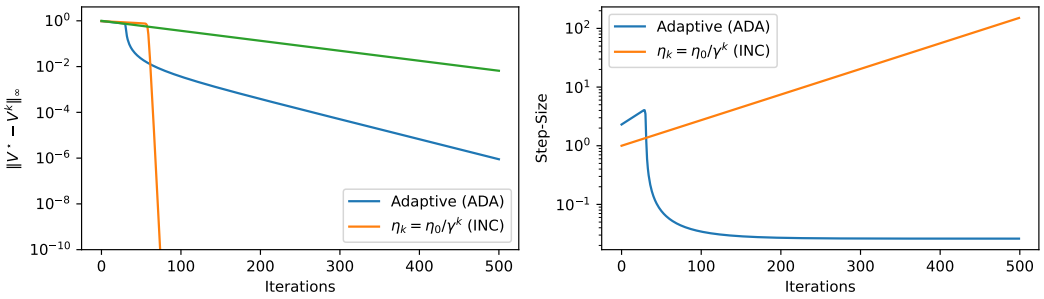

Figure 8.2: $\alpha = 10^{-1}$. Green curve: $y = \gamma^x$.

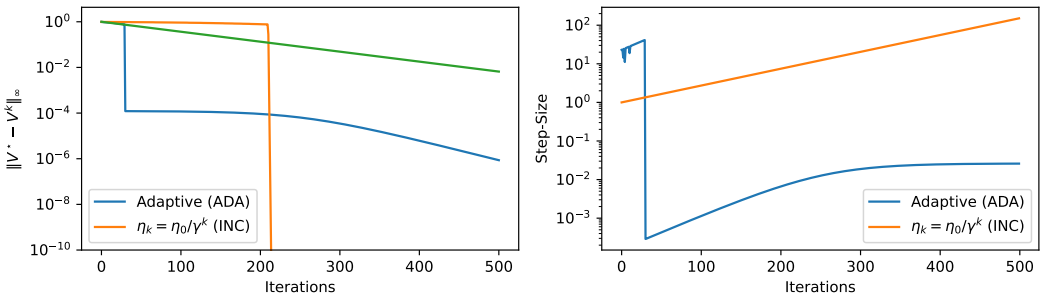

Figure 8.3: $\alpha = 10^{-10}$. Green curve: $y = \gamma^x$.

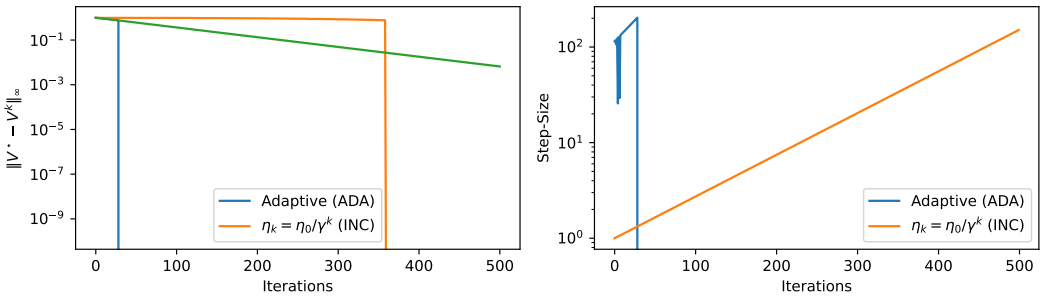

Figure 8.4: $\alpha = 10^{-50}$. Green curve: $y = \gamma^x$.

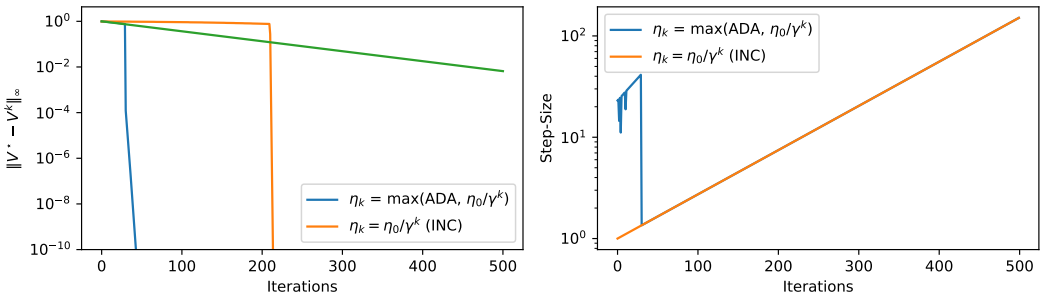

Figure 8.5: $\alpha = 10^{-10}$. $\eta_k = \max(\text{Adaptive}, \eta_0/\gamma^k)$ means the step-size is chosen as the largest between our "Adaptive" and the geometrically increasing of (Xiao, 2022). Green curve: $y = \gamma^x$.

2. We then show that enforcing a γ-contraction at the iteration when a state first switches preference between actions imposes a constraint on the policy update at that state.

3. Finally, we translate this constraint into a lower bound on the step-size, and show that these constraints occur at distinct iterations.

Setup and preliminaries. Fix $n > 0$ and $\delta \in (0, (1-\gamma)\gamma^n)$. We consider the discounted MDP depicted in Figure 8.1, which is the same construction used in the proof of Theorem 8.5.1. We briefly recall the key elements and notation needed for the present proof.

The state space is $\mathcal{S} = \{s_0, s_1, s'_1, \dots, s_n, s'_n\}$ and the action space is $\mathcal{A} = \{a_1, a_2\}$. The states $\{s_0, s_1, \dots, s_n\}$ form a chain of length $n+1$, indexed from left to right. The left-most state s_0 is absorbing and yields reward 1 at every time step. For each state s_i with $i = 1, \dots, n$, the agent has two available actions:

- action a_1, which deterministically moves the agent to s_{i-1} with zero reward;
- action a_2, which transitions to a state-specific absorbing state s'_i with reward

$$r_i := \gamma^{i+1} + \delta,$$

  and this reward is received at all subsequent time steps.

Formally, for $1 \leqslant i \leqslant n$,

$$\begin{aligned} &p(s_{i-1} \mid s_i, a_1) = 1, \quad r(s_i, a_1) = 0, \\ &p(s'_i \mid s_i, a_2) = 1, \quad r(s_i, a_2) = r_i, \\ &p(s'_i \mid s'_i, a) = 1, \quad r(s'_i, a) = r_i \qquad \forall a \in \mathcal{A}. \end{aligned}$$

The parameter δ is chosen small enough so that, despite the immediate reward r_i, the optimal action at every non-absorbing state s_i is a_1, which leads leftward toward the high-reward absorbing state s_0.

Let

$$c := \frac{1-\gamma}{8}.$$

Note that for $\gamma > 0.2$,

$$c < \frac{\sqrt{\gamma}}{1+\sqrt{\gamma}} \cdot \frac{1-\gamma}{2},$$

since $\frac{1}{4} < \frac{\sqrt{\gamma}}{1+\sqrt{\gamma}}$ in this regime.

We run Natural Policy Gradient (NPG), i.e., PMD with the negative entropy mirror map, starting from an initial policy satisfying

$$\pi^0(a_1 \mid s_i) = \alpha,$$

for all states s_i, where $\alpha \in (0,1)$ is a parameter chosen sufficiently small. In particular, throughout the proof we assume $\alpha \leqslant 1-\gamma$ and, when needed, take the limit $\alpha \to 0$.

NPG is the instance of PMD with negative-entropy regularisation. From the update of the exponentiated gradient algorithm in Chapter 2, it can be shown to admit the closed-form update

$$\pi^{k+1}(a \mid s) = \frac{\pi^k(a \mid s)\exp(\eta_k Q^k(s,a))}{\sum_{a'} \pi^k(a' \mid s)\exp(\eta_k Q^k(s,a'))}. \tag{8.18}$$

From the proof of Theorem 8.5.1, we will repeatedly use the fact that, regardless of the step-size sequence,

$$Q^k(s_i,a_1) \leqslant Q^k(s_i,a_2) \qquad \text{for all } i > k+1. \tag{8.19}$$

Step 1: Bounding the initial value error. We first show that the initial error is maximized at the left-most state. This shows that the left-most state controls the largest possible error, and therefore any contraction requirement will first affect the policy update in this state.

Lemma 8.10.1.

$$\|V^\star - V^0\|_\infty = V^\star(s_1) - V^0(s_1) \leqslant \gamma - \frac{\delta}{1-\gamma}.$$

Proof. By direct computation,

$$V^\star(s_1) - V^0(s_1) = \pi^0(a_2 \mid s_1)\frac{\gamma - r_1}{1-\gamma} = (1-\alpha)\Big(\gamma - \frac{\delta}{1-\gamma}\Big) \leqslant \gamma - \frac{\delta}{1-\gamma}.$$

We now show that this error upper-bounds the error at all other states. Observe that for all k,

$$\frac{\gamma^k - r_k}{1-\gamma} = \gamma^k - \frac{\delta}{1-\gamma} \leqslant (1-\alpha)\Big(\gamma - \frac{\delta}{1-\gamma}\Big) = V^\star(s_1) - V^0(s_1), \tag{8.20}$$

which holds provided $\alpha \leqslant 1-\gamma$.

Base case.

$$\begin{aligned} V^\star(s_2) - V^0(s_2) &= \alpha\gamma\big(V^\star(s_1) - V^0(s_1)\big) + (1-\alpha)\frac{\gamma^2 - r_2}{1-\gamma} \\ &\leqslant \alpha\gamma\big(V^\star(s_1) - V^0(s_1)\big) + (1-\alpha)\big(V^\star(s_1) - V^0(s_1)\big) \\ &\leqslant V^\star(s_1) - V^0(s_1). \end{aligned}$$

Inductive step. Assuming the bound holds for s_k, we obtain

$$\begin{aligned} V^\star(s_{k+1}) - V^0(s_{k+1}) &= \alpha\gamma\big(V^\star(s_k) - V^0(s_k)\big) + (1-\alpha)\frac{\gamma^{k+1} - r_{k+1}}{1-\gamma} \\ &\leqslant \alpha\gamma\big(V^\star(s_1) - V^0(s_1)\big) + (1-\alpha)\big(V^\star(s_1) - V^0(s_1)\big) \\ &\leqslant V^\star(s_1) - V^0(s_1), \end{aligned}$$

which concludes the proof. □

Step 2: Enforcing γ-contraction at a critical iteration. We now turn to the necessity argument. Fix a state s_k, and let k_0 be the first iteration such that

$$Q^{k_0}(s_k, a_1) > Q^{k_0}(s_k, a_2).$$

By (8.19), we necessarily have $k_0 \geqslant k-1$. Moreover, since $Q^t(s_k, \cdot)$ favors a_2 for all $t \leqslant k_0$, the policy mass on a_1 cannot have increased, and hence

$$\pi^{k_0}(a_1 \mid s_k) \leqslant \alpha.$$

We assume that the iterates satisfy the contraction bound in the theorem statement:

$$V^\star(s_k) - V^{k_0+1}(s_k) \leqslant \gamma^{k_0+1}\big(\|V^\star - V^0\|_\infty + c\big) \leqslant \gamma^{k_0+1}\Big(\gamma - \frac{\delta}{1-\gamma} + c\Big). \tag{8.21}$$

On the other hand, a direct lower bound yields

$$\begin{aligned} V^\star(s_k) - V^{k_0+1}(s_k) &= \pi^{k_0+1}(a_1 \mid s_k)\gamma\big(V^\star(s_{k-1}) - V^{k_0+1}(s_{k-1})\big) + \pi^{k_0+1}(a_2 \mid s_k)\frac{\gamma^k - r_k}{1-\gamma} \\ &\geqslant \pi^{k_0+1}(a_2 \mid s_k)\Big(\gamma^k - \frac{\delta}{1-\gamma}\Big). \end{aligned}$$

This lower bound arises by splitting the expected value according to the two possible actions at s_k, with the term involving a_2 providing a minimal contribution to the value error. Combining with (8.21), we obtain the necessary condition

$$\pi^{k_0+1}(a_2 \mid s_k) \leqslant \beta := \frac{\gamma^{k_0+1}\big(\gamma - \frac{\delta}{1-\gamma} + c\big)}{\gamma^k - \frac{\delta}{1-\gamma}}. \tag{8.22}$$

Here, β gives an upper bound on the probability mass that can be assigned to the suboptimal action a_2 while still achieving the γ-contraction of the value error.

Step 3: Translating contraction into a step-size lower bound. Recall from (8.22) that enforcing γ-contraction at iteration k_0 requires

$$\pi^{k_0+1}(a_2 \mid s_k) \leqslant \beta := \frac{\gamma^{k_0+1}\big(\gamma - \frac{\delta}{1-\gamma} + c\big)}{\gamma^k - \frac{\delta}{1-\gamma}}.$$

We now choose δ such that $\beta \leqslant \sqrt{\gamma}$, which in turn enforces

$$\pi^{k_0+1}(a_2 \mid s_k) \leqslant \sqrt{\gamma}.$$

This condition will then later translate directly into a lower bound on the NPG step-size. We start from the condition $\beta \leqslant \sqrt{\gamma}$:

$$\begin{aligned}
&\frac{\gamma^{k_0+1}\big(\gamma - \frac{\delta}{1-\gamma} + c\big)}{\gamma^k - \frac{\delta}{1-\gamma}} \leqslant \sqrt{\gamma} \\
\impliedby\ &\frac{\gamma^{k}\big(\gamma - \frac{\delta}{1-\gamma} + c\big)}{\gamma^k - \frac{\delta}{1-\gamma}} \leqslant \sqrt{\gamma}, \qquad \text{since } k_0 + 1 \geqslant k \\
\iff\ &\gamma^k\Big(\gamma - \frac{\delta}{1-\gamma} + c\Big) \leqslant \sqrt{\gamma}\Big(\gamma^k - \frac{\delta}{1-\gamma}\Big) \\
\impliedby\ &\gamma^k(\gamma + c) \leqslant \sqrt{\gamma}\Big(\gamma^k - \frac{\delta}{1-\gamma}\Big) \\
\iff\ &\gamma^{k-\frac{1}{2}}(\gamma + c) \leqslant \gamma^k - \frac{\delta}{1-\gamma} \\
\iff\ &\frac{\delta}{1-\gamma} \leqslant \gamma^{k-\frac{1}{2}}\big(\sqrt{\gamma} - \gamma - c\big).
\end{aligned}$$

Using the choice

$$c = \frac{1-\gamma}{8} < \frac{\sqrt{\gamma}}{1+\sqrt{\gamma}}\frac{1-\gamma}{2} = -\frac{\sqrt{\gamma}}{2}(1-\sqrt{\gamma}),$$

we further obtain

$$\beta \leqslant \sqrt{\gamma} \impliedby \frac{\delta}{1-\gamma} \leqslant \gamma^{k-\frac{1}{2}}\Big(\sqrt{\gamma} - \gamma - \frac{\sqrt{\gamma}}{2}(1-\sqrt{\gamma})\Big) = \gamma^{k-\frac{1}{2}}\Big(\frac{\sqrt{\gamma}}{2} - \frac{\gamma}{2}\Big),$$
$$\iff \delta \leqslant \frac{1}{2}(1-\gamma)(1-\sqrt{\gamma})\gamma^k.$$

This is precisely the condition on δ we impose, so we have $\pi^{k_0+1}(a_2 \mid s_k) \leqslant \sqrt{\gamma}$.

We now translate the requirement

$$\pi^{k_0+1}(a_2 \mid s_k) \leqslant \sqrt{\gamma}$$

into a constraint on the step-size. Letting $m = k_0$ for simplicity and using the closed-form NPG update (8.18), we have

$$\begin{aligned}
&\pi^{m+1}(a_2 \mid s_k) = \frac{\pi^m(a_2 \mid s_k)\exp(\eta_m Q^m(s,a_2))}{\pi^m(a_1 \mid s_k)\exp(\eta_m Q^m(s_k,a_1)) + \pi^m(a_2 \mid s_k)\exp(\eta_m Q^m(s_k,a_2))} \leqslant \sqrt{\gamma} \\
\iff &\pi^m(a_2 \mid s_k)\exp(\eta_m Q^m(s_k,a_2)) \\
&\qquad \leqslant \sqrt{\gamma}\Big(\pi^m(a_2 \mid s_k)\exp(\eta_m Q^m(s_k,a_2)) + \pi^m(a_1 \mid s_k)\exp(\eta_m Q^m(s_k,a_1))\Big) \\
\iff &\frac{1}{\sqrt{\gamma}} \leqslant 1 + \frac{\pi^m(a_1 \mid s_k)}{\pi^m(a_2 \mid s_k)}\exp\big(\eta_m(Q^m(s_k,a_1) - Q^m(s_k,a_2))\big) \\
\iff &\frac{1-\sqrt{\gamma}}{\sqrt{\gamma}}\frac{\pi^m(a_2 \mid s_k)}{\pi^m(a_1 \mid s_k)} \leqslant \exp\big(\eta_m(Q^m(s_k,a_1) - Q^m(s_k,a_2))\big) \\
\iff &\eta_m(Q^m(s_k,a_1) - Q^m(s_k,a_2)) \geqslant \log\Big(\frac{1-\sqrt{\gamma}}{\sqrt{\gamma}}\frac{\pi^m(a_2 \mid s_k)}{\pi^m(a_1 \mid s_k)}\Big).
\end{aligned}$$

Rewriting the logarithmic term yields

$$\eta_m \geqslant \frac{1}{Q^m(s_k,a_1) - Q^m(s_k,a_2)}\Big[\log\big((1-\sqrt{\gamma})\pi^m(a_2 \mid s_k)\big) + \log\Big(\frac{1}{\pi^m(a_1 \mid s_k)}\Big)\Big].$$

Recall that $\tilde{\pi}_{s_k}^{k_0+1}$ denotes the greedy policy at state s_k,

$$\tilde{\pi}_{s_k}^{k_0+1}(a_1) = 1, \qquad \tilde{\pi}_{s_k}^{k_0+1}(a_2) = 0,$$

since by definition of k_0 we have $Q^{k_0}(s_k,a_1) > Q^{k_0}(s_k,a_2)$. For the negative entropy mirror map,

the associated Bregman divergence is the Kullback–Leibler divergence. Therefore,

$$\begin{aligned} KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}) &= \sum_{a\in\mathcal{A}} \tilde{\pi}_{s_k}^{k_0+1}(a) \log \frac{\tilde{\pi}_{s_k}^{k_0+1}(a)}{\pi^{k_0}(a \mid s_k)} \\ &= \log \frac{1}{\pi^{k_0}(a_1 \mid s_k)}. \end{aligned}$$

Since $\pi^{k_0}(a_2 \mid s_k) \geqslant 1-\alpha$, this implies

$$\eta_{k_0} \geqslant \frac{1}{Q^{k_0}(s_k,a_1) - Q^{k_0}(s_k,a_2)} \Big[\log\big((1-\alpha)(1-\sqrt{\gamma})\big) + KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}) \Big].$$

Taking $\alpha \to 0$, the KL term dominates. In particular, since $\alpha < 1-\gamma$, we have

$$(1-\alpha)(1-\sqrt{\gamma}) > \gamma(1-\sqrt{\gamma}),$$

and imposing the stronger condition $\alpha < \gamma^2(1-\sqrt{\gamma})^2$ ensures

$$(1-\alpha)(1-\sqrt{\gamma}) > \sqrt{\alpha} \geqslant \sqrt{\pi^{k_0}(a_1 \mid s_k)}.$$

Consequently,

$$\begin{aligned} \eta_{k_0} &\geqslant \frac{1}{Q^{k_0}(s_k,a_1) - Q^{k_0}(s_k,a_2)} \Big[\log(\sqrt{\pi^{k_0}(a_1|s_k)}) + KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}) \Big] \\ &= \frac{1}{Q^{k_0}(s_k,a_1) - Q^{k_0}(s_k,a_2)} \Big[-\frac{1}{2} KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}) + KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}) \Big] \\ &= \frac{1}{2(Q^{k_0}(s_k,a_1) - Q^{k_0}(s_k,a_2))} KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}) \end{aligned} \tag{8.23}$$

which establishes the desired step-size lower bound.

Upper Bounding Q-value difference: We now upper-bound the Q-value difference appearing in the denominator of (8.23):

$$\begin{aligned} Q^{k_0}(s_k,a_2) &= \frac{r_k}{1-\gamma} = \frac{\gamma^{k+1}+\delta}{1-\gamma}, \\ Q^{k_0}(s_k,a_1) &= \gamma V^{k_0}(s_{k-1}) \leqslant \frac{\gamma^k}{1-\gamma}. \end{aligned}$$

Hence,

$$\begin{aligned} Q^{k_0}(s_k, a_1) - Q^{k_0}(s_k, a_2) &\leqslant \frac{\gamma^k}{1-\gamma} - \frac{\gamma^{k+1} + \delta}{1-\gamma} \\ &= \gamma^k - \frac{\delta}{1-\gamma} \\ &\leqslant \gamma^k. \end{aligned}$$

Plugging this into (8.23), we obtain the explicit step-size lower bound:

$$\eta_{k_0} \geqslant \frac{1}{2\gamma^k} KL(\tilde{\pi}_{s_k}^{k_0+1}, \pi_{s_k}^{k_0}).$$

Distinct iterations. As in the proof of Theorem 8.5.1, the iteration $k_0(s_k)$ at which state s_k switches action preference is distinct across states. Thus, there exist n distinct iterations $\{k_1, \ldots, k_n\}$ at which the above lower bound must hold, completing the proof. □

8.10.3 MDP on which Distribution-Mismatch Coefficient Scales with Size of State Space

We construct an MDP on which

$$\theta_\rho = \frac{1}{1-\gamma} \left\| \frac{d_\rho^\star}{\rho} \right\|_\infty,$$

scales with $|\mathcal{S}|$, and hence so does the iteration complexity of the bound of Xiao (2022) for exact PMD.

Consider an MDP with state-space $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ of size n and arbitrary action space $\mathcal{A}$. s_1 is an absorbing state giving out rewards of 1 at each time-step, regardless of the action taken, i.e

$$p(s_1|s_1, a) = 1, \quad r(s_1, a) = 1 \quad \forall a \in \mathcal{A}.$$

All others states have an action, say a_1, that gives out a reward of 1 and with probability $1 - \delta$ brings the agent to state s_1 for some $\delta > 0$ and spreads the remaining δ probability arbitrarily amongst the other states. The other actions have arbitrary rewards strictly less than 1 associated

to them, and arbitrary transition probabilities that place 0 mass on state s_1, i.e

$$p(s_1|s,a_1) = 1-\delta, \quad r(s,a_1) = 1 \quad \forall s \neq s_1,$$
$$p(s_1|s,a) = 0, \quad r(s,a) < 1 \quad \forall s \neq s_1, \forall a \neq a_1.$$

Denote $r_{\max} = \max_{s\neq s_1, a\neq a_1} r(s,a) < 1$. The following condition ensures that a_1 is the optimal action in all states,

$$\delta \leqslant \frac{1-\gamma}{\gamma}(1-r_{\max})$$

so that $\pi^\star(s) = a_1$ for all states s. To see this, consider $s_i \neq s_1$, $a_m \neq a_1$ and an arbitrary policy π,

$$\begin{aligned} Q^\pi(s_i,a_1) &= 1+\gamma\Big(\frac{1-\delta}{1-\gamma} + \sum_{j=2}^n p(s_j|s_i,a_1)V^\pi(s_j)\Big) \\ &\geqslant 1+\gamma\frac{1-\delta}{1-\gamma} \\ Q^\pi(s_i,a_m) &= r(s_i,a_m) + \gamma\sum_{j=2}^n p(s_j|s_i,a_1)V^\pi(s_j) \\ &\leqslant r_{\max} + \gamma\frac{1}{1-\gamma} \end{aligned}$$

and solving

$$r_{\max} + \gamma\frac{1}{1-\gamma} \leqslant 1+\gamma\frac{1-\delta}{1-\gamma}$$

will yield the condition above. Then for $t \geqslant 1$ (abusing notation, s_t denotes the state at time t),

$$\begin{aligned} \mathbb{P}^{\pi^\star}(s_t = s_1|s_0 = s) &= \sum_{s'}\mathbb{P}^{\pi^\star}(s_t = s_1, s_{t-1} = s'|s_0 = s) \\ &= \sum_{s'} p(s_1|s',a_1)\mathbb{P}^{\pi^\star}(s_{t-1} = s'|s_0 = s) \\ &\geqslant \sum_{s'}(1-\delta)\mathbb{P}^{\pi^\star}(s_{t-1} = s'|s_0 = s) \\ &= 1-\delta \end{aligned}$$

and

$$\begin{aligned} d_\rho^\star(s_1) &= (1-\gamma)\sum_s \rho(s)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}^{\pi^\star}(s_t = s_1 | s_0 = s) \\ &\geqslant (1-\gamma)\sum_s \rho(s)\sum_{t=1}^{\infty}\gamma^t (1-\delta) \\ &\geqslant (1-\gamma)\sum_s \rho(s)\frac{\gamma}{1-\gamma}(1-\delta) \\ &= \gamma(1-\delta). \end{aligned}$$

Now

$$\left\|\frac{d_\rho^\star}{\rho}\right\|_\infty \geqslant \frac{d_\rho^\star(s_1)}{\rho(s_1)} \geqslant \frac{\gamma(1-\delta)}{\rho(s_1)}$$

and depending on what ρ you consider, θ_ρ can be arbitrarily large. In particular, the natural choice of the uniform starting-state distribution $\rho(s) = 1/n$ leads to

$$\theta_\rho \geqslant n\frac{\gamma(1-\delta)}{(1-\gamma)}$$

which gives an iteration complexity under the result of Xiao (2022) for an ε-optimal policy of

$$n\frac{\gamma(1-\delta)}{(1-\gamma)}\log\frac{2}{(1-\gamma)\varepsilon}.$$

Recall that $n = |\mathcal{S}|$, so this iteration complexity scales linearly with the size of the state space.

8.10.4 Family of MDPs on which Sub-Optimality Gaps can be Made Arbitrarily Small

We present how to construct a family of MDPs on which $\Delta^k(s)$ defined in Section 8.4 can be made arbitrarily small.

Consider an arbitrary MDP $\mathcal{M}$ with state space $\mathcal{S}$ and action space $\mathcal{A}$. For each state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, create a duplicate action a' s.t the transitions from that action in that state are the same as for the original pair, i.e

$$p(s'|s, a) = p(s'|s, a') \quad \forall s' \in \mathcal{S}$$

and the reward is shifted down by $\delta > 0$ from the original reward, i.e

$$r(s, a') = r(s, a) - \delta.$$

This results in a new MDP $\mathcal{M}'$ with an augmented action space $\mathcal{A}'$, that is twice the size of the action space of the original MDP $\mathcal{M}$. In terms of action-value of an arbitrary policy π, this results in

$$Q^{\pi}_{\mathcal{M}'}(s, a) - Q^{\pi}_{\mathcal{M}'}(s, a') = \delta,$$

where the notation $Q^{\pi}_{\mathcal{M}'}$ refers to action-values in the MDP $\mathcal{M}'$. In terms of sub-optimality gaps, this gives

$$\Delta^{\pi}(s) \leqslant \delta.$$

Choosing δ small enough, we can make the step-size of Khodadadian et al. (2021) arbitrarily large, at least in early iterations. The step-size condition (8.8) of Theorem 8.4.1 will be less affected by this issue as it does not depend directly on $\Delta^k(s)$, and not at all in the first iteration. Beyond its generality to PMD, this illustrates the benefit of our result restricted to NPG over the result of Khodadadian et al. (2021).

Chapter 9

Conclusion and Open Questions

This thesis has explored the theoretical foundations of gradient-based methods for sequential decision-making, with a specific focus on the challenges and opportunities presented by high-dimensional settings. Motivated by the empirical success of large-scale gradient-based optimisation, this work sought to bridge part of the gap between empirical scalability and theoretical understanding across three complementary directions: establishing fundamental separations between low and high dimension regimes, exploiting low-dimensional structure, and analysing the performance limits of policy gradient methods in reinforcement learning.

9.1 Summary of Contributions

9.1.1 Part I - High Dimensional Problems are Fundamentally Different

We established that some high-dimensional problems are intrinsically distinct from their low-dimensional counterparts, necessitating different algorithmic approaches.

- **Online Convex Optimisation (OCO):** We demonstrated that high-dimensional settings impose unique geometric constraints that require a fundamentally different type of regularisation strategy to low-dimensional settings for gradient-based algorithms like Follow-the-Regularised-Leader (FTRL).
- **Multi-Batch Reinforcement Learning:** We identified that the ability to learn efficiently from data is fundamentally limited by the frequency of interaction, which must scale with the problem's dimensionality.

9.1.2 Part II - Exploiting Sparsity

We demonstrated that prohibitively large dimensions can be overcome by leveraging low-dimensional structures, specifically sparsity.

- **Sparse Stochastic Shortest Path:** We proved that it is possible to make the regret of stochastic shortest-path problems depend on the underlying sparsity rather than the ambient dimension of the state-action space. This can be achieved through gradient-based methods like Online Mirror Descent.

9.1.3 Part III - Gradient-Based Methods in Reinforcement Learning

We provided sharp analyses of gradient-based algorithms in fundamental reinforcement learning settings to characterise their performance.

- **Stochastic Gradient Bandit:** We characterised a sharp transition in the performance of gradient bandits from logarithmic to polynomial regret, showing that learning rates must be carefully tuned relative to the problem's sub-optimality gaps.
- **Policy Mirror Descent:** We established that achieving optimal, dimension-independent convergence in reinforcement learning is possible using general mirror-descent inspired alternatives to policy iteration.

Collectively, these results clarify the interplay between dimensionality, geometry, and gradient-based optimisation. While high dimensionality poses significant theoretical challenges, this thesis demonstrates that through careful algorithmic design, gradient-based methods can achieve optimal performance in complex sequential decision-making problems but there remains plenty to understand.

9.2 Open Questions and Future Directions

This thesis leaves open several natural and important questions, pointing to promising avenues for future research. While specific technical extensions have been discussed within the individual chapters, we outline here more general open questions and broader conceptual directions.

- **Beyond FTRL:** The identified non-universality of Follow-The-Regularised-Leader (FTRL) with fixed-regularisation in Online Convex Optimisation suggests that, while practical workarounds like doubling tricks or switches in regularisation exist, there is a compelling

opening for a new generation of OCO algorithms that are intrinsically adaptive to the dimensional regime. Given that FTRL has few known theoretical weaknesses, our results highlight a significant limitation that invites the development of new methods for universal online learning.

- **Beyond Sparsity: Exploiting General Low-Dimensional Structure:** While this thesis focused on sparsity as a tractable form of structure, many real-world high-dimensional problems exhibit other forms of low-dimensional latent geometry (e.g. low-rank transitions). Developing methods that can automatically adapt to and exploit these diverse structural properties, ensuring that the computational and sample complexity remains governed by the intrinsic difficulty of the task rather than the extrinsic representation of the data is a direction of significant theoretical and practical interest in stochastic shortest path and sequential decision-making problems more broadly.

- **Bridging the Gap to Practical Reinforcement Learning:** While this thesis provides sharp analyses for several gradient-based methods, a significant gap remains between current theoretical guarantees and realistic settings. Even in the simple tabular setting with inexact gradients and generative models, we lack gradient-based methods that achieve minimax optimal sample complexity. Establishing a fundamental understanding of these methods in simplified settings is a vital prerequisite for developing a rigorous theory of gradient-based learning in the more complex environments encountered in practice.

- **Characterising Algorithms Beyond the Optimal Regime:** A central theme of this thesis has been the characterisation of algorithmic performance as a function of hyperparameters, such as learning rates and regularisers, rather than merely establishing guarantees for their optimal values. In practice, parameter settings are frequently sub-optimal, making the study of algorithm-specific lower bounds, which are relatively under-explored in current literature, a vital measure of an algorithm's practical robustness. While research often prioritises the search for an "optimal" algorithm under "optimal" parameters, we argue that characterising the global behaviour of widely-used frameworks, or understanding precisely how performance degrades with parameter choice, is an equally valuable contribution. Such analyses are essential for identifying which methods are truly viable for deployment in the uncertain environments encountered in the real world.

  Ultimately, this work advocates for an approach to theory that prioritises the deep characterisation of algorithms well beyond the narrow regimes where they are optimal. By

shifting focus from finding theoretically "best" methods to understanding the global behaviour and inherent trade-offs of practical gradient-based frameworks, we move closer to a comprehensive understanding of why these methods scale so effectively in high-dimensional settings.

Bibliography

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellert Weisz. POLITEX: Regret bounds for policy iteration using expert prediction. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3692–3702, 2019.

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of the Twenty-First Conference on Learning Theory*, 2008.

Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin. Interior-point methods for full-information and bandit online learning. *IEEE Transactions on Information Theory*, 58(7): 4164–4175, 2012.

Alekh Agarwal, Sham Kakade, and Lin F. Yang. Model-based reinforcement learning with a generative model is minimax optimal. In *Proceedings of 33rd Conference on Learning Theory*, volume 125, pages 67–83, 2020.

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Alekh Agarwal, Nan Jiang, Sham M. Kakade, and Wen Sun. *Reinforcement Learning: Theory and Algorithms*. rltheorybook.github.io, 2022. URL `https://rltheorybook.github.io/rltheorybook_ABJKS.pdf`. Working draft, January 31, 2022.

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceedings of the 25th Annual Conference on Learning Theory*, 2012.

Carlo Alfano and Patrick Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization, 2022a. Preprint, arXiv: 2209.15382.

Carlo Alfano and Patrick Rebeschini. Linear convergence for natural policy gradient with log-linear policy parametrization, 2022b. Preprint arXiv:2209.15382.

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 30681–30725, 2023a.

Carlo Alfano, Rui Yuan, and Patrick Rebeschini. A novel framework for policy mirror descent with general parameterization and linear convergence. In *Advances in Neural Information Processing Systems*, volume 36, 2023b.

Philip Amortila, Nan Jiang, and Tengyang Xie. A variant of the wang-foster-kakade lower bound for the discounted setting, 2020. Preprint, arXiv: 2011.01075.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine Learning*, 91 (3):325–349, 2013.

Dominique Azé and Jean-Paul Penot. Uniformly convex and uniformly smooth convex functions. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 4, pages 705–730, 1995.

Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Keith Ball, Eric A. Carlen, and Elliott H. Lieb. Sharp uniform convexity and smoothness inequalities for trace norms. *Inventiones Mathematicae*, 115:463–482, 1994.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(58):1705–1749, 2005.

Necdet Batir. Bounds for the gamma function, 2017. Preprint, arXiv: 1705.06167.

Dorian Baudry, Emmeran Johnson, Simon Vary, Ciara Pike-Burke, and Patrick Rebeschini. Does stochastic gradient really succeed for bandits? In *The 39th Annual Conference on Neural Information Processing Systems*, 2025.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Richard Bellman. *Dynamic Programming*. Dover Publications, 1957.

Richard Bellman, Robert Kalaba, and Bella Kotkin. Polynomial approximation–a new computational technique in dynamic programming: Allocation processes. *Mathematics of Computation*, 17(82):155–161, 1963.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning, 2019. Preprint, arXiv: 1912.06680.

Joseph Bertrand. Solution d'un probleme. *CR Acad. Sci. Paris*, 105(1887):369, 1887.

Dimitri P Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011.

Dimitri P Bertsekas and John N Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996.

Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods, 2019. Preprint, arXiv: 1906.01786.

Jalaj Bhandari and Daniel Russo. On the linear convergence of policy gradient methods for finite mdps. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 2386–2394, 2021.

Aditya Bhaskara, Ashok Cutkosky, Ravi Kumar, and Manish Purohit. Online learning with imperfect hints. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 822–831, 2020.

Avrim Blum and Adam Tauman Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 35:193–205, 1997.

A. V. Boyd. Inequalities for mills' ratio. *Reports of Statistical Application Research (Union of Japanese Scientists and Engineers)*, 6(1959):44–46, 1959.

Sébastien Bubeck. Introduction to online optimization. *Lecture notes*, 2011.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4), 2015.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: Stochastic and adversarial bandits. In *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23, 2012.

Sébastien Bubeck, Michael Cohen, and Yuanzhi Li. Sparsity, variance and curvature in multi-armed bandits. In *Proceedings of Algorithmic Learning Theory*, volume 83, pages 111–127, 2018.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization. *Operations Research*, 70 (4):2563–2578, 2021.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3), 1997a.

Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997b.

Sunrit Chakraborty, Saptarshi Roy, and Ambuj Tewari. Thompson sampling for high-dimensional sparse linear contextual bandits. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 3979–4008, 2023.

Gong Chen and Marc Teboulle. Convergence Analysis of a Proximal-Like Minimization Algorithm Using Bregman Functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

Lin Chen, Bruno Scherrer, and Peter L Bartlett. Infinite-horizon offline reinforcement learning with linear function approximation: Curse of dimensionality and algorithm, 2021a. Preprint, arXiv: 2103.09847.

Liyu Chen and Haipeng Luo. Finding the stochastic shortest path with low regret: the adversarial cost and unknown transition case. *Proceedings of the 38th International Conference on Machine Learning*, 139:1651–1660, 2021.

Liyu Chen, Mehdi Jafarnia-Jahromi, Rahul Jain, and Haipeng Luo. Implicit finite-horizon approximation and efficient optimal algorithms for stochastic shortest path. *Advances in Neural Information Processing Systems*, 34:10849–10861, 2021b.

Liyu Chen, Haipeng Luo, and Chen-Yu Wei. Minimax regret for stochastic shortest path with adversarial costs and known transition. *Proceedings of the 34th Conference on Learning Theory*, 134:1180–1215, 2021c.

Liyu Chen, Rahul Jain, and Haipeng Luo. Improved no-regret algorithms for stochastic shortest path with linear MDP. *Proceedings of the 39th International Conference on Machine Learning*, 162:3204–3245, 2022a.

Liyu Chen, Haipeng Luo, and Aviv Rosenberg. Policy optimization for stochastic shortest path. *Proceedings of the 35th Conference on Learning Theory*, 178:982–1046, 2022b.

Zaiwei Chen and Siva Theja Maguluri. Sample complexity of policy-based methods under off-policy sampling and linear function approximation. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 11195–11214, 2022.

James A. Clarkson. Uniformly convex spaces. *Transactions of the American Mathematical Society*, 40:396–414, 1936.

Alon Cohen, Yonathan Efroni, Yishay Mansour, and Aviv Rosenberg. Minimax regret for stochastic shortest path. *Advances in Neural Information Processing Systems*, 34:28350–28361, 2021.

John H. Conway, Ronald H. Hardin, and Neil J. A. Sloane. Packing lines, planes, etc.: packings in Grassmannian spaces. *Experimental Mathematics*, 5(2):139 – 159, 1996.

Thomas M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

Ashok Cutkosky. Artificial constraints and hints for unbounded online learning. In *Proceedings of the 32nd Conference on Learning Theory*, volume 99, pages 874–894, 2019.

Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 1493–1529, 2018.

Christoph Dann, Lihong Li, Wei Wei, and Emma Brunskill. Policy certificates: Towards accountable reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1507–1516, 2019.

Alexandre d'Aspremont, Cristóbal Guzmán, and Martin Jaggi. Optimal affine-invariant smooth minimization algorithms. *SIAM Journal on Optimization*, 28(3):2384–2405, 2018.

Rémy Degenne and Vianney Perchet. Anytime optimal algorithms in stochastic multi-armed bandits. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1587–1595, 2016.

Ofer Dekel, Arthur Flajolet, Nika Haghtalab, and Patrick Jaillet. Online learning with a hint. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Benjamin Doerr. An elementary analysis of the probability that a binomial random variable exceeds its expectation. *Statistics & Probability Letters*, 139:67–74, 2018.

Yaqi Duan, Zeyu Jia, and Mengdi Wang. Minimax-optimal off-policy evaluation with linear function approximation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 2701–2709, 2020.

John Duchi, Feng Ruan, and Chulhee Yun. Minimax bounds on stochastic batched convex optimization. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 3065–3162, 2018.

Hossein Esfandiari, Amin Karbasi, Abbas Mehrabian, and Vahab Mirrokni. Regret bounds for batched bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7340–7348, 2021.

John Fearnley. Exponential lower bounds for policy iteration. In *Automata, Languages and Programming*, pages 551–562, 2010.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, 1997.

Minbo Gao, Tianle Xie, Simon S Du, and Lin F Yang. A provably efficient algorithm for linear markov decision process with low switching cost, 2021. Preprint, arXiv: 2101.00494.

Zijun Gao, Yanjun Han, Zhimei Ren, and Zhengqing Zhou. Batched multi-armed bandits problem. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Dan Garber and Elad Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 541–549, 2015.

Armengol Gasull and Frederic Utzet. Approximating mills ratio. *Journal of Mathematical Analysis and Applications*, 420(2):1832–1853, 2014.

Khashayar Gatmiry, Jon Schneider, and Stefanie Jegelka. Computing optimal regularizers for online linear optimization. In *Proceedings of the 38th Conference on Learning Theory*, volume 291, pages 2362–2402, 2025.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2160–2169, 2019.

John C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the royal statistical society series b-methodological*, 41:148–164, 1979.

Geoffrey J Gordon. Regret bounds for prediction problems. In *Proceedings of the 12 annual conference on Computational learning theory*, pages 29–40, 1999.

Cristóbal Guzmán. Open problem: The oracle complexity of smooth convex optimization in nonstandard settings. In *Proceedings of The 28th Conference on Learning Theory*, 2015.

Cristóbal Guzmán and Arkadi Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1–14, 2015.

Yanjun Han, Zhengqing Zhou, Zhengyuan Zhou, Jose Blanchet, Peter W Glynn, and Yinyu Ye. Sequential batch learning in finite-action linear contextual bandits, 2020. Preprint, arXiv: 2004.06321.

Olof Hanner. On the uniform convexity of Lp and lp. *Arkiv för Matematik*, 3(3):239 – 244, 1956.

Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. In *JACM*, 2010.

Botao Hao, Tor Lattimore, and Mengdi Wang. High-dimensional sparse linear bandits. In *Advances in Neural Information Processing Systems*, volume 33, pages 10753–10763, 2020.

Nicholas JA Harvey, Christopher Liaw, Edwin Perkins, and Sikander Randhawa. Optimal anytime regret with two experts. *Mathematical Statistics and Learning*, 6(1):87–142, 2023.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Proceedings of the Twenty-First Conference on Learning Theory*, 2008.

David P. Helmbold, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8, 1998.

Romain Hollanders, Jean-Charles Delvenne, and Raphaël M. Jungers. The complexity of policy iteration is exponential for discounted markov decision processes. In *51st IEEE Conference on Decision and Control*, pages 5997–6002, 2012.

Ronald A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

Jiawei Huang, Jinglin Chen, Li Zhao, Tao Qin, Nan Jiang, and Tie-Yan Liu. Towards deployment-efficient reinforcement learning: Lower bound and optimality. In *The 10th International Conference on Learning Representations*, 2022.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in online linear prediction: Curved constraint sets and other regularities. *Journal of Machine Learning Research*, 18(145):1–31, 2017.

Anatoli Iouditski and Yuri Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions, 2014. Preprint, arXiv: 1401.1792.

Mehdi Jafarnia-Jahromi, Liyu Chen, Rahul Jain, and Haipeng Luo. Posterior sampling-based online learning for the stochastic shortest path model. *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence*, 216:922–931, 2023.

Emmeran Johnson, Ciara Pike-Burke, and Patrick Rebeschini. Optimal convergence rate for exact policy mirror descent in discounted markov decision processes. In *37th Conference on Neural Information Processing Systems*, 2023.

Emmeran Johnson, Ciara Pike-Burke, and Patrick Rebeschini. Sample-efficiency in multi-batch reinforcement learning: The need for dimension-dependent adaptivity. In *The 12th International Conference on Learning Representations*, 2024.

Emmeran Johnson, David Martínez-Rubio, Ciara Pike-Burke, and Patrick Rebeschini. On the necessity of adaptive regularisation: Optimal anytime online learning on ℓ_p-balls. In *The 39th Annual Conference on Neural Information Processing Systems*, 2025a.

Emmeran Johnson, Alberto Rumi, Ciara Pike-Burke, and Patrick Rebeschini. Stochastic shortest path with sparse adversarial costs. In *The 39th Annual Conference on Neural Information Processing Systems*, 2025b.

Anatoli B. Juditsky and Arkadii S. Nemirovski. First Order Methods for Nonsmooth Convex Large-Scale Optimization, I: General Purpose Methods. In *Optimization for Machine Learning*, pages 1–28. MIT Press, 2010.

Kwang-Sung Jun, Kevin Jamieson, Robert Nowak, and Xiaojin Zhu. Top arm identification in multi-armed bandits with batch arm pulls. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 139–148, 2016.

Sham Kakade and John Langford. Approximately Optimal Approximate Reinforcement Learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pages 267–274, 2002a.

Sham M Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, 2002b.

Michael Kearns and Satinder Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in Neural Information Processing Systems*, volume 11, 1998.

Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Projection-free optimization on uniformly convex sets. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 19–27, 2021a.

Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Local and global uniform convexity conditions, 2021b. Preprint, arXiv: 2102.05134.

Thomas Kerdreux, Christophe Roux, Alexandre d'Aspremont, and Sebastian Pokutta. Linear bandits on uniformly convex sets. *Journal of Machine Learning Research*, 22(284):1–23, 2021c.

Sajad Khodadadian, Prakirt Raj Jhunjhunwala, Sushil Mahavir Varma, and Siva Theja Maguluri. On the linear convergence of natural policy gradient algorithm. In *60th IEEE Conference on Decision and Control*, pages 3794–3799, 2021.

Toshinori Kitamura, Tadashi Kozuno, Yunhao Tang, Nino Vieillard, Michal Valko, Wenhao Yang, Jincheng Mei, Pierre Ménard, Mohammad Gheshlaghi Azar, Rémi Munos, Olivier Pietquin, Matthieu Geist, Csaba Szepesvári, Wataru Kumagai, and Yutaka Matsuo. Regularization and variance-weighted regression achieves minimax optimality in linear MDPs: Theory and practice, 2023. Preprint, arXiv: 2305.13185.

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R. Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14):7684–7689, 2020.

Junpei Komiyama and Masaaki Imaizumi. High-dimensional contextual bandit problem without sparsity. In *Advances in Neural Information Processing Systems*, volume 36, pages 49416–49427, 2023.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, volume 12, 1999.

Joon Kwon and Vianney Perchet. Gains and losses are fundamentally different in regret minimization: The sparse case. *Journal of Machine Learning Research*, 17(227):1–32, 2016.

Joon Kwon, Vianney Perchet, and Claire Vernade. Sparse stochastic bandits. In *Proceedings of the 30th Conference on Learning Theory*, volume 65, pages 1269–1270, 2017.

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of machine learning research*, 4:1107–1149, 2003.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Guanghui Lan. Policy Mirror Descent for Reinforcement Learning: Linear Convergence, New Sampling Complexity, and Generalized Problem Classes. *Mathematical Programming*, 2021.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. *Batch Reinforcement Learning*, pages 45–73. Springer, 2012.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Tor Lattimore, Csaba Szepesvari, and Gellert Weisz. Learning with good feature representations in bandits and in RL with a generative model. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 5662–5670, 2020.

Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Analysis of classification-based policy iteration algorithms. *Journal of Machine Learning Research*, 17(19):1–30, 2016.

Daniel Levy and John C Duchi. Necessary and sufficient geometries for gradient methods. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Bin Li and Steven C. H. Hoi. Online portfolio selection: A survey. *ACM Comput. Surv.*, 46(3), 2014.

Gen Li, Yuting Wei, Yuejie Chi, Yuantao Gu, and Yuxin Chen. Breaking the sample size barrier in model-based reinforcement learning with a generative model. In *Advances in Neural Information Processing Systems*, volume 33, pages 12861–12872, 2020.

Gen Li, Yuting Wei, Yuejie Chi, and Yuxin Chen. Softmax policy gradient methods can take exponential time to converge. *Mathematical Programming*, 201(1-2):707–802, 2023a.

Ke Li, Yun Yang, and Naveen N Narisetty. Regret lower bound and optimal algorithm for high-dimensional contextual linear bandit. *Electronic Journal of Statistics*, 15(2):5652–5695, 2021.

Wenjie Li, Adarsh Barik, and Jean Honorio. A simple unified framework for high dimensional bandit problems. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 12619–12655, 2022.

Wenye Li, Hongxu Chen, Jiacai Liu, and Ke Wei. Policy mirror descent with temporal difference learning: Sample complexity under online markov data, 2025. Preprint, arXiv:2512.24056.

Yan Li, Guanghui Lan, and Tuo Zhao. Homotopic policy mirror descent: Policy convergence, implicit regularization, and improved sample complexity. *Mathematical Programming*, 2023b.

N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

Jiacai Liu, Wenye Li, and Ke Wei. Elementary analysis of policy gradient methods, 2024. Preprint, arXiv:2404.03372.

Jiacai Liu, Wenye Li, Dachao Lin, Ke Wei, and Zhihua Zhang. On the convergence of projected policy gradient for any constant step sizes. *Journal of Machine Learning Research*, 26(193): 1–35, 2025.

Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. In *Advances in Neural Information Processing Systems*, volume 33, pages 7624–7636, 2020.

Haipeng Luo and Robert Schapire. Towards minimax online learning with unknown time horizon. In *Proceedings of the 31st International Conference on Machine Learning*, pages 226–234, 2014.

Haipeng Luo and Robert E. Schapire. Achieving all with no parameters: Adanormalhedge. In *Proceedings of The 28th Conference on Learning Theory*, volume 40, pages 1286–1304, 2015.

Tatsuya Matsushima, Hiroki Furuta, Yutaka Matsuo, Ofir Nachum, and Shixiang Gu. Deployment-efficient reinforcement learning via model-based offline optimization. In *The 9th International Conference on Learning Representations*, 2021.

Prathamesh Mayekar and Himanshu Tyagi. Limits on gradient compression for stochastic optimization. In *IEEE International Symposium on Information Theory (ISIT)*, 2020.

Brendan D McKay. On littlewood's estimate for the binomial distribution. *Advances in Applied Probability*, 21(2):475–478, 1989.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6820–6829, 2020.

Jincheng Mei, Yue Gao, Bo Dai, Csaba Szepesvari, and Dale Schuurmans. Leveraging non-uniformity in first-order non-convex optimization. In *Proceedings of the 38th International Conference on Machine Learning*, pages 7555–7564, 2021.

Jincheng Mei, Zixin Zhong, Bo Dai, Alekh Agarwal, Csaba Szepesvari, and Dale Schuurmans. Stochastic gradient succeeds for bandits. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 24325–24360, 2023.

Jincheng Mei, Bo Dai, Alekh Agarwal, Sharan Vaswani, Anant Raj, Csaba Szepesvári, and Dale Schuurmans. Small steps no more: Global convergence of stochastic gradient bandits for arbitrary learning rates. In *Advances in Neural Information Processing Systems*, 2024.

Shahar Mendelson. Learning bounded subsets of lp. *IEEE Transactions on Information Theory*, 67(8):5269–5282, 2021.

Alberto Maria Metelli, Matteo Pirotta, Daniele Calandriello, and Marcello Restelli. Safe policy iteration: A monotonically improving approximate policy iteration approach. *Journal of Machine Learning Research*, 22(97):1–83, 2021.

Zakaria Mhammedi. Exploiting the curvature of feasible sets for faster projection-free online learning, 2022. Preprint, arXiv: 2205.11470.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Marco Molinaro. Strong convexity of feasible sets in off-line and online optimization. *Mathematics of Operations Research*, 48(2):865–884, 2023.

Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, 2005.

Arkadij Nemirovskij and David Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley, 1983.

Yurii Nesterov. *Lectures on convex optimization*, volume 137. Springer, 2018.

Yurii Nesterov and Arkadi Nemirovski. On first-order algorithms for l1/nuclear norm minimization. *Acta Numerica*, 22:509–575, 2013.

Gergely Neu. First-order regret bounds for combinatorial semi-bandits. *Proceedings of the 28th Conference on Learning Theory*, 40:1360–1375, 2015.

Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. Reinforcement learning for optimized trade execution. In *Proceedings of the 23rd International Conference on Machine Learning*, page 673–680, 2006.

Francesco Orabona. A modern introduction to online learning, 2019. Preprint, arXiv: 1912.13213.

Francesco Orabona and Kwang-Sung Jun. Tight concentrations and confidence sequences from the regret of universal portfolio. *IEEE Transactions on Information Theory*, 70(1):436–455, 2024.

Francesco Orabona and David Pal. Optimal non-asymptotic lower bound on the minimax regret of learning with expert advice, 2015. Preprint, arXiv: 1511.02176.

Francesco Orabona and Dávid Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018.

Grigoris Paouris, Petros Valettas, and Joel Zinn. Random version of dvoretzky's theorem in lpn. *Stochastic Processes and their Applications*, 2017.

Christos Pelekis and Jan Ramon. A lower bound on the probability that a binomial random variable is exceeding its mean. *Statistics & Probability Letters*, 119:305–309, 2016.

Vianney Perchet, Philippe Rigollet, Sylvain Chassang, and Erik Snowberg. Batched bandit problems. In *Proceedings of The 28th Conference on Learning Theory*, volume 40, pages 1456–1456, 2015.

Kimon Protopapas and Anas Barakat. Policy mirror descent with lookahead. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994.

Dan Qiao and Yu-Xiang Wang. Near-optimal deployment efficiency in reward-free reinforcement learning with linear function approximation. In *The 11th International Conference on Learning Representations*, 2023.

Dan Qiao, Ming Yin, Ming Min, and Yu-Xiang Wang. Sample-efficient reinforcement learning with loglog(T) switching cost. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 18031–18061, 2022.

Dan Qiao, Ming Yin, and Yu-Xiang Wang. Logarithmic switching cost in reinforcement learning beyond linear MDPs, 2023. Preprint, arXiv: 2302.12456.

Garvesh Raskutti and Sayan Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, 2015.

Marc Renault. Lost (and found) in translation: André's actual method and its application to the generalized ballot problem. *The American Mathematical Monthly*, 115(4):358–363, 2008.

Herbert Robbins. A remark on stirling's formula. *The American mathematical monthly*, 62(1): 26–29, 1955.

Herbert E. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, 1970.

Aviv Rosenberg and Yishay Mansour. Stochastic shortest path with adversarially changing costs. *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 2936–2942, 2021.

Aviv Rosenberg, Alon Cohen, Yishay Mansour, and Haim Kaplan. Near-optimal regret bounds for stochastic shortest path. *Proceedings of the 37th International Conference on Machine Learning*, 119:8210–8219, 2020.

Yufei Ruan, Jiaqi Yang, and Yuan Zhou. Linear bandits with limited adaptivity and learning distributional optimal design. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, page 74–87, 2021.

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1314–1322, 2014a.

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1314–1322, 2014b.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2021. Preprint, arXiv: 1707.06347.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5668–5675, 2020.

Aaron Sidford, Mengdi Wang, Xian Wu, Lin Yang, and Yinyu Ye. Near-optimal time and sample complexities for solving markov decision processes with a generative model. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, L. Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016.

Mahdi Soleymani and Hessam Mahdavifar. New packings in grassmannian space. In *2021 IEEE International Symposium on Information Theory*, pages 807–812, 2021.

Karthik Sridharan. *Learning From An Optimization Viewpoint*. PhD thesis, Toyota Technological Institute, 2012.

Karthik Sridharan and Ambuj Tewari. Convex games in banach spaces. In *Proceedings of the 23rd Conference on Learning Theory*, 2010.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, second edition, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, 1999.

Jean Tarbouriech, Evrard Garcelon, Michal Valko, Matteo Pirotta, and Alessandro Lazaric. No-regret exploration in goal-oriented reinforcement learning. *Proceedings of the 37th International Conference on Machine Learning*, 119:9428–9437, 2020.

Jean Tarbouriech, Runlong Zhou, Simon S Du, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Stochastic shortest path: Minimax, parameter-free and towards horizon-free regret. *Advances in Neural Information Processing Systems*, 34:6843–6855, 2021.

Jérôme Taupin, Yassir Jedra, and Alexandre Proutiere. Best policy identification in discounted linear MDPs. In *Sixteenth European Workshop on Reinforcement Learning*, 2023.

Gerald Tesauro. Temporal difference learning and td-gammon. *Commun. ACM*, 38(3), 1995.

William Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Tim van Erven. Why FTRL Is Better than Online Mirror Descent, 2021. [Online; accessed 24-Januray-2025], `https://www.timvanerven.nl/blog/ftrl-vs-omd/`.

Taira Tsuchiya and Shinji Ito. Fast rates in stochastic online convex optimization by exploiting the curvature of feasible sets. In *Advances in Neural Information Processing Systems*, 2024.

Taira Tsuchiya, Shinji Ito, and Junya Honda. Stability-penalty-adaptive follow-the-regularized-leader: Sparsity, game-dependency, and best-of-both-worlds. *Advances in Neural Information Processing Systems*, 36:47406–47437, 2023.

Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

Andrew J Wagenmaker, Yifang Chen, Max Simchowitz, Simon Du, and Kevin Jamieson. First-order regret in reinforcement learning with linear function approximation: A robust estimation approach. *Proceedings of the 39th International Conference on Machine Learning*, 162:22384–22429, 2022.

Martin Wainwright. Variance-reduced Q-learning is minimax optimal, 2019a. Preprint, arXiv: 1906.04697.

Martin J. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge University Press, 2019b.

Tianhao Wang, Dongruo Zhou, and Quanquan Gu. Provably efficient reinforcement learning with linear function approximation under adaptivity constraints. In *Advances in Neural Information Processing Systems*, volume 34, pages 13524–13536, 2021.

Chen-Yu Wei and Haipeng Luo. More adaptive algorithms for adversarial bandits. *Proceedings of the 31st Conference On Learning Theory*, 75:1263–1291, 2018.

Gellert Weisz, Philip Amortila, Barnabás Janzer, Yasin Abbasi-Yadkori, Nan Jiang, and Csaba Szepesvari. On query-efficient planning in mdps under linear realizability of the optimal state-value function. In *Proceedings of 34th Conference on Learning Theory*, volume 134, pages 4355–4385, 2021.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Andreas Winkelbauer. Moments and absolute moments of the normal distribution, 2012. Preprint, arXiv: 1209.4340.

Chenjun Xiao, Ilbin Lee, Bo Dai, Dale Schuurmans, and Csaba Szepesvari. The curse of passive data collection in batch reinforcement learning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 8413–8438, 2022.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022.

Tengyang Xie and Nan Jiang. Q* approximation schemes for batch reinforcement learning: A theoretical comparison. In *Conference on Uncertainty in Artificial Intelligence*, pages 550–559, 2020.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 27395–27407, 2021.

Nuoya Xiong, Zhaoran Wang, and Zhuoran Yang. A general framework for sequential decision-making under adaptivity constraints, 2023. Preprint, arXiv: 2306.14468.

Zhihan Xiong, Maryam Fazel, and Lin Xiao. Dual approximation policy optimization, 2024. Preprint arXiv:2410.01249.

Zhenghao Xu, Xiang Ji, Minshuo Chen, Mengdi Wang, and Tuo Zhao. Sample complexity of neural policy mirror descent for policy optimization on low-dimensional manifolds. *Journal of Machine Learning Research*, 25(226):1–67, 2024.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011.

Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Comput. Surv.*, 55(1):1–36, 2021.

Rui Yuan, Simon S Du, Robert M Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *The 10th International Conference on Learning Representations*, 2022a.

Rui Yuan, Robert M. Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 3332–3380, 2022b.

Rui Yuan, Simon Shaolei Du, Robert M. Gower, Alessandro Lazaric, and Lin Xiao. Linear convergence of natural policy gradient methods with log-linear policies. In *The 11th International Conference on Learning Representations*, 2023.

Andrea Zanette. Exponential lower bounds for batch reinforcement learning: Batch rl can be exponentially harder than online rl. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 12287–12297, 2021.

Andrea Zanette. When is realizability sufficient for off-policy reinforcement learning? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 40637–40668, 2023.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent Bellman error. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 10978–10989, 2020.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023.

Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. Variational policy gradient method for reinforcement learning with general utilities. In *Advances in Neural Information Processing Systems*, 2020a.

Ruiqi Zhang and Andrea Zanette. Policy finetuning in reinforcement learning via design of experiments using offline data. In *Advances in Neural Information Processing Systems*, 2023.

Zihan Zhang, Yuan Zhou, and Xiangyang Ji. Almost optimal model-free reinforcement learningvia reference-advantage decomposition. In *Advances in Neural Information Processing Systems*, volume 33, pages 15198–15207, 2020b.

Peng Zhao, Long-Fei Li, and Zhi-Hua Zhou. Dynamic regret of online Markov decision processes. *Proceedings of the 39th International Conference on Machine Learning*, 162:26865–26894, 2022.

Zhang Zihan, Yuhang Jiang, Yuan Zhou, and Xiangyang Ji. Near-optimal regret bounds for multi-batch reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Alexander Zimin and Gergely Neu. Online learning in episodic markovian decision processes by relative entropy policy search. *Advances in Neural Information Processing Systems*, 26, 2013.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning*, pages 928–936, 2003.

Constantin Zălinescu. On uniformly convex functions. *Journal of Mathematical Analysis and Applications*, 95:344–374, 1983.