Politecnico di Milano
Department of Electronics, Information and Bioengineering
Doctoral Programme In Information Technology

New Directions in Pre-Training for Reinforcement Learning

Doctoral Dissertation of:
Riccardo Zamboni

Supervisor:
Prof. Marcello Restelli

Tutor:
Prof. Nicola Gatti

The Chair of the Doctoral Program:
Prof. Luigi Piroddi

2025 – XXXVII Cycle

Acknowledgements

This dissertation is more of a byproduct of a collective effort rather than merely the outcome of my Ph.D. As such, I would like to spend a few words to thank the people who made it possible.

First and foremost, I thank my Ph.D. advisor, Marcello, for his supervision on many aspects of being a (good) researcher and for smoothly pointing me towards what, rather unexpectedly, turned out to be the main topic of my research. I also thank David Abel and Herke Van Hoof for accepting to review this dissertation and for providing extremely valuable comments.

I am also very grateful to have had the chance to rely on my brilliant co-authors, Alberto, Davide, Duilio, Mirco, Vincenzo. They brought extremely valuable contributions to the papers that resulted in this thesis.

I also thank the extremely talented researchers from whom I had the chance to take example, even though they might all smile out of humbleness, commenting on how they have not done much: Alberto, for his extremely valuable and patient advice since the very beginning; Amarildo, for his huge support and endurance in harsh conditions; Dave, for being kindly present and extremely supportive, and for highlighting, with just a few hints, what a great researcher looks like; Mirco, for becoming de facto my partner in crime and for having the endurance to bear me during all of our submissions.

I am grateful to have been able to share my time with some truly great office mates: Alberto^2, Alessandro, Alessio, Amarildo, Davide^2, Filippo, Luca, Gianluca, Giovanni, Marco, Martino, Matteo, Mirco, Paolo, Riccardo^2, Simone, Quokka, Vincenzo. A special thank you goes to the ones that drapped with me: even though I will not disclose their names to preserve their integrity, sharing this journey with them is probably what made it a great journey in the first place.

I also want to thank all the wonderful people I had the chance to meet during my visit to Edinburgh. In particular, Sam, who ended up inspiring the introduction; my office mates Kale-ab and Leo, for welcoming me and sharing their research and emotional space; and L, for entering the scene quite abruptly and becoming a main character shortly after.

I thank all my friends, the long-time, the more recent, and the once-long-time-then-lost-then-recent ones. While not being technically involved in this thesis, they made

the last few years working towards it far more enjoyable. I am particularly grateful to have been borne by my Community: Edoardo, Francesca, Grace, Maria, Simona, Vittorio. I cannot think back on the path that led to this thesis without bumping into them countless times. I am also immensely grateful to my family, Barbara, Sandro, Mattia, Nonna, and Nonno, for always supporting me unconditionally and in any way they could.

Finally, I want to thank Nicolò, for what I cannot speak, thereof I will be silent.

As I expected yet failed to resolve, these acknowledgements have been written in the midst of quite an emotional breakdown, which means I am likely forgetting to thank many who deserve to be named. However, I hope I have had or will have the chance to thank you anyway, if you joined me through these years. I thank you all.

Riccardo Zamboni
October 8th, 2025

Abstract

REINFORCEMENT LEARNING offers a powerful paradigm for solving sequential decision-making problems, particularly in environments characterized by complex dynamics, partial and noisy observations, or the need for coordinated behavior among multiple agents. Despite remarkable progress, contemporary Reinforcement Learning methods often struggle to acquire general-purpose behaviors that reliably transfer across tasks, especially in non-trivial and realistic settings.

In contrast, unsupervised pre-training has become a fundamental driver of generalization in non-sequential domains, as demonstrated by the success of Large Language Models trained on massive unlabeled corpora. Inspired by this paradigm, *unsupervised pre-training in Reinforcement Learning* has recently emerged as a promising approach to improve generalization across diverse tasks. This framework typically unfolds in two phases: an initial phase in which the agent interacts with the environment without any task-specific supervision, followed by a fine-tuning phase where the acquired knowledge is adapted to a specific downstream objective. This two-phase approach allows agents to first interact freely with the environment to acquire transferable knowledge, which can later be fine-tuned for specific downstream tasks.

Despite its promise, prior research on unsupervised pre-training in Reinforcement Learning has remained largely confined to simplified settings, often involving a single agent with full access to the environment's state, or focused narrowly on representation learning under partial observability. This thesis broadens this scope by building on the empirical success of *Maximum State Entropy* methods in fully observable, single-agent settings, and extending their applicability to more challenging and realistic domains, specifically those involving partial observability and multiple agents.

Thus, our approach to unsupervised pre-training is centered on the maximization of state entropy, with the goal of inducing policies that generate diverse and informative state distributions, even when the true state is hidden or distributed across agents. Through a combination of theoretical analysis and empirical validation, this work generalizes entropy-based objectives to complex scenarios, laying the groundwork for a principled and scalable framework for unsupervised pre-training. Ultimately, our goal is to support more scalable and general-purpose Reinforcement Learning systems applicable to real-world domains.

Sommario

Il Reinforcement Learning rappresenta un paradigma estremamente efficace per affrontare problemi decisionali sequenziali, soprattutto in contesti caratterizzati da dinamiche complesse, osservazioni parziali, o dalla necessità di comportamenti coordinati tra più agenti. Nonostante i notevoli progressi, gli approcci attuali al Reinforcement Learning faticano ancora a sviluppare comportamenti generali, in grado di trasferirsi con affidabilità da un compito all'altro, in particolare in scenari realistici. Al contrario, il pre-training non supervisionato si è affermato come un fattore determinante per la generalizzazione in domini non sequenziali, come dimostrato dal successo dei Large Language Models addestrati su vasti insiemi di dati non etichettati. Ispirandosi a questo paradigma, il *pre-training non supervisionato nel Reinforcement Learning* è emerso come una strategia promettente per favorire la generalizzazione su un ampio spettro di tasks. Tale approccio si articola generalmente in due fasi: una fase preliminare in cui l'agente interagisce liberamente con l'ambiente senza vincoli specifici, e una successiva fase di fine-tuning in cui le conoscenze acquisite vengono adattate a un obiettivo specifico. Questo schema a due fasi consente di esplorare l'ambiente in modo autonomo per acquisire conoscenze trasferibili, che possono poi essere raffinate in funzione di compiti mirati. Nonostante il suo potenziale, la ricerca sul pre-training non supervisionato nel Reinforcement Learning è rimasta finora confinata a contesti semplificati, spesso limitati a un singolo agente con piena osservabilità dello stato dell'ambiente. Questa tesi amplia tale prospettiva, facendo leva sul successo empirico dei metodi di *Maximum State Entropy* in scenari con un singolo agente e piena osservabilità, ed estendendone l'applicazione a contesti più complessi e realistici, caratterizzati da osservazioni parziali e dalla presenza di più agenti. L'approccio proposto si fonda dunque sulla massimizzazione dell'entropia degli stati, con l'obiettivo di indurre politiche capaci di generare distribuzioni di stati diversificate e informative, anche quando lo stato reale è nascosto o distribuito tra diversi agenti. Attraverso un'analisi teorica combinata a una solida validazione empirica, questo lavoro estende gli obiettivi basati sull'entropia a scenari complessi, ponendo le basi per un framework rigoroso e scalabile di pre-training non supervisionato. In ultima analisi, l'obiettivo è contribuire allo sviluppo di sistemi di Reinforcement Learning più generali e scalabili.

Contents

CHAPTER 1

Introduction

Let us imagine you have been invited to your friend's house for a board game night.[1] The game you are about to play is wildly complex; an intimidating mix of cards, a tangle of buffs and debuffs, and enough quests and side quests to make David Foster Wallace blush. Inevitably, there is *that* friend who insists you just dive in: "Don't worry, you'll get the hang of it as we go!" Fast forward thirty minutes, and you hear: "Oh yeah, that move *did* make sense back then... but now that you've done *that*... well." Not only is the whole situation deeply humbling, but, more importantly, it hinders your ability to learn.

As anyone who has lived through such chaos can confirm, a better approach might be to first explore the game without the pressure of winning. You could take time to understand how cards interact, how actions lead to consequences, and how the system behaves overall. If your friends are patient enough to let you play around without worrying about scoring points or advancing the quest, this more *unsupervised* approach to learning can significantly boost your understanding, and later make your efforts toward actually completing the quest far more effective.

Now, let us translate this little allegory into the more formal language of reinforcement learning. You, the learner, are called an *agent*. Playing a card is referred to as taking an *action*. The pawns, the board, the face-up cards, all of that is part of the *environment*. A particular configuration of the environment is a *state*. The current quest you are attempting to complete is a *task*, and the points you earn are *rewards*. In reinforcement learning, an agent interacts with its environment by choosing actions sequentially in order to maximize the cumulative reward, ultimately, learning how to play and win the game.

[1] The probability of the author being invited to a board game night is extremely low. This is, in fact, why this thesis exists and why it is about Reinforcement Learning.

This thesis is for those who would like to ensure their AI agent does not suffer the same confusion and humiliation, or more formally, those interested in enabling better generalization across tasks with limited supervision. While past work has explored how unsupervised learning can be framed within reinforcement learning, especially in simplified settings with a single agent and fully observable environments, we push the boundary further. This work investigates how unsupervised pre-training can be scaled to realistic scenarios: where other agents exist, the underlying conditions of the environment are hidden (i.e., under partial observability), or they must cooperate (i.e., in multi-agent settings).

We focus on encouraging agents to explore diverse and novel states as a proxy for acquiring useful experience. This strategy, known as *state entropy maximization*, is compelling for several reasons: it is conceptually simple, paves the way for a richer class of problems than standard RL, and remains practical by using the standard RL policy-learning pipeline. Nonetheless, this strategy becomes substantially more nuanced in the presence of partial observability or multiple agents. In such settings, what constitutes a "diverse" state becomes ambiguous, and exploration itself may require coordination or belief inference. One of the goals of this work is to shed light on these challenges and demonstrate that even seemingly straightforward methods become significantly more complex in realistic environments.

The contributions of this thesis span both theory and practice. We offer a theoretical characterization of unsupervised pre-training under these challenging conditions, and proposing practical, scalable solutions designed for richer, more realistic domains.

Reinforcement Learning

Reinforcement Learning [RL, Sutton, 2018] sits comfortably under the broad umbrella of Machine Learning (ML). Famously defined by Tom M. Mitchell as "the study of computer algorithms that allow computer programs to automatically improve through experience" [Mitchell, 1997], ML has at its heart a central character: a computer, or more specifically, an *artificial agent*, learning from experience. That is, data.

What sets RL apart from its ML cousins is its *sequential* nature. Unlike classification or regression tasks, where a model makes one-shot predictions, in RL each decision affects not only the immediate outcome but also the *future* data that the agent will collect. This means that learning is not just about understanding the world, it is also about influencing how you get to interact with it next. In short, learning shapes experience, and experience shapes learning, in an ongoing feedback loop.

The term *reinforcement* might suggest something soft and fuzzy, perhaps a treat for a well-behaved puppy. In reality, it has its roots in behavioral psychology, where it is defined as "a consequence applied that will strengthen an organism's future behaviour whenever that behaviour is preceded by a specific antecedent stimulus" [Skinner, 1938, Schultz, 2015]. Translated into AI terms, this simply means the agent receives feedback, a scalar signal known as a *reward*, which helps it understand how well it is doing at a given task. RL agents are goal-driven, and this goal is usually framed as maximizing some form of utility function. That function is often, quite simply, the sum of rewards collected over time.

For such a seemingly straightforward framework, RL is surprisingly powerful. One of its greatest strengths is its *environment-agnosticism*: you do not need to tell the agent how the environment works. Instead, it can learn through trial and error, making it especially valuable for tasks where explicit models are hard to come by. This quality has fueled RL's recent explosion across practical applications: robotic locomotion [Haarnoja et al., 2018a, Smith et al., 2022] and autonomous driving [Kiran et al., 2021, Cusumano-Towner et al., 2025], video games [Mnih et al., 2013, 2015, Silver et al., 2016, Berner et al., 2019, Wurman et al., 2022], industrial robotics [Meyes et al., 2017, Gu et al., 2017], manipulation [Akkaya et al., 2019, Andrychowicz et al., 2020, Lu et al., 2022], nuclear-reactor control [Duval et al., 2024], strategic multi-agent coordination [Samvelyan et al., 2019, Vinyals et al., 2019], planning under uncertainty [Brown and Sandholm, 2019, Perolat et al., 2022], and even economic behaviour like trading, bartering and diplomacy [Johanson et al., 2022, Bakhtin et al., 2022].

But RL comes with a catch. It leans heavily on the existence of a well-defined reward feedback: a precise, scalar signal that tells the agent how it is doing. This assumption is so central that it has its own name: the *reward hypothesis*, articulated by Sutton and Littman as the idea that "all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)" [Sutton, 2004]. This hypothesis has sparked plenty of debate and empirical inquiry, with Bowling et al. [2023] being among the most notable recent efforts to test its limits.

In this thesis, we take a different stance. We ask:

What happens when the reward signal goes missing entirely?

Rather than treating the absence of reward as a bug in the system, we embrace it. We explore what RL can still accomplish when there is no task to optimize, no score to chase, and no "good job" signal at the end of a move.

Unsupervised Reinforcement Learning

Ideally, reward is a natural part of the system at hand, an emergent signal that flows organically from the task itself. In practice, however, it rarely shows up uninvited. Instead, reward functions are often handcrafted, requiring meticulous design choices to ensure the agent does something even remotely useful. This process is tedious, brittle, and deeply task-specific. As a result, Reinforcement Learning becomes unnecessarily constrained: each new task demands a new design, and generalisation across tasks remains limited at best.

Supervised learning, by contrast, has built an impressive toolbox for tackling the generalisation problem. A particularly fruitful strategy has been *unsupervised pre-training*, where models are first trained without supervision, then fine-tuned on a specific downstream task. This simple yet powerful recipe lies at the heart of recent breakthroughs in generative models for images [Ramesh et al., 2022], video [Singer et al., 2022], and most famously, language [Brown et al., 2020b, Alayrac et al., 2022]. In these domains, learning from scratch every time is not just inefficient, it is old fashioned. As argued in Agarwal et al. [2022], real-world applications call for methods that can leverage prior experience, not ignore it.

Which brings us to RL: can it enjoy the same benefits from unsupervised pre-training? The question is still open, but the framework of *unsupervised Reinforcement*

Learning, formally articulated in Laskin et al. [2021] (though its roots go back to ideas in Hazan et al. [2019], Mutti and Restelli [2020]), attempts to answer it directly.

The intuition is simple: rather than chasing a reward from the start, the agent is set free to explore the environment without any specific goal, no score to maximise, no task to complete. This is the pre-training phase, where the agent's job is simply to "solve the environment", in the sense of acquiring reusable knowledge, structure, and behaviours. What this knowledge looks like can vary: it could be a compact set of policies that span relevant behaviours [Mutti et al., 2022c], a compressed abstract state representation [Agarwal et al., 2020], a model of environmental dynamics [Jin et al., 2020], a strategy that ensures thorough exploration [Hazan et al., 2019], or even just a rich dataset of diverse interactions [Yarats et al., 2022]. Indeed, the idea of using unsupervised signals to boost RL dates back to longer-term contributions [Barto et al., 2004, Oudeyer and Kaplan, 2007, Little and Sommer, 2013, Schmidhuber, 2010], but the unsupervised RL framework was the one to provide a clean and concise way to think about it in a pre-training and fine-tuning paradigm.

Once this unsupervised pre-training is complete, the second phase begins: the agent is finally given a task, and its success depends on how well the pre-trained model helps it perform. This is *supervised fine-tuning*, where we evaluate the utility of all that prior unsupervised effort, be it via planning with a learned model, or collecting samples using a pre-trained policy.

In this thesis, however, we zoom in on the first act. Our focus lies squarely on the unsupervised pre-training itself, with fine-tuning considered mainly as a means of evaluation. In particular, we will explore one prominent and conceptually appealing instantiation of unsupervised pre-training: state entropy maximisation.

State Entropy Maximization

Within the unsupervised pre-training paradigm, state entropy maximisation offers a particularly elegant and compelling objective. The idea is deceptively simple: learn a policy that induces a state distribution as entropic as possible [Hazan et al., 2019]. In other words, we ask the agent to roam far and wide, reaching many different parts of the environment with high probability. Yet, this is not the same as randomly pressing buttons. A policy that samples actions uniformly at random in every state might look "exploratory", but it completely ignores the sequential nature of RL. Reaching some states requires very particular chains of decisions, not just noise. This is why state entropy maximisation is a non-trivial and deeply RL-specific challenge.

What makes this objective unsupervised is its indifference to the environment's feedback. There is no reward signal guiding behaviour, just an intrinsic desire to visit as many different states as possible. Learning such a policy is far from easy, but the benefits are significant. In the context of offline RL, the importance of collecting diverse data,also known as coverage,has been shown to be fundamental [Levine et al., 2020, Antos et al., 2008, Chen and Jiang, 2019, Foster et al., 2021, Jin et al., 2021b, Zhan et al., 2022]. And the story does not stop there: in online RL, coverage can accelerate learning [Xie et al., 2022], and in fine-tuning scenarios, a well-initialised policy with broad state coverage can lead to better downstream performance [Xie et al., 2021]. Even representation learning and reward inference benefit: a state entropy-maximising policy helps uncover the structure of the environment more efficiently [Tarbouriech et al., 2021, Jin et al., 2020].

Of course, state entropy is not a silver bullet. It does not always align with every downstream goal, for instance, coverage as formalised via concentrability coefficients [Antos et al., 2008] may require different kinds of exploratory behaviour. But unlike many theoretically sound exploration criteria, state entropy maximisation has a powerful redeeming quality: it is practical.

This thesis will argue that, despite its abstract appeal, state entropy maximisation can in fact be implemented in realistic environments. We will show that it remains feasible even when the agent sees only noisy, partial observations or when there is more than one agent in play. What looks intuitive on paper turns out to be far from straightforward in these richer settings. But that is precisely the point: by tackling these cases, we aim to understand not just what state entropy maximisation is in theory, but what it can become in practice.

Contributions

This thesis offers a systematic extension of unsupervised pre-training in Reinforcement Learning to decision-making settings that go beyond the classic single-agent, fully observable case. In single-agent settings, the problem has been extensively studied and is largely understood. A popular approach, first introduced by Hazan et al. [2019] as *state entropy maximization*, has shown remarkable empirical success [Zisselman et al., 2023] and has also been framed within the broader convex Reinforcement Learning framework [Hazan et al., 2019, Zhang et al., 2020a], in which agents optimize convex utility functions. However, this framework was recently shown to suffer from a potentially harmful mismatch between theoretical tractability and practical effectiveness [Mutti et al., 2022a].

Nonetheless, key aspects, ranging from problem formulation, structural properties, algorithm design, to the effect of pre-training in more complex scenarios, remained open. To fill this gap, we first extend the convex Reinforcement Learning framework to partially observable settings [Åström, 1965], i.e., settings in which the agent lacks direct access to the environment's true state. We then describe how state entropy maximization can be meaningfully extended to such settings. We characterize the fundamental limitations of directly maximizing entropy over raw observations through performance bounds and propose an entropy-regularized algorithm to address this issue in practice, accounting for the mismatch introduced by partial observability. Subsequently, we explore a more contemporary approach by learning belief or latent representations. We characterize both the potential and, more importantly, the risks associated with such representations through the notion of *hallucinations*, and we propose an algorithm that leverages these representations while actively counteracting their drawbacks. These results appeared in Zamboni et al. [2024a,b].

We then turn our attention to environments with multiple agents, such as Markov Games [Littman, 1994a], or parallel Markov Decision Processes [Sucar, 2007]. We first characterizing alternative formulations, highlighting their respective advantages and limitations. We extend results from single-agent convex Reinforcement Learning to these settings, showing how these problems, while theoretically tractable, remain non-trivial in practice. We introduce a scalable, decentralized algorithm to address these challenges and demonstrate that policy pre-training via state entropy maximization, when done properly, offers surprising benefits in terms of accelerated learning and zero-shot performance. These results, to some extent, corroborate findings in single-agent environments [Zisselman et al., 2023], and appeared in Zamboni et al. [2025b], De Paola et al. [2025].

Additionally, we offer a novel perspective on Maximum Entropy methods by exploring their utility in another corner of the RL landscape: representation learning. We introduce a new algorithm that combines the maximum entropy principle with ideas from distributional Reinforcement Learning [Bellemare et al., 2017, 2023] to learn representations that are both informative and useful for downstream tasks. These considerations were first presented in Zamboni et al. [2023].

Overview

Chapter 2 provides a primer on sequential decision-making and Reinforcement Learning, introducing the more complex models that will be central to our investigation, ranging from partially observable and multi-agent settings to those characterized by convex utilities. **Chapter 3** offers a bird's-eye view of the unsupervised RL framework: we present the general problem formulation and survey a (deliberately non-exhaustive) portion of the relevant literature. Special attention is given to the state entropy maximization formulation, which remains central throughout the thesis. In **Chapter 4**, we explore how state entropy maximization can be adapted to partially observable settings. **Chapter 5** extends the discussion to environments with multiple agents. **Appendix A** contains an additional example demonstrating the utility of Maximum Entropy methods for representation learning. Finally, **Appendix B** provides all omitted formal results and proofs from the main chapters, while **Appendix C** includes additional experimental details for the practical methodologies presented.

Notation

Throughout this thesis, we adopt the following notational conventions. For any integer $N < \infty$, we define $[N] := \{0, 1, \dots, N-1\}$ and, more generally, $[n : N] := \{n, n+1, \dots, N-1\}$. Sets are denoted using calligraphic letters, e.g., $\mathcal{A}$, and their cardinality by $|\mathcal{A}|$. The T-fold Cartesian product of a set $\mathcal{A}$ is denoted $\mathcal{A}^T := \times_{t\in[T]}\mathcal{A}$. The probability simplex over $\mathcal{A}$ is denoted by $\Delta_{\mathcal{A}} := \{p \in [0,1]^{|\mathcal{A}|} : \sum_{a\in\mathcal{A}} p(a) = 1\}$, and the space of conditional distributions from $\mathcal{A}$ to another set $\mathcal{B}$ is written as $\Delta_{\mathcal{A}}^{\mathcal{B}}$.

Given vectors $v = (v_1, \dots, v_T)$ and $u = (u_1, \dots, u_T)$, we denote by $v \oplus u := (v_1, u_1, \dots, v_T, u_T)$ their concatenation and $v^\top, u^\top$ their transpose. A random vector of dimension T is denoted as $\mathbf{x} = (x_1, \dots, x_T)$, with $\mathbf{x}[t]$ indicating the entry at index $t \in [T]$, or simply x_t when clarity allows. For a vector $v \in \mathbb{R}^N$, we define its infinity norm as $\|v\|_\infty := \max_{i\in[N]} v_i$. Similarly, for a matrix $\mathbb{V} \in \mathbb{R}^{N\times M}$, we denote the infinity norm as $\|\mathbb{V}\|_\infty := \max_{(i,j)\in[N]\times[M]} |V_{ij}|$, its conjugate transpose as $\mathbb{V}^*$, and its Hadamard (element-wise) inverse as $\mathbb{V}^{\circ-1}$, where $(\mathbb{V}^{\circ-1})_{ij} := 1/V_{ij}$ for all i, j. The vectors of eigenvalues and singular values of $\mathbb{V}$ are denoted $\lambda(\mathbb{V})$ and $\sigma(\mathbb{V})$, respectively. The spectral norm is given by $\|\mathbb{V}\|_2 := \sqrt{\lambda_{\max}(\mathbb{V}^*\mathbb{V})} = \sigma_{\max}(\mathbb{V})$, where $\lambda_{\max}(\mathbb{V}) := \|\lambda(\mathbb{V})\|_\infty$ and $\sigma_{\max}(\mathbb{V}) := \|\sigma(\mathbb{V})\|_\infty$.

For probability distributions p_1 and p_2 over the same domain, we denote their total variation distance by $d_{\mathrm{TV}}(p_1, p_2) := \frac{1}{2}\sum_{x\in\mathcal{X}} |p_1(x) - p_2(x)|$, and their Kullback–Leibler divergence by $D_{\mathrm{KL}}(p_1\|p_2) := \sum_{x\in\mathcal{X}} p_1(x) \log \frac{p_1(x)}{p_2(x)}$. Let X be a random variable with support $\mathcal{X}$ and associated distribution p_X. The Rényi entropy of order α is defined as

$\mathcal{H}_\alpha(p_X) = \frac{1}{1-\alpha}\log\left(\sum_{x\in\mathcal{X}} p_X(x)^\alpha\right)$. From this, we recover the Shannon entropy as the limit $\mathcal{H}(X) := \lim_{\alpha\to 1}\mathcal{H}_\alpha(X) = -\sum_{x\in\mathcal{X}} p_X(x)\log p_X(x)$, and also the min-entropy as $\mathcal{H}_\infty(X) := \lim_{\alpha\to\infty}\mathcal{H}_\alpha(X) = -\log\left(\max_{x\in\mathcal{X}} p_X(x)\right)$. Finally, for any differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we denote its gradient as $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$.

CHAPTER 2

Reinforcement Learning

This chapter introduces the foundational concepts of RL that serve as the backbone for the remainder of this dissertation. Our aim is not to exhaustively detail every nuance of the field, but rather to provide a focused summary of the core ideas necessary to understand and contextualize the subsequent contributions.

We begin by outlining the key models that formalize decision-making under uncertainty, including the widely adopted Markov Decision Processes and their generalizations. We then present a high-level classification of RL algorithms, highlighting their main structural features and practical considerations. Finally, we discuss more recent directions aimed at addressing complex scenarios, like learning with partial observations, multiple agents and under convex utility functions.

In the following chapter, we will leave out several important aspects of RL that fall outside the scope of this thesis but might be of interest for particularly curious readers. For a comprehensive treatment of sequential decision-making and the theory of Markov Processes, we refer such readers to Puterman [2014]. For an accessible and more modern overview of RL, we recommend the excellent monographs [Sutton, 2018, Agarwal et al., 2019, Szepesvári, 2022].

2.1 A Primer on Sequential Decision-Making

In this section, we introduce the fundamental concepts underlying the framework of sequential decision-making and establish the notation that will be used throughout the thesis.

2.1.1 Introduction to Markov Processes

We begin by defining a *Controlled Markov Process* (CMP) as a (stochastic) system whose dynamics can be influenced through the execution of actions. Formally, a CMP is described by the tuple

$$\mathcal{M}_{T \vee \gamma} := (\mathcal{S}, \mathcal{A}, \mathbb{P}, (T \vee \gamma), \mu), \tag{2.1}$$

where $\mathcal{S}$ denotes the *state space*, with (finite) cardinality $S := |\mathcal{S}|$, and $\mathcal{A}$ is the *action space*, with size $A := |\mathcal{A}|$. The function $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ defines the *transition model*, such that $P(s'|s, a)$ represents the probability of reaching state s' after executing action a in state s. The process starts from an initial state s_0 sampled from a distribution $\mu \in \Delta_{\mathcal{S}}$. Finally, the dynamics of the system may either terminate after a finite number of steps $T < \infty$ (finite horizon), or continue under the presence of a discount factor $\gamma \in [0, 1]$, such that the probability of the process ending at the next step is $1 - \gamma$.

We use the notation $(T \vee \gamma)$ to indicate that either a time horizon or a discount factor is specified, but not both. In the former case, the process is said to be *episodic*, while in the latter it is *discounted* if $\gamma < 1$, and *undiscounted* when $\gamma = 1$.

The defining characteristic of any Markovian process is the *Markov Property*, which states that the probability of transitioning to a future state depends only on the current state and action, and not on the sequence of states and actions that preceded it. This property is reflected in the form of the transition model $\mathbb{P}(s'|s, a)$.

In RL, we are primarily interested in those CMPs that are endowed with a feedback mechanism in the form of scalar rewards. These are known as *Markov Decision Processes* [MDPs, Puterman, 2014], and are defined as:

$$\mathcal{M}^R_{T \vee \gamma} := (\mathcal{S}, \mathcal{A}, \mathbb{P}, R, (T \vee \gamma), \mu), \tag{2.2}$$

where $(\mathcal{S}, \mathcal{A}, \mathbb{P}, (T \vee \gamma), \mu)$ is a CMP and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function assigning a finite reward $R(s, a)$ to each state-action pair.[1]

An MDP unfolds through the interaction between an agent and the environment. At time step $t = 0$, the agent begins in an initial state $s_0 \sim \mu$. At each time step t, it selects an action a_t, receives a reward $R(s_t, a_t)$, and transitions to a new state $s_{t+1} \sim \mathbb{P}(\cdot|s_t, a_t)$. In the episodic setting, this process repeats for a fixed number of steps T, until the episode ends. In the discounted setting, interaction proceeds for a random number of steps T_γ, where $T_\gamma - 1 \sim \text{Geo}(1 - \gamma)$. In the undiscounted setting ($\gamma = 1$), the process continues indefinitely.

Each episode generates a trajectory consisting of a state sequence $\mathbf{s} = (s_0, \dots, s_{T-1}) \in \mathcal{T}^T_{\mathcal{S}} \subseteq \mathcal{S}^T$, an action sequence $\mathbf{a} = (a_0, \dots, a_{T-1}) \in \mathcal{T}^T_{\mathcal{A}} \subseteq \mathcal{A}^T$, and the corresponding state-action sequence $\mathbf{sa} = (s_t, a_t)_{t \in [T]} \in \mathcal{T}^T_{\mathcal{SA}} \subseteq \mathcal{S}^T \times \mathcal{A}^T$.[2]

2.1.2 Policy Classes

In RL, the agent's behavior is encoded by a *policy*, denoted as π, which governs the selection of actions during interaction with an environment. A policy is defined as a sequence of decision rules $\pi := (\pi_t)_{t \in [T]}$, where each rule π_t maps past state trajectories $\mathbf{s} \in \mathcal{T}^t_{\mathcal{S}}$ to a probability distribution over actions. Formally, $\pi_t : \mathcal{T}^t_{\mathcal{SA}} \to \Delta_{\mathcal{A}}$, and the

[1] Reward functions can also be defined solely over states, or as functions of (s, a, s').

[2] In the discounted setting, T_γ replaces T, while for undiscounted cases, the sequences are infinite.

conditional probability of taking action a after observing $\mathbf{sa}$ is denoted $\pi_t(a|\mathbf{sa})$. The full policy space is denoted Π, and the subset of deterministic policies, where each π_t maps deterministically to a single action, is denoted Π^{D}.

We distinguish between different classes of policies. *Non-Markovian* (NM) policies, denoted Π^{NM}, allow decision rules to depend on the entire history of visited states. In contrast, *Markovian* (M) policies, denoted Π^{M}, are time-invariant and memoryless, consisting of a constant decision rule $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$ applied at every step.

When a CMP is coupled with a Markovian policy $\pi \in \Pi^{\mathrm{M}}$, the resulting system forms a *Markov Chain* with a state-to-state transition model defined as:

$$\mathbb{P}^\pi(s'|s) := \sum_{a\in\mathcal{A}} \pi(a|s)\mathbb{P}(s'|s,a). \tag{2.3}$$

2.1.3 Induced Distributions

The repeated interaction of an agent with an environment under a given policy π induces a sequence of probability distributions over states. Specifically, the state distribution at time t is given by:

$$d_t^\pi(s) := \Pr(s_t = s \mid \pi), \tag{2.4}$$

which evolves recursively according to the *flow equation*:

$$d_t^\pi(s) = \sum_{s'\in\mathcal{S}} d_{t-1}^\pi(s')\mathbb{P}^\pi(s \mid s'). \tag{2.5}$$

For undiscounted processes and under standard regularity conditions [Puterman, 2014], this sequence converges to a *stationary state distribution*, defined as:

$$d_\infty^\pi(s) := \lim_{t\to\infty} d_t^\pi(s). \tag{2.6}$$

In discounted processes with $\gamma < 1$, the policy π induces a *discounted state distribution*, defined as the weighted sum of the time-indexed state distributions:

$$d_\gamma^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t d_t^\pi(s), \tag{2.7}$$

which satisfies the recursive identity:

$$d_\gamma^\pi(s) = (1-\gamma)\mu(s) + \gamma\sum_{s'\in\mathcal{S}} d_\gamma^\pi(s')\mathbb{P}^\pi(s \mid s'). \tag{2.8}$$

In the episodic case with finite horizon T, the policy induces a *marginal state distribution*:

$$d_T^\pi(s) := \frac{1}{T}\sum_{t\in[T]} d_t^\pi(s), \tag{2.9}$$

where the starting distribution is $d_0^\pi(s) = \mu(s)$.

Beyond state-wise distributions, a policy π also induces a probability distribution over entire trajectories. In the episodic setting, the joint probability of a state-action trajectory $\mathbf{sa} = (s_0, a_0, \dots, s_{T-1}, a_{T-1}) \in \mathcal{T}_{\mathcal{SA}}^T$ is given by:

$$p^\pi(\mathbf{sa}) = \mu(s_0)\prod_{t\in[T]} \pi(a_t \mid s_t)\mathbb{P}(s_{t+1} \mid s_t, a_t). \tag{2.10}$$

Moreover, a single trajectory s induces an empirical state distribution $d \in \Delta_{\mathcal{S}}$ via:

$$d(s \mid \mathbf{s}) := \frac{1}{T} \sum_{t \in [T]} \mathbf{1}(\mathbf{s}[t] = s). \tag{2.11}$$

We will slightly abuse notation and denote by p^{π} the probability of sampling this empirical distribution under policy π. When collecting n independent trajectories from π, the resulting empirical distribution is:

$$d_n(s) := \frac{1}{n} \sum_{k \in [n]} d_k(s), \tag{2.12}$$

and its sampling distribution is denoted p_n^{π}. Accordingly, we define the expected state distribution induced by π as:

$$d^{\pi}(s) := \mathop{\mathbb{E}}_{d_n \sim p_n^{\pi}} [d_n(s)]. \tag{2.13}$$

2.1.4 Performance Indexes, Value Functions and Solution Concepts

As previously introduced, the objective of a RL agent interacting with a MDP is to maximize the long-term accumulation of rewards. The term "long-term" reflects the agent's far-sighted nature: rather than focusing solely on immediate rewards, the agent considers future outcomes as well. This notion is formalized through the concept of a *value function*, which depends on the structure of the underlying MDP.

In an episodic MDP $\mathcal{M}_T^R$, the value function $V_t^{\pi}(s)$ represents the expected cumulative reward obtained by starting in state s at time step t and following a policy π until the end of the episode. Formally, this is defined as:

$$V_t^{\pi}(s) := \mathop{\mathbb{E}}_{\pi, \mathcal{M}_T^R} \Big[\sum_{t' \in [t:T]} R(s_{t'}, a_{t'}) \,\Big|\, s_t = s \Big]. \tag{2.14}$$

The corresponding state-action value function $Q_t^{\pi}(s, a)$ captures the expected return from taking action a in state s at time t and thereafter following policy π. This can be expressed as:

$$Q_t^{\pi}(s, a) := R(s, a) + \mathop{\mathbb{E}}_{s' \sim \mathbb{P}(\cdot|s,a)} \Big[V_{t+1}^{\pi}(s') \Big]. \tag{2.15}$$

For a discounted MDP $\mathcal{M}_{\gamma}^R$ with discount factor $\gamma \in [0, 1)$, the value function is modified to include the discounting of future rewards. The state value function is given by:

$$V_{\gamma}^{\pi}(s) := \mathop{\mathbb{E}}_{\pi, \mathcal{M}_{\gamma}^R} \Big[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\Big|\, s_0 = s \Big], \tag{2.16}$$

while the corresponding state-action value function becomes:

$$Q_{\gamma}^{\pi}(s, a) := R(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim \mathbb{P}(\cdot|s,a)} \big[V_{\gamma}^{\pi}(s') \big]. \tag{2.17}$$

Value functions serve as performance indexes conditioned on the choice of initial state and play a fundamental role in RL. They are primarily used for *policy evaluation*, i.e., assessing the performance of a given policy. However, for the purpose of *policy optimization*, i.e., improving or discovering high-performing policies without access to the environment model, the state-action value function becomes central as it contains more granular information about the decision process. Additionally, the *advantage function* is defined as the difference between the state-action value and the value function under policy π:

$$A^{\pi}_{t\vee\gamma}(s,a) = Q^{\pi}_{t\vee\gamma}(s,a) - V^{\pi}_{t\vee\gamma}(s). \tag{2.18}$$

Intuitively, it quantifies how much better (or worse) taking action a in state s is compared to the average expected return from that state.

An important property of value functions in MDPs is their recursive nature. For example, in the discounted setting, the value of a state is expressed in terms of the values of successor states through the relation:

$$V^{\pi}_{\gamma}(s) := \mathop{\mathbb{E}}_{a\sim\pi(\cdot|s)}\Big[Q^{\pi}_{\gamma}(s,a)\Big], \tag{2.19}$$

where we have assumed $\gamma < 1$.[3]

This recursive property leads to the *Bellman Expectation Equation* [Bellman, 1952], which defines the value function as:

$$V^{\pi}_{\gamma}(s) = \mathop{\mathbb{E}}_{a\sim\pi(\cdot|s)}\Big[R(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim\mathbb{P}(\cdot|s,a)}\big[V^{\pi}_{\gamma}(s')\big]\Big]. \tag{2.20}$$

The above relation can be interpreted as the application of the *Bellman Expectation Operator* T^{π} to a generic function $f : \mathcal{S} \rightarrow \mathbb{R}$, which is defined as:

$$(T^{\pi}f)(s) = \mathop{\mathbb{E}}_{a\sim\pi(\cdot|s)}\Big[R(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim\mathbb{P}(\cdot|s,a)}\big[f(s')\big]\Big]. \tag{2.21}$$

Since the agent seeks to maximize long-term return, we also introduce the *Bellman Optimality Operator*, denoted as $T^{\star}$, which incorporates a maximization over actions:

$$(T^{\star}f)(s) = \max_{a\in\mathcal{A}}\Big[R(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim\mathbb{P}(\cdot|s,a)}\big[f(s')\big]\Big]. \tag{2.22}$$

With these definitions in place, we define the performance objective for an MDP $\mathcal{M}^{R}_{T\vee\gamma}$ as:

$$\mathcal{J}_{\mathcal{M}^{R}_{T\vee\gamma}}(\pi) := \mathop{\mathbb{E}}_{s\sim\mu}\Big[V^{\pi}_{0\vee\gamma}(s)\Big], \tag{2.23}$$

where the value function used depends on whether the MDP is episodic ($\gamma = 0$) or discounted.

An alternative but equivalent formulation of the objective considers the distribution over trajectories. In episodic MDPs, the *return function* G_t maps a trajectory $\mathbf{sa}$ to the cumulative reward collected up to time t:

$$G_t(\mathbf{sa}) := \sum_{t'\in[t]} R(s_{t'}, a_{t'}). \tag{2.24}$$

[3] The interested reader can refer to Puterman [2014] for the corresponding recursion in episodic MDPs.

In the discounted case, the return is defined with discounting:

$$G_\gamma(\mathbf{sa}) := \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t). \tag{2.25}$$

The expected return over trajectories sampled from a policy then yields the same objective:

$$\mathcal{J}_{\mathcal{M}^R_{T\vee\gamma}}(\pi) := \mathop{\mathbb{E}}_{\mathbf{sa}\sim p^\pi}\Big[G_{T\vee\gamma}(\mathbf{sa})\Big]. \tag{2.26}$$

Now, *solving* an MDP consists of identifying a policy in a pre-specified policy space Π that maximizes this objective. The problem is thus formalized as:

$$\max_{\pi\in\Pi} \mathcal{J}_{\mathcal{M}^R_{T\vee\gamma}}(\pi). \tag{2.27}$$

It is known from [Puterman, 2014, Theorem 5.5.3] that there exists an optimal policy $\pi^\star \in \Pi \subseteq \Pi^{\mathrm{D}} \cap \Pi^{\mathrm{M}}$ that is deterministic and Markovian, such that:

$$\pi^\star \in \arg\max_{\pi\in\Pi} \mathcal{J}_{\mathcal{M}^R}(\pi). \tag{2.28}$$

This definition of optimality is relative to the initial state distribution μ, and is often referred to as *initial-state optimality*. Stronger notions, such as *uniform optimality*, require that the policy maximizes the value function for every state in $\mathcal{S}$.

2.1.5 Exact Solution Methods

In this context, we briefly review various approaches for deriving the optimal policy, assuming full knowledge of the MDP. The core idea is to first compute the optimal value function and then derive the corresponding optimal policy. While complete knowledge of the environment's dynamics is rarely available in practical applications, these algorithms are of fundamental importance, as they form the basis for many value-based RL methods discussed later.

Dynamic Programming

Dynamic Programming [DP, Bellman, 1952] can be integrated into sample-based methods and is also applicable to MDPs with infinite state or action spaces. The first DP algorithm we consider is policy iteration [Sutton, 2018], which alternates between two phases: policy evaluation and policy improvement. The policy evaluation phase computes the value function corresponding to the current policy, which is initially chosen at random. Notably, this step relies on the fact that the value function of a policy is the unique fixed point of the Bellman Expectation Operator of Eq. (2.20). The policy improvement phase, based on the policy improvement theorem [Sutton, 2018], guarantees an improvement by greedifying the policy with respect to the current value function. The pseudocode for the algorithm is provided below.

Algorithm 1: Policy Iteration (**PI**)

```
Input: Randomly initialized policy $\pi_0$
for $k = 0, \ldots,$ until convergence do
    Randomly initialize $V_0(s), \forall s \in \mathcal{S}$
    for $j = 0, \ldots,$ until convergence do
        Apply operator $V_{j+1}(s) = (T^{\pi_k} V_j)(s), \forall s \in \mathcal{S}$
    end for
    Compute $Q^{\pi_k}(s, a) = R(s, a) + \gamma \, \mathbb{E}_{s' \sim P(\cdot|s,a)}[V_j(s')], \forall (s, a) \in \mathcal{S} \times \mathcal{A}$
    Compute the greedy policy $\pi_{k+1}(s) \in \arg\max_{a \in \mathcal{A}} Q^{\pi_k}(s, a), \forall s \in \mathcal{S}$
end for
Output: Optimal policy $\pi_i$
```

It is important to note that the inner-loop policy evaluation step can be computationally expensive. The value iteration algorithm [Sutton, 2018] addresses this issue by performing only a single application of the Bellman Expectation Operator before greedifying the result. This method relies on the Bellman Optimality Equation (2.22) and its associated operator. In particular, repeatedly applying the Bellman Optimality Operator converges to the optimal value function. Hence, value iteration iteratively applies this operator to a randomly initialized function to obtain the optimal value. The pseudocode is provided below.

Algorithm 2: Value Iteration (**VI**)

```
Input: Function $f_0 : \mathcal{S} \to \mathbb{R}$ with random initialization.
for $k = 0, \ldots,$ until convergence do
    Apply operator $f_{k+1}(s) = (T^* f_k)(s), \forall s \in \mathcal{S}$
end for
Compute $\pi^*(s) \in \arg\max_{a \in \mathcal{A}} \left\{ R(s, a) + \gamma \, \mathbb{E}_{s' \sim P(\cdot|s,a)}[f_k(s')] \right\}$
Output: Optimal policy $\pi^*$
```

Linear Programming

The solution of finite MDPs can also be formulated as a linear program [LP, Wang et al., 2007]. In the discounted case ($\gamma < 1$), the primal LP problem is:

$$
\begin{aligned}
\min_{v \in \mathbb{R}^{|\mathcal{S}|}} \quad & \sum_{s \in \mathcal{S}} \nu_0(s) v(s) \\
\text{subject to} \quad & v(s) \geqslant R(s, a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}(s' \mid s, a) v(s'), \quad \forall s \in \mathcal{S}, \; \forall a \in \mathcal{A}, \\
& \nu_0 \in \Delta_{\mathcal{S}}.
\end{aligned}
$$

This results in a LP with $|\mathcal{S}|$ variables and $|\mathcal{S}||\mathcal{A}|$ constraints. The solution of the primal yields the optimal value function V^*, from which an optimal policy can be recovered by selecting the greedy action with respect to V^*. Applying Lagrangian duality, the dual LP is formulated as in Wang et al. [2007]:

$$\begin{aligned}
\max_{\nu \in \Delta_{\mathcal{S}\times\mathcal{A}}} \quad & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \nu(s,a)R(s,a) \\
\text{subject to} \quad & \sum_{a\in\mathcal{A}} \nu(s',a) = (1-\gamma)\mu(s') + \gamma \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} \nu(s,a)\mathbb{P}(s' \mid s,a), \quad \forall s' \in \mathcal{S}.
\end{aligned}$$

The dual formulation involves $|\mathcal{S}||\mathcal{A}|$ variables and $|\mathcal{S}|$ constraints (excluding non-negativity constraints), and is generally more practical to solve. The optimal solution ν^* corresponds to the γ-discounted stationary state-action distribution compatible with the initial state distribution μ and an optimal policy. An optimal policy can then be recovered as:

$$\pi^*(a|s) = \frac{\nu^*(s,a)}{\sum_{a'\in\mathcal{A}} \nu^*(s,a')}. \tag{2.29}$$

2.2 Reinforcement Learning Algorithms

All the methods for solving MDPs that we have previously discussed crucially rely on full knowledge of the transition model $\mathbb{P}$. However, in many relevant decision-making problems, the transition model is either unknown, such as the laws governing human behavior, or too complex to be explicitly represented, as in many robotics applications. This is where RL becomes essential, as it enables the agent to learn an (approximately) optimal policy through mere (sampled) interactions with the MDP.

In the following sections, we briefly present some of the most common RL algorithms, following a widely adopted taxonomy: *critic-only*, *actor-based*, and *actor-critic* methods. Critic-only (or value-based) methods aim to learn an optimal value function and subsequently derive the optimal policy via greedification [Watkins and Dayan, 1992, Rummery and Niranjan, 1994, Munos, 2005, Scherrer, 2014]. Actor-only (or policy-based) methods, by contrast, directly optimize the policy without explicitly modeling the value function [Williams, 1992, Baxter and Bartlett, 2001]. Finally, actor-critic approaches [Konda and Tsitsiklis, 1999, Lillicrap et al., 2016] combine the previous two, maintaining both an explicit policy (the actor) and a value function corresponding to the current policy (the critic). For a more comprehensive treatment, we refer the reader to standard textbooks [Sutton, 2018, Szepesvári, 2022].

2.2.1 Critic-Only Methods

In general, *critic-only* methods fall under the umbrella of approximate dynamic programming, a family of algorithms that blend exact DP solutions with function approximation techniques. The idea is to represent value functions within a functional space $\mathcal{F}$, enabling the algorithm to handle large or even infinite state spaces effectively, provided that $\mathcal{F}$ is suitably chosen. The objective is to find a function $f \in \mathcal{F}$ that accurately approximates the optimal value or action-value function.

The most well-known critic-only method is *Q-learning* [Watkins and Dayan, 1992], which learns the optimal action-value function through sampled interactions with the MDP. The core of the algorithm is the *Q-learning update*, which uses the most recent interaction (s_t, a_t, s_{t+1}) to refine the estimate of the state-action value function Q_t. The learning rate $\alpha \in \mathbb{R}^+$ controls the trade-off between retaining previous estimates and incorporating new information. Crucially, under proper learning-rate control and mild assumptions on the behavioral policy β, the algorithm is guaranteed to *asymptotically converge* to the optimal value function Q^*, and consequently to the optimal policy π^*. Importantly, at any given time, the estimate Q_t may not correspond to the value function of any actual policy, as the update uses a maximization over possible actions. This makes the algorithm *off-policy*: the data is generated by a behavioral policy β that can differ from the policy implicitly induced by Q_t. The algorithm is outlined below.

Algorithm 3: Q-learning

```
Input: Learning rate $\alpha$, behavioral policy $\beta$
Initialize $Q_0(s, a)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ (e.g., randomly)
Sample initial state $s_0 \sim \mu$
for $t = 0, \ldots$ until convergence do
    Sample action $a_t \sim \beta(\cdot|s_t)$
    Collect reward $R(s_t, a_t)$ and next state $s_{t+1} \sim P(\cdot|s_t, a_t)$
    Q-value update:
        $Q_{t+1}(s_t, a_t) = (1 - \alpha)Q_t(s_t, a_t) + \alpha \left(R(s_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q_t(s_{t+1}, a')\right)$
    (Optional) Update behavioral policy $\beta$
end for
Greedification step:
    $\pi^*(s) \in \arg\max_{a \in \mathcal{A}} Q_t(s, a), \quad \forall s \in \mathcal{S}$
Output: Optimal policy $\pi^*$
```

When the MDP has an infinite number of states (as in continuous control tasks), storing a table of SA Q-values becomes impractical. A common solution is to use a function approximator for the Q-function, typically a parametric differentiable function Q_ϕ defined by a parameter vector $\phi \in \mathbb{R}^d$. That is, the functional space becomes $\mathcal{F} = \{Q_\phi : \phi \in \mathbb{R}^d\}$. In this setting, the Q-learning update is replaced by:

$$\phi_{t+1} \leftarrow \phi_t + \alpha \left(R(s_t, a_t) + \gamma \max_{a \in \mathcal{A}} Q_\phi(s_{t+1}, a) - Q_\phi(s_t, a_t)\right) \nabla_\phi Q_\phi(s_t, a_t).$$

This rule can be interpreted as a stochastic gradient descent step minimizing the mean squared error. However, convergence guarantees are limited: they hold under restrictive assumptions, such as linear function approximators [Melo and Ribeiro, 2007], although these were relaxed more recently [Carvalho et al., 2020]. For overparameterized ReLU neural networks, finite-sample convergence bounds were also established [Xu and Gu, 2020]. Despite limited theoretical guarantees, the combination of Q-learning with deep neural networks, known as *Deep Q-Networks* (DQNs), has shown remarkable empirical success [Mnih et al., 2015]. This coupling, however, introduces several challenges that require a number of modifications to stabilize and enhance the learning process.

A key issue is that samples in RL are not i.i.d., since consecutive states are temporally correlated. To mitigate this, DQNs use *experience replay*, where observed transitions are stored in a *replay buffer*. At each update step, transitions are sampled (uniformly) from this buffer to approximate the gradient of the squared TD error. In practice, the target is treated as fixed to avoid differentiating through the max operator:

$$\begin{aligned} f(\phi) &:= \frac{1}{n} \sum_{i \in [n]} \left(R_i + \gamma \max_{a \in \mathcal{A}} Q_\phi(s'_i, a) - Q_\phi(s_i, a_i) \right)^2 , \\ \nabla_\phi f(\phi) &= -\frac{1}{n} \sum_{i \in [n]} \left(R_i + \gamma \max_{a \in \mathcal{A}} Q_\phi(s'_i, a) - Q_\phi(s_i, a_i) \right) \nabla_\phi Q_\phi(s_i, a_i). \end{aligned}$$

The replay buffer is typically a finite-size FIFO queue. Once full, older transitions are replaced by newly collected samples obtained from an exploration-driven policy. A common choice is ϵ-greedy exploration, with ϵ decaying over time, though alternatives like Boltzmann exploration are also used.

To further stabilize learning, DQNs often employ a *target network* that is updated more slowly than the main Q-network. This prevents the learning targets from shifting too rapidly. Moreover, double Q-learning [Hasselt, 2010] is often used to reduce overestimation bias, leading to the Double DQN algorithm [Van Hasselt et al., 2016]. Additional improvements include *prioritized experience replay* [Schaul et al., 2015], where transitions are sampled with priority based on their mean squared error, and *dueling architectures* [Wang et al., 2016], which separately estimate state values and advantages. A comprehensive empirical study of these enhancements and their combinations was conducted in Hessel et al. [2018].

2.2.2 Actor-Only Methods

The approaches discussed thus far were value-based, meaning they rely on an intermediate step involving the estimation of a value function, from which a greedy policy is subsequently extracted. However, this intermediate step often becomes impractical when the action space is large or continuous, as the greedification step can be computationally expensive or even intractable. In such cases, *Policy Optimization* (PO) methods become particularly relevant. These methods explicitly represent the policy in an appropriate space Π. Importantly, this explicit policy representation enables the imposition of structural or behavioral constraints, which often proves beneficial in practice.

A wide range of policy optimization strategies have been proposed in the literature, including model-based techniques [Ng and Jordan, 2013, Ko and Fox, 2009], expectation-maximization algorithms [Kober and Peters, 2008], variational inference approaches [Neumann, 2011], and evolutionary computation methods [Heidrich-Meisner and Igel, 2009]. Typically, PO methods adopt a trajectory-based formulation of the objective. When the state and action spaces of the MDP are large or continuous, an efficient non-parametric representation of the policy π is generally not feasible. In such cases, the policy space is modeled via a set of parametric policies $\Pi_\Theta = \{\pi_\theta : \theta \in \Theta \subseteq \mathbb{R}^q\}$. The policy optimization problem then becomes

$$\max_{\pi_\theta \in \Pi_\Theta} J_{\mathcal{M}^R}(\pi_\theta), \tag{2.30}$$

where the objective is directly optimized over the policy parameter space Θ. This is often approached using first-order methods such as gradient ascent. In fact, it is relatively straightforward to derive the gradient of the objective $J_{\mathcal{M}^R}(\pi_\theta)$ with respect to the policy parameters [Kober and Peters, 2008], yielding the well-known *Policy Gradient Theorem* [Sutton et al., 1999]:

$$\nabla_\theta J_{\mathcal{M}^R}(\pi_\theta) = \mathop{\mathbb{E}}_{\mathbf{sa}\sim p^{\pi_\theta}} \left[\sum_{t\in[T]} \nabla_\theta \log \pi_\theta(a_t|s_t) Q^{\pi_\theta}(s_t, a_t) \right]. \tag{2.31}$$

Based on this key result, policy optimization can be performed through gradient ascent using (Monte Carlo) estimates of Eq. (2.31). The resulting algorithm is known as G(PO)MDP [Baxter and Bartlett, 2001], and we report its pseudo-code below.

Due to its ability to handle continuous action spaces and its conceptual simplicity, this algorithmic blueprint, more generally referred to as *Policy Gradient* (PG), has seen widespread adoption. However, a notable inefficiency of the base PG algorithm is that it discards collected experience after a single update. As a result, new samples must be collected afresh for each update, making PG methods inherently online. This is clearly suboptimal, as the same batch of samples could, in principle, be reused to perform multiple updates.

Algorithm 4: G(PO)MDP

```
Input: learning rate $\alpha$, parameter space $\Theta$
Initialize $\theta_0 \in \Theta$ (randomly)
for $i = 0, \dots,$ until convergence do
    Collect $\{\mathbf{sa}_j\}_{j=1}^N$, i.e., a batch of trajectories, using policy $\pi_{\theta_i}$
    Estimate the policy gradient:
        $\widehat{\nabla}_{\theta_i} J_{\mathcal{M}_R}(\pi_{\theta_i}) = \sum_{j\in[N]} \sum_{t\in[T]} \nabla_{\theta_i} \log \pi_{\theta_i}(a_t^{(j)}|s_t^{(j)}) \left( \sum_{k\in[t:T]} R(s_k^{(j)}, a_k^{(j)}) \right)$
    Update the parameters:
        $\theta_{i+1} = \theta_i + \alpha \widehat{\nabla}_{\theta_i} J_{\mathcal{M}_R}(\pi_{\theta_i})$
end for
Output: $\pi_{\theta_i}$
```

2.2.3 Actor-Critic Methods

The Actor-Critic framework combines value-based and policy-based methods by applying a form of generalised policy iteration over two main components: the *actor*, which is a parameterized policy $\pi_\theta(a|s)$, and the *critic*, which is a parameterized estimator of a value function. The critic can take the form of an action-value function $Q_\phi(s, a)$, a state-value function $V_\phi(s)$, or an advantage function $A_\phi(s, a)$. These two components are trained in tandem, either synchronously or asynchronously. Despite its conceptual simplicity, this architecture has proven highly effective in practice, particularly when combined with deep function approximation, and Actor-Critic algorithms have emerged as some of the most widely used methods in reinforcement learning.

The original Actor-Critic architecture dates back to Sutton et al. [1999], who proposed leveraging the Policy Gradient Theorem (2.31) to design a learnable critic. The key insight was that the true action-value function $Q^{\pi_\theta}(s,a)$ can be replaced by a learned approximation $Q_\phi(s,a)$ without introducing bias, provided the approximation satisfies the compatibility condition. This framework is known as *compatible function approximation*, and it ensures that the resulting gradient estimate remains unbiased. The conditions are: the critic $Q_\phi(s,a)$ must be linear in the score function, i.e., the gradient of the log-policy: $Q_\phi(s,a) = \nabla_\theta \log \pi_\theta(a|s)^\top w$, where w is a vector of weights; the weights w must minimise the mean squared error between the true advantage function and its linear approximation

$$w^* = \arg\min_{w} \mathop{\mathbb{E}}_{s,a\sim d^{\pi_\theta}} \left[\left(A^\pi(s,a) - \nabla_\theta \log \pi_\theta(a|s)^\top w \right)^2 \right]. \tag{2.32}$$

When both conditions are satisfied, the approximated value function $Q_\phi(s,a)$ can be safely used in the policy gradient:

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{s,a\sim d^{\pi_\theta},\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s)\, Q_\phi(s,a) \right], \tag{2.33}$$

which forms the theoretical foundation of the Actor-Critic architecture and enables the construction of stable and efficient policy gradient algorithms.

Building upon these foundations, Peters and Schaal [2008a] introduced the use of *natural gradients*, initially proposed by Amari [1998]. Instead of performing standard gradient ascent, natural policy gradient methods adjust the update direction by the inverse of the Fisher information matrix:

$$\tilde{\nabla}_\theta J(\theta) = F^{-1} \nabla_\theta J(\theta), \tag{2.34}$$

where

$$F = \mathop{\mathbb{E}}_{s,a\sim d^{\pi_\theta}} \left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^\top \right] \tag{2.35}$$

is the Fisher information matrix. Natural gradients account for the geometry of the policy space and are empirically found to yield more stable and sample-efficient updates. This gave rise to the family of *Natural Actor-Critic* algorithms.

Later, Schulman et al. [2015] employed a parametric policy and an estimator of the advantage to build the *Trust Region Policy Optimization* (TRPO) algorithm, which improves policies with guaranteed monotonic improvement in expected return. TRPO optimizes a surrogate objective while constraining the KL divergence between the new and old policies:

$$\begin{aligned} \max_\theta \quad & \mathop{\mathbb{E}}_{s,a\sim d^{\pi_\text{old}}} \left[\frac{\pi_\theta(a|s)}{\pi_\text{old}(a|s)} \hat{A}^{\pi_\text{old}}(s,a) \right] \\ \text{s.t.} \quad & \mathop{\mathbb{E}}_{s\sim d^{\pi_\text{old}}} \left[D_\text{KL} \left(\pi_\text{old}(\cdot|s) \,\|\, \pi_\theta(\cdot|s) \right) \right] \leqslant \delta, \end{aligned}$$

where $\hat{A}^{\pi_\text{old}}(s,a)$ is an estimator of the advantage under the old policy and δ is a hyperparameter controlling the size of the policy update.

Although TRPO enjoys strong theoretical guarantees and strong empirical performance, its implementation can be complex due to its reliance on second-order optimization. To simplify this, Schulman et al. [2017] introduced *Proximal Policy Optimization* (PPO), which approximates TRPO's trust region with a clipped surrogate objective:

$$L^{\text{CLIP}}(\theta) = \mathop{\mathbb{E}}_{t}\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{2.36}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the estimated advantage, and ϵ is a small hyperparameter (e.g., 0.1-0.3). PPO's simplicity and empirical performance have made it a standard in deep RL.

In parallel, Silver et al. [2014] developed the *Deterministic Policy Gradient* (DPG) Theorem, which enables direct optimization of deterministic policies $\mu_\theta(s)$. The deterministic policy gradient is:

$$\nabla_\theta J(\mu_\theta) = \mathop{\mathbb{E}}_{s\sim d^{\mu_\theta}}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu_\theta}(s,a)\big|_{a=\mu_\theta(s)}\right]. \tag{2.37}$$

This formulation enables off-policy learning using a replay buffer. When extended with deep networks, this leads to the *Deep Deterministic Policy Gradient* (DDPG) algorithm [Lillicrap et al., 2016], effective in high-dimensional continuous control.

Another significant development was Mnih et al. [2016]'s introduction of the *Asynchronous Advantage Actor-Critic* (A3C) algorithm. Here, the policy gradient uses the advantage function in place of the full Q-value:

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{s,a\sim d^{\pi_\theta}}\left[\nabla_\theta \log \pi_\theta(a|s)\, A^\pi(s,a)\right]. \tag{2.38}$$

The advantage is estimated via a value function baseline $V^\pi(s)$, trained using temporal-difference learning. A3C employs multiple parallel agents to stabilise training and improve data throughput.

Finally, Haarnoja et al. [2018b] introduced the *Soft Actor-Critic* (SAC) algorithm, which augments the reward with an entropy term to encourage exploration. The objective is:

$$J(\pi) = \mathop{\mathbb{E}}_{(s_t,a_t)\sim d^{\pi_\theta}}\left[r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))\right], \tag{2.39}$$

where α is a temperature parameter that balances reward and entropy. The resulting policy gradient is:

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{s_t,a_t\sim d^{\pi_\theta}}\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\left(Q^\pi(s_t,a_t) - \alpha \log \pi_\theta(a_t|s_t)\right)\right]. \tag{2.40}$$

SAC achieves state-of-the-art performance on continuous control benchmarks and is noted for its robustness and sample efficiency.

2.3 The Frontiers of Reinforcement Learning

Throughout the previous sections, we have surveyed the core methodologies used to address MDPs. While these models offer a solid foundation for sequential decision-making, they are often too restrictive to capture the richness and complexity of real-world environments. This section explores some of the most promising and actively researched extensions of RL that move beyond the standard MDP framework. We focus in particular on models designed to operate under partial observability, where the agent receives only indirect or noisy information about the underlying state, as well as those that account for interactions between multiple decision-makers. Additionally, we briefly examine a broader class of decision-making problems in which the objective function is no longer required to be linear in the state distribution, relaxing a key structural assumption of traditional MDPs.

2.3.1 Markov Processes with Partial Observability

MDPs assume that the agent has full access to the underlying state of the environment. In practice, however, the agent may only receive partial, noisy, or indirect observations. A more general framework is required to handle such scenarios. Consider, for instance, an autonomous robot deployed in *rescue operations*. The robot operates in an unknown terrain with the goal of locating and assisting a wounded person. It cannot access its true position or the location of the human but instead perceives the environment through noisy sensors and cameras. In this context, naively maximizing the performance related to the observations is unlikely to be helpful, as it is usually defined over the true conditions of the system.

Such partially observable settings are indeed common in applications including robotics [Cassandra et al., 1996, Akkaya et al., 2019], resource allocation [Bower and Gilbert, 2005], medical diagnosis [Hauskrecht and Fraser, 2000], recommendation systems [Li et al., 2010], and business management [De Brito and Van Der Laan, 2009]. These are typically modeled using the framework of Partially Observable Markov Decision Processes [POMDPs, Åström, 1965], in which observations are stochastic functions of the hidden underlying state.

Interaction Protocol

A finite-horizon POMDP is defined as the tuple $\mathcal{M}^R := (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathbb{P}, \mathbb{O}, R, T, \mu)$, where $\mathcal{S}$ is the state space with cardinality $S = |\mathcal{S}|$, $\mathcal{A}$ is the action space with $A = |\mathcal{A}|$, and $\mathcal{O}$ is the observation space with $O = |\mathcal{O}|$. The transition kernel is given by $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$, with $\mathbb{P}(s' \mid s, a)$ representing the probability of transitioning to s' from s after taking action a. The observation function $\mathbb{O} : \mathcal{S} \to \Delta_{\mathcal{O}}$ defines the likelihood $\mathbb{O}(o \mid s)$ of receiving observation o when in state s. Rewards are governed by a function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The episode horizon is a finite integer $T < \infty$, and the initial state distribution is given by $\mu \in \Delta_{\mathcal{S}}$.

At the beginning of an episode, a state s_0 is drawn from the distribution μ. At each timestep $t < T$, the agent receives an observation o_t drawn from $\mathbb{O}(\cdot \mid s_t)$ and selects an action a_t. The environment transitions to a new state s_{t+1} drawn from $\mathbb{P}(\cdot \mid s_t, a_t)$. At the final time step $T - 1$, an observation o_{T-1} is sampled and the episode ends. Thus, each trajectory yields sequences of states $\mathrm{s} = (s_0, \ldots, s_{T-1})$, actions $\mathbf{a} = (a_0, \ldots, a_{T-1})$, state-action pairs $\mathbf{sa} = (s_t, a_t)_{t\in[T]}$, and observations $\mathbf{o} = (o_0, \ldots, o_{T-1})$.

Beliefs, Policies, and Distributions

Since the agent cannot directly observe the true states, it often reasons over a *belief* [Kaelbling et al., 1998], which is a probability distribution over the possible states given the sequence of previous observations and actions. We denote this belief as $b \in \mathcal{B} \subseteq \Delta_{\mathcal{S}}$. Typically, the initial belief is uniform, that is, $b_1 = \mathcal{U}(\mathcal{S})$. Beliefs are updated recursively via Bayes' rule. Given an action a and subsequent observation o at time t, the updated belief is computed as

$$b_t^{ao}(s) = \frac{\mathbb{O}(o \mid s) \sum_{s'\in\mathcal{S}} \mathbb{P}(s \mid s', a) b_{t-1}(s')}{\sum_{\tilde{s}\in\mathcal{S}} \mathbb{O}(o \mid \tilde{s}) \sum_{s''\in\mathcal{S}} \mathbb{P}(\tilde{s} \mid s'', a) b_{t-1}(s'')}. \tag{2.41}$$

This expression defines a belief transition operator $\mathcal{T}^{ao} : \mathcal{B} \to \mathcal{B}$ such that the next belief $b' = \mathcal{T}^{ao}(b)$. In this way, beliefs evolve through a trajectory $\boldsymbol{b} = (b_0, \ldots, b_{T-1})$.

Let $i \in \mathcal{I}$ denote the information available to the agent at each time step. In POMDPs, a policy $\pi : \mathcal{I} \to \Delta_{\mathcal{A}}$ defines the action selection strategy, where $\pi(a \mid i)$ is the probability of selecting action a given information i. Depending on the modeling choice, the information space $\mathcal{I}$ can correspond to a single observation from $\mathcal{O}$, a sequence of past observations from $\mathcal{T}_{\mathcal{O}}$, or a trajectory of beliefs from $\mathcal{T}_{\mathcal{B}}$. This design allows the definition of stationary Markov policies with respect to the information space, even when the process is non-Markovian in the original state or observation spaces.

A policy π induces a marginal distribution over states, denoted $d_{\mathcal{S}}^{\pi} \in \Delta_{\mathcal{S}}$, which is defined as $d_{\mathcal{S}}^{\pi}(s) = \frac{1}{T}\sum_{t\in[T]} \Pr(s_t = s)$. Similarly, it induces a marginal distribution over observations $d_{\mathcal{O}}^{\pi}(o) = \frac{1}{T}\sum_{t\in[T]} \Pr(o_t = o)$. These two distributions are linked by the observation function via the relationship

$$d_{\mathcal{O}}^{\pi}(o) = \sum_{s\in\mathcal{S}} d_{\mathcal{S}}^{\pi}(s)\mathbb{O}(o \mid s). \tag{2.42}$$

Let $\mathbf{s}, \mathbf{o}, \mathbf{a}$ represent random vectors corresponding to state, observation, and action sequences of length T. The full trajectory of states is distributed as $\mathbf{s} \sim p_{\mathcal{S}}^{\pi} \in \Delta_{\mathcal{S}^T}$, where $p_{\mathcal{S}}^{\pi}(\mathbf{s}) = \prod_{t\in[T]} \Pr(s_t = \mathbf{s}[t])$, and likewise the trajectory of observations is distributed as $\mathbf{o} \sim p_{\mathcal{O}}^{\pi} \in \Delta_{\mathcal{O}^T}$ with $p_{\mathcal{O}}^{\pi}(\mathbf{o}) = \prod_{t\in[T]} \Pr(o_t = \mathbf{o}[t])$.

Based on a trajectory $(\mathbf{s}, \mathbf{o})$, one can define empirical distributions over states and observations. The empirical state distribution is $d_{\mathcal{S}}(s \mid \mathbf{s}) = \frac{1}{T}\sum_{t\in[T]} \mathbf{1}(\mathbf{s}[t] = s)$, and the empirical observation distribution is $d_{\mathcal{O}}(o \mid \mathbf{o}) = \frac{1}{T}\sum_{t\in[T]} \mathbf{1}(\mathbf{o}[t] = o)$. The probability distributions $p_{\mathcal{S}}^{\pi}$ and $p_{\mathcal{O}}^{\pi}$ can also be used to denote the distributions over these empirical marginals. When averaging over n episodes, the empirical distributions become $d_{n,\mathcal{S}}(s) = \frac{1}{n}\sum_{k\in[n]} d_{k,\mathcal{S}}(s)$, and the corresponding probability of sampling such empirical distributions under policy π is denoted by $p_{n,\mathcal{S}}^{\pi}$. Analogous definitions hold for $d_{n,\mathcal{O}}$ and $p_{n,\mathcal{O}}^{\pi}$.

The full trajectory $\mathbf{h}$ includes states, actions, observations, and beliefs, and is defined as $\mathbf{h} = \mathbf{s} \oplus \mathbf{a} \oplus \mathbf{o} \oplus \boldsymbol{b}$. The probability of observing $\mathbf{h}$ under policy π is given by the expression

$$p^{\pi}(\mathbf{h}) = \mu(s_0) \prod_{t \in [T]} \mathbb{O}(o_t \mid s_t) \pi(a_t \mid i_t) \mathbb{P}(s_{t+1} \mid s_t, a_t) \mathcal{T}^{o_t a_t}(b_{t+1} \mid b_t). \tag{2.43}$$

Finally, the belief framework also allows us to define a trajectory of *believed states*, denoted $\tilde{\mathbf{s}} = (\tilde{s}_0, \ldots, \tilde{s}_{T-1})$, where each $\tilde{s}_t$ is drawn from the belief b_t. These believed trajectories are distributed according to

$$p(\tilde{\mathbf{s}} \mid \boldsymbol{b}) = \prod_{t \in [T]} b_t(\tilde{s}_t). \tag{2.44}$$

Solving Markov Processes under Partial Observability

In a POMDP, the agent's observations do not uniquely determine the underlying state of the environment. Since both rewards and transitions still depend on the true latent state, the observation alone does not constitute a Markovian signal. This fundamental non-Markovianity implies that mapping observations directly to actions is generally insufficient for optimal decision-making. Instead, solving POMDPs typically requires the agent to retain memory, reason about its belief over the possible states, and actively explore the environment to reduce uncertainty. These challenges make learning in partially observable settings considerably more complex. Despite these difficulties, modern RL systems have achieved notable successes in such domains, including games like Poker [Brown and Sandholm, 2019] and StarCraft [Vinyals et al., 2019]. Nonetheless, theoretical results confirm that learning and planning in POMDPs is statistically and computationally intractable in the general case, among others [Papadimitriou and Tsitsiklis, 1987, Mundhenk et al., 2000, Vlassis et al., 2012, Mossel and Roch, 2005, Krishnamurthy et al., 2016]. These hardness results, however, reflect worst-case scenarios. Recent work has identified rich subclasses of POMDPs, such as latent MDPs [Kwon et al., 2021b,a] and weakly revealing POMDPs [Liu et al., 2022a], where efficient learning algorithms are indeed possible and practically relevant.

A direct consequence of the non-Markovian nature of observations is that the agent would, in principle, need to remember the full history of actions and observations to make optimal decisions [Fact 3, Singh et al., 1994b]. However, maintaining such a growing history becomes impractical, especially over long time horizons. One common strategy to address this is to reframe the POMDP as a belief-state MDP.[4] In this formulation, the agent encodes all available information about the past into a belief state, a probability distribution over the latent states, which restores the Markov property and enables the use of standard RL algorithms. The resulting policy, denoted $\pi^*(b)$, maps beliefs to actions and can be evaluated through a belief-dependent value function $V^{\pi} : \Delta_{\mathcal{S}} \to \mathbb{R}$, defined as the expected discounted return under policy π starting from belief b:

$$V^{\pi}(b) = \mathop{\mathbb{E}}_{\pi} \left[\sum_{t \in [T_{\gamma}]} \gamma^t R(b_t, \pi(\cdot \mid b_t)) \,\middle|\, b_0 = b \right], \tag{2.45}$$

[4] The interested reader can refer to Appendix B.1 for a more detailed characterization.

where the expected reward at time t is computed as $R(b_t, \pi(\cdot \mid b_t)) = \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \pi(a \mid b_t) R(s, a) b_t(s)$. The optimal policy π^* is the one that maximizes V^π across all beliefs. Its value is given by the optimal value function V^*, which satisfies the Bellman optimality equation [Spaan, 2012]:

$$V^* = T^*_{\text{PO}} V^*, \tag{2.46}$$

where the Bellman backup operator T^*_{PO} applied to a generic function $f : \Delta_{\mathcal{S}} \to \mathbb{R}$ is defined as:

$$(T^*_{\text{PO}}) f(b) = \max_{a \in \mathcal{A}} \left[R(b, a) + \gamma \sum_{o \in \mathcal{O}} p(o \mid b, a) f(b^{ao}) \right], \tag{2.47}$$

with observation probabilities given by $p(o \mid b, a) = \sum_{s' \in \mathcal{S}} \mathbb{O}(o \mid s') \sum_{s \in \mathcal{S}} \mathbb{P}(s' \mid s, a) b(s)$. Interestingly, although the belief space is continuous, the optimal value function over beliefs is piecewise linear and convex [Spaan, 2012], a property that has been heavily exploited in the design of efficient solution methods, to name a few [Sondik, 1978, Cassandra et al., 1994, 1997, Zhang, 2001, 2010].

To mitigate worst-case complexity, numerous approximate methods have been developed. Many of these approaches focus on sampling or approximating the belief space to make planning tractable [Pineau et al., 2003, Spaan and Vlassis, 2005, Shani et al., 2007, Kurniawati et al., 2008, Poupart et al., 2011, Smith and Simmons, 2012]. However, constructing a belief state typically requires full knowledge of the environment's dynamics, which is not always available in practice. When the model is unknown, alternatives include learning memoryless policies that act directly on observations [Littman, 1994b, Jaakkola et al., 1994, Loch and Singh, 1998, Williams and Singh, 1999, Li et al., 2011], or using finite-state controllers that encode a compact internal memory structure [McCallum, 1993, Whitehead and Lin, 1995, Meuleau et al., 1999, Amato et al., 2010].

In recent years, deep RL techniques have enabled significant progress in handling partial observability by leveraging recurrent or memory-based architectures. A wide range of methods have incorporated neural networks to model value functions and policies, yielding strong empirical results across diverse tasks, among others [Bakker, 2002, Wierstra et al., 2007,?, Heess et al., 2015, Ha and Schmidhuber, 2018, Baisero and Amato, 2018, Igl et al., 2018, Zhang et al., 2019, Hafner et al., 2019]. While these methods are highly performant, they lack theoretical guarantees. An important step toward bridging this gap has been the recent introduction of approximate information states [Subramanian et al., 2022], which offer a formal approximation framework for belief tracking.

A distinct line of research avoids belief modeling altogether by using Predictive State Representations [PSR, Littman et al., 2002, Singh et al., 2003], which encode state using statistics over future observable sequences. PSRs provide a compelling alternative since they rely exclusively on observable quantities, and have been effectively used across various RL settings [Rosencrantz et al., 2004, Boots et al., 2011, Kulesza et al., 2015, Jiang et al., 2016].

2.3.2 Markov Processes with Multiple Agents

MDPs model scenarios involving a single agent interacting with an environment. However, in many real-world applications, multiple agents may interact with the same environment, often influencing one another. Consider, for instance, multiple robots being deployed in a collapsed building to carry out a rescue mission. Their objective is to explore a large area to find and save injured individuals. In some cases, effective rescue operations may require coordination, such as helping each other access otherwise unreachable areas. Requiring every robot to cover the entire area is clearly inefficient and unnecessary. On the other hand, if each robot acts purely independently, without any incentive to cooperate, opportunities for coordination may be lost, especially when cooperation comes at a cost. These kinds of scenarios are typically modeled using Markov Games [MGs, Littman, 1994a], where multiple agents interact with a shared environment.

Interaction Protocol

A finite-horizon MG is defined as $\mathcal{M}^R := (\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathbb{P}, R, \mu, T)$, where $\mathcal{N}$ is the set of agents; $\mathcal{S} = \times_{i\in[|\mathcal{N}|]}\mathcal{S}_i$ is the joint state space; $\mathcal{A} = \times_{i\in[|\mathcal{N}|]}\mathcal{A}_i$ is the joint action space, both $\mathcal{S}$ and $\mathcal{A}$ being discrete and finite; $\mu \in \Delta_{\mathcal{S}}$ is the initial state distribution; $\mathbb{P} \in \Delta^{\mathcal{S}}_{\mathcal{S}\times\mathcal{A}}$ is the transition model; $R = \{R^i\}_{i\in[|\mathcal{N}|]}$ is the collection of reward functions $R^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$; and $T < \infty$ is the time horizon.

At the beginning of each episode, the initial state s_0 is drawn from μ. Upon observing s_0, each agent i takes an action $a^i_0 \in \mathcal{A}_i$, resulting in a joint action $a_0 = (a^i_0)_{i\in[|\mathcal{N}|]}$. The system transitions to $s_1 \sim \mathbb{P}(\cdot|s_0, a_0)$, and each agent receives a reward according to $R^i(s_0, a_0)$. This process continues for $t < T$.

Policies and Distributions

Each agent follows a policy, which may be either Markovian, $\pi^i \in \Delta^{\mathcal{A}^i}_{\mathcal{S}}$, or non-Markovian, $\pi^i \in \Delta^{\mathcal{A}^i}_{\mathcal{S}^t\times\mathcal{A}^t}$. We denote the space of valid per-agent policies by Π^i and the space of joint policies by Π. Policies are said to be decentralized-information if conditioned only on an agent's local information (e.g., $\mathcal{S}_i$ or $\mathcal{S}^t_i \times \mathcal{A}^t_i$), and centralized-information if conditioned on the full state or joint histories. The joint policy is denoted by $\pi = (\pi^i)_{i\in[|\mathcal{N}|]} \in \Delta^{\mathcal{A}}_{\mathcal{S}}$.

Let S and S_i denote the random variables corresponding to the joint state and agent-i's state, respectively. Then, under policy π, we define the marginal state distributions as

$$d^\pi(s) = \frac{1}{T}\sum_{t\in[T]} \Pr(s_t = s|\pi, \mu), \tag{2.48}$$

$$d^\pi_i(s_i) = \frac{1}{T}\sum_{t\in[T]} \Pr(s_{t,i} = s_i|\pi, \mu). \tag{2.49}$$

Solving Markov Processes with Multiple Agents

As in the single-agent case, we can define value functions for any joint policy π composed of centralized per-agent policies. The value function and joint Q-function for agent i at time t are defined as

$$V_{i,t}^{\pi}(s) := \mathop{\mathbb{E}}_{\pi}\left[\sum_{t'\in[t:T]} R^i(s_{t'}, a_{t'}) \mid s_t = s\right] \tag{2.50}$$

$$Q_{i,t}^{\pi}(s,a) := \mathop{\mathbb{E}}_{\pi}\left[\sum_{t'\in[t:T]} R^i(s_{t'}, a_{t'}) \mid (s_t, a_t) = (s,a)\right]. \tag{2.51}$$

We further define the marginal Q-function for agent i as

$$Q_{i,t}^{\pi}(s,a_i) := \mathop{\mathbb{E}}_{\pi}\left[\sum_{t'\in[t:T]} R^i(s_{t'}, a_{t'}) \mid (s_t, a_{i,t}) = (s,a_i)\right], \tag{2.52}$$

which marginalizes over the actions of all other agents under π.

The Bellman operator for the marginal Q-function is

$$[\mathcal{T}_{i,t}^{\pi} f](s,a_i) := \mathop{\mathbb{E}}_{\substack{a_{-i}\sim\pi_{-i}(\cdot|s),\\ s'\sim P(\cdot|s,a),\\ a_i'\sim\pi_i(\cdot|s')}}\left[R^i(s,a) + f(s',a_i')\right], \tag{2.53}$$

for centralized Markov policies $\pi_i : \mathcal{S} \to \Delta_{\mathcal{A}_i}$ and $\pi_{-i} : \mathcal{S} \to \Delta_{\mathcal{A}_{-i}}$, possibly correlated. We define a best-response policy for agent i as $\pi^{\dagger}(\pi^{-i}) : \mathcal{S} \to \Delta_{\mathcal{A}_i}$ such that

$$V_{i,0}^{\pi^{i,\dagger}(\pi^{-i})\times\pi^{-i}}(s) = \sup_{\bar{\pi}^i} V_{i,0}^{\bar{\pi}^i\times\pi^{-i}}(s), \quad \forall s\in\mathcal{S}, \tag{2.54}$$

and we denote

$$V_i^{\dagger,\pi_{-i}}(s) := V_i^{\pi^{i,\dagger}(\pi_{-i})\times\pi_{-i}}(s), \quad V_i^{\dagger,\pi_{-i}}(\mu) := \mathop{\mathbb{E}}_{s\sim\mu}\left[V_i^{\dagger,\pi_{-i}}(s)\right]. \tag{2.55}$$

We are now ready to define the following solution concepts for Markov Games with centralized policies:

Definition 2.3.1 ((Markov) Equilibria)**.** *For $\epsilon > 0$, a (Markov) policy* [5] $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$ *is a (Markov) ϵ-Approximate Coarse Correlated Equilibrium (CCE) if*

$$\max_{i\in[|\mathcal{N}|]}\left\{V_i^{\dagger,\pi_{-i}}(\mu) - V_i^{\pi}(\mu)\right\} \leqslant \epsilon. \tag{2.56}$$

It is a (Markov) Coarse Correlated Equilibrium if $\epsilon = 0$. A product policy $\pi \in \Delta(\mathcal{A})^{\mathcal{S}}$ satisfying the same condition is called a (ϵ-Approximate) Nash Equilibrium (NE).

Numerous works have studied the computation and learning of such equilibria using centralized policies. Nash Equilibria have mostly been explored in games with favorable reward structure: identical, interest games (or more generally, potential games) with $R^i(\cdot) = R^j(\cdot)$ for all $i, j \in [|\mathcal{N}|]$ [Macua et al., 2018, Chen et al., 2022, Ding et al., 2022, Zhang et al., 2022b, Maheshwari et al., 2022, Fox et al., 2022, Leonardos et al., 2022, Alatur et al., 2023, Aydin and Eksin, 2023]; and two-player zero-sum

[5] Potentially a correlated policy.

MGs with $R^1(\cdot) = -R^2(\cdot)$ [Perolat et al., 2015, Daskalakis et al., 2020, Zhang et al., 2020b, Sayin et al., 2021, Wei et al., 2021, Huang et al., 2022, Cui and Du, 2022, Zeng et al., 2022, Pattathil et al., 2023, Yang and Ma, 2023, Arslantas et al., 2023, Cen et al., 2023, Cai et al., 2023, Chen et al., 2023], though with notable exceptions [Giannou et al., 2022, Kalogiannis and Panageas, 2023, Kalogiannis et al., 2023, Qin and Etesami, 2023, Sayin, 2023, Park et al., 2023]. In general Markov Games, computing Nash equilibria is intractable, and focus has shifted to coarse correlated equilibria [Jin et al., 2021a, Erez et al., 2023, Liu et al., 2022b, Daskalakis et al., 2023, Zhang et al., 2022a, Wang et al., 2023, Foster et al., 2023].

Just as with POMDPs, certain structured MGs permit more tractable solutions. For instance, polymatrix MGs with separable interactions allow for efficiently computable Nash equilibria [Kalogiannis and Panageas, 2023, Park et al., 2023], although stationary NE computation remains PPAD-hard [Daskalakis et al., 2023, Jin et al., 2023]. Stationarity is desirable in large-scale settings due to memory efficiency, particularly when leveraging deep neural policies. This aligns with the growing interest in networked MGs in MARL [Zhang et al., 2018, Chu et al., 2020, Zamboni et al., 2025a].

Gradient-based approaches have also received considerable attention. While policy gradient methods may fail to converge to general NEs even in simple classes of games [Vlatakis-Gkaragkounis et al., 2020], certain special cases such as strict equilibria have been shown to be attractors of the gradient dynamics [Giannou et al., 2022], particularly in adversarial team MGs [Kalogiannis et al., 2023], which generalize both zero-sum and potential games.

On the other hand, when considering decentralized policies, the literature is more scattered. With additional structural assumptions like separable interactions, Networked MGs [Lin et al., 2021, Qu et al., 2022, Zhou et al., 2023, Zhang et al., 2023, Jin et al., 2024] offer promising convergence results, contingent on the informativeness of the states available to each agent. These can be viewed as a subclass of Partially Observable MGs, which inherit the computational challenges of POMDPs. Even in the identical-interest case, known as Decentralized POMDPs, hardness results are well established [Nair et al., 2003, Bernstein et al., 2005, Oliehoek et al., 2008, Szer et al., 2012, Dibangoye et al., Liu et al., 2017, Amato et al., 2019], with some positive results under weakly-revealing structures [Liu et al., 2022b].

Nevertheless, a vast number of practical methods based on deep RL have been proposed to address these challenges. Among recent developments, trust-region-based policy optimization methods have shown remarkable empirical success [Yu et al., 2022], yet it remains an open question if the environments addressed in contemporary MARL do offer the usual challenges of Decentralized POMDPs [Tessera et al., 2025]. We refer to Albrecht et al. [2024] for a comprehensive review.

2.3.3 Markov Processes with Concave Utilities

As stated earlier, standard RL focuses on solving MDPs, where the utility is typically expressed as a linear combination of scalar reward terms. This is referred to as the *dual formulation*, in which the coefficients of this linear combination are given by the state distribution induced by the agent's policy [Puterman, 2014]:

$$\max_{\pi\in\Pi} \Big(R \cdot d^\pi\Big) =: \mathcal{J}_\infty(\pi), \tag{2.57}$$

here, $R \in \mathbb{R}^S$ is the reward vector, and d^π is the state distribution induced by π.

However, not all relevant objectives can be captured through this linear representation [Abel et al., 2021]. Several works have extended the standard RL formulation to address non-linear objectives of practical interest. These include imitation learning [Hussein et al., 2017, Osa et al., 2018], where the goal is to minimize the distance between the induced state distribution and the state distribution induced by expert demonstrations [Abbeel and Ng, 2004, Ho and Ermon, 2016, Kostrikov et al., 2019, Lee et al., 2020, Ghasemipour et al., 2020, Dadashi et al., 2020]; risk-averse RL [Garcıa and Fernández, 2015], where the objective accounts for the tail behavior of the agent's policy [Tamar and Mannor, 2013, Prashanth and Ghavamzadeh, 2013, Tamar et al., 2015, Chow et al., 2017, Bisi et al., 2020, Zhang et al., 2021b]; pure exploration [Hazan et al., 2019], where the aim is to maximize the entropy of the induced state distribution [Tarbouriech and Lazaric, 2019, Lee et al., 2020, Mutti and Restelli, 2020, Mutti et al., 2021, Zhang et al., 2021a, Guo et al., 2021, Liu and Abbeel, 2021b, Seo et al., 2021, Yarats et al., 2021, Mutti et al., 2022d,b]; diverse skill discovery [Gregor et al., 2017, Eysenbach et al., 2018, Hansen et al., 2019, Sharma et al., 2019, Campos et al., 2020, Liu and Abbeel, 2021a, He et al., 2022, Zahavy et al., 2022]; and constrained RL [Altman, 1999, Achiam et al., 2017, Brantley et al., 2020, Miryoosefi et al., 2019, Qin et al., 2021, Yu et al., 2021, Bai et al., 2022], among others.

Interaction Protocol

While these objectives may look very dissimilar at first glance, they can be viewed as different instances of a more general model, called *convex MDPs* [cMDPs, Zhang et al., 2020a, Zahavy et al., 2021, Geist et al., 2022]. A cMDP is defined as a tuple $\mathcal{M}^{\mathcal{F}} := (\mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{F}, (T \vee \gamma), \mu)$, where $(\mathcal{S}, \mathcal{A}, \mathbb{P}, (T \vee \gamma), \mu)$ is a classical CMP and $\mathcal{F}$ is a bounded *concave* utility function $\mathcal{F} : \Delta_{\mathcal{S}} \to \mathbb{R}$ with $F < \infty$,[6] and is a function of the state distribution d^π. The RL objective in cMDPs is then:

$$\max_{\pi \in \Pi} \Big(\mathcal{F}(d^\pi) \Big) =: \zeta_\infty(\pi). \tag{2.58}$$

The non-linearity of the concave utility breaks the additive structure of standard RL, invalidating classical Bellman equations. As a result, dynamic programming approaches become infeasible, requiring the development of novel methodologies. Fortunately, the problem remains largely tractable: it admits a dual formulation similar to standard RL [Puterman, 2014], and principled algorithms with sub-linear regret-only slightly worse than in classical RL have been developed [Zhang et al., 2020a, Zahavy et al., 2021].

Number of Trials Matters with Concave Utilities

Rather then focusing on how to solve the cMDP, we are first interested in the implications of the non-linearity of the objective on the intrinsic nature of the problem. In particular, we will show that the number of trials matters when optimizing the objective (2.58), a point first outlined in Mutti et al. [2022a,b] that will be of central interest in the subsequent sections.

[6] In practice, the function can be convex, concave, or even non-convex. The term "concave" is used to distinguish the objective from the standard (linear) RL setting. We assume $\mathcal{F}$ is concave unless otherwise stated.

In the standard RL setting, the objective $\mathcal{J}_\infty(\pi)$ evaluates the expected sum of rewards collected over an infinite number of episodes under policy π, as it depends on $d^\pi = \mathbb{E}_{d\sim p^\pi}[d]$. We then refer to problem (2.57) as the *infinite trials* RL formulation.

However, in practice, we cannot draw infinitely many episodes for any given policy. Instead, we collect a finite (and often small) batch $d_n \sim p_n^\pi$. This motivates a *finite trials* RL formulation that better reflects what is optimized in practice:

$$\max_{\pi\in\Pi} \left(\mathop{\mathbb{E}}_{d_n\sim p_n^\pi} \big[r \cdot d_n\big]\right) =: \mathcal{J}_n(\pi). \tag{2.59}$$

One might wonder whether optimizing the finite trials objective (2.59) leads to different results than the infinite trials one (2.57). In the standard RL case, the two objectives are in fact equivalent:

$$\mathcal{J}_n(\pi) = \mathop{\mathbb{E}}_{d_n\sim p_n^\pi} \big[R \cdot d_n\big] = R \cdot \mathop{\mathbb{E}}_{d_n\sim p_n^\pi} \big[d_n\big] = R \cdot d^\pi = \mathcal{J}_\infty(\pi), \tag{2.60}$$

since R is constant and expectation is a linear operator. Hence, both formulations yield the same optimal policies. This allows us to enjoy the computational tractability of the infinite trials formulation while optimizing the objective used in practice. However, as we will see, this equivalence does not hold in the convex RL setting.

The standard convex objective (2.58) has been introduced as $\zeta_\infty(\pi)$ to denote the infinite trials version of the convex RL objective. Analogously thought, we can define a finite trials version of the convex RL objective:

$$\max_{\pi\in\Pi} \left(\mathop{\mathbb{E}}_{d_n\sim p_n^\pi} \big[\mathcal{F}(d_n)\big]\right) =: \zeta_n(\pi). \tag{2.61}$$

Comparing (2.58) and (2.61), we note that both involve expectations over sampled state distributions. Specifically, we can write:

$$\zeta_\infty(\pi) = \mathcal{F}(d^\pi) = \mathcal{F}(\mathop{\mathbb{E}}_{d_n\sim p_n^\pi}[d_n]) \leqslant \mathop{\mathbb{E}}_{d_n\sim p_n^\pi}[\mathcal{F}(d_n)] = \zeta_n(\pi) \tag{2.62}$$

by Jensen's inequality. Thus, the equality does not hold in general. A policy optimized for infinite trials may be suboptimal when deployed under finite trials. The core reason is that $\mathcal{F}$ is applied after averaging in Eq. (2.58), while in Eq. (2.61), it is applied before. This subtle difference can lead to significant discrepancies in performance.

Despite this mismatch, most works continue to optimize (2.58), even when only a finite number of episodes are available. It is therefore important to assess how much is lost by approximating the finite trials objective with the infinite trials one. We begin by introducing a regularity assumption on $\mathcal{F}$:

Assumption 2.3.1 (Lipschitz). *A function $\mathcal{F} : \mathcal{A} \to \mathbb{R}$ is Lipschitz-continuous with constant $L < \infty$, or L-Lipschitz, if:*

$$|\mathcal{F}(x) - \mathcal{F}(y)| \leqslant L\|x - y\|_1, \ \forall (x, y) \in \mathcal{A}^2 \tag{2.63}$$

Mutti et al. [2022a] was then the first to characterize the approximation error by means of an upper bound:

Theorem 2.3.1 (Approximation Error [Mutti et al., 2022a]). *Let $n \in \mathbb{N}$ be the number of trials, $\delta \in (0, 1]$ a confidence level, $\pi^\dagger \in \arg\max_{\pi \in \Pi} \zeta_n(\pi)$, and $\pi^\star \in \arg\max_{\pi \in \Pi} \zeta_\infty(\pi)$. Then, with probability at least $1 - \delta$, it holds:*

$$err := \left|\zeta_n(\pi^\dagger) - \zeta_n(\pi^\star)\right| \leqslant 4LT\sqrt{\frac{2S\log(4T/\delta)}{n}} \tag{2.64}$$

This result provides an instance-agnostic upper bound on the approximation error: $err = O(LT\sqrt{S/n})$. As expected, the bound decreases at rate $O(1/\sqrt{n})$, reflecting the concentration of d_n around its expectation as n increases [Weissman et al., 2003]. Consequently, using the infinite trials objective in place of the finite one can be harmful in low-data regimes.

For example, when training a policy in simulation and deploying it in the real world-often with just one episode ($n = 1$), the observed performance may be significantly worse than the predicted $\zeta_\infty(\pi)$, leading to unsafe or undesirable behaviors. In this thesis, we will be particularly interested in the *finite trials* case, as this is the most realistic in almost any real-world application. Due to this, we will focus on algorithmic solutions able to address the finite trials objective directly.

Finally, note that although Theorem 2.3.1 gives an upper bound, it is not necessarily tight for every instance. However, the bound is informative in many practical applications, as thoroughly analyzed in Mutti et al. [2022a].

Reward is enough with Infinite Trials

While finite trials objectives will be our main focus, it is still useful to understand the properties of the infinite trials formulation and more specifically how to solve such problems. Fortunately enough, the following general result holds:

Lemma 2.3.2 (Sufficiency of Markovian Policies [Puterman, 2014]). *For any possibly non-Markovian policy $\pi \in \Pi^{\mathrm{NM}}$, define a stationary Markov policy $\pi' \in \Pi^{\mathrm{M}}$ as $\pi'(a|s) = \frac{d^\pi(s,a)}{d^\pi(s)}$. Then, $d^{\pi'} = d^\pi$.*

This result states that for any non-Markovian policy deployed in a (c)MDP, there exists a Markovian policy inducing the same state distribution. Since the infinite trials objective depends only on the state distribution, it follows that Markovian policies are indeed sufficient to optimize it. This is a powerful property, as it allows us to restrict our search to Markovian policies without loss of optimality.

Almost surprisingly, the previous fact is accompanied by another powerfull result, first described in Zahavy et al. [2021]: in cMDPs, the reward function is sufficient to optimize the infinite trials objective. More formally, they show that the infinite trials cRL objective (2.58) can be optimized by solving a sequence of standard RL problems with appropriately defined (non-stationary) reward functions.

In particular, this can be done by reformulating the cMDP as a zero-sum game between a "policy player" (the agent) and a "cost player" (who sets the rewards), by exploiting Fenchel duality:

Theorem 2.3.3 (Fenchel Duality for cMDPs [Zahavy et al., 2021]). *A cMDP $\mathcal{M}^{\mathcal{F}}$, with $\mathcal{F}$ being a* convex *utility function, can be solved via the following min-max problem:*

$$\min_{d^\pi \in \mathcal{V}} \mathcal{F}(d^\pi) = \min_{d^\pi \in \mathcal{V}} \max_{\lambda \in \Lambda} (\lambda \cdot d^\pi - \mathcal{F}^*(\lambda)) = \max_{\lambda \in \Lambda} \min_{d^\pi \in \mathcal{V}} (\lambda \cdot d^\pi - \mathcal{F}^*(\lambda)), \tag{2.65}$$

where

$$\mathcal{V} = \left\{ \nu \in \Delta_{\mathcal{S}\times\mathcal{A}} : \sum_{a\in\mathcal{A}} \nu(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s'\in\mathcal{S}, a'\in\mathcal{A}} \mathbb{P}(s|s',a')\nu(s',a'),\ \forall s \in \mathcal{S} \right\}$$

is the set of valid discounted state-action distributions, Λ is the closure of (sub-)gradient space $\{\partial\mathcal{F}(\nu)|\nu \in \mathcal{V}\}$ and $\mathcal{F}^*(\lambda) := \sup_\lambda \lambda \cdot d^\pi - \mathcal{F}(d^\pi)$ is the Fenchel conjugate.

With this, they define the Lagrangian as $\mathcal{L}(d^\pi, \lambda) := \lambda \cdot d^\pi - \mathcal{F}^*(\lambda)$: for a fixed $\lambda \in \Lambda$, minimizing the Lagrangian is then a standard RL problem, i.e., equivalent to maximizing a reward $R = -\lambda$[7] and the overall problem of maximizing a complex utility function $\mathcal{F}(\cdot)$ can be converted into solving a sequence of RL sub-problems characterized by different reward functions at each stage. In other words, any cMDP can be addressed by the following Algorithm:

Algorithm 2.2.3: Meta-Algorithm for cMDPs

```
Input: convex-concave Lagrangian $\mathcal{L} : \mathcal{V} \times \Lambda \to \mathbb{R}$, sub-routine algorithms
$\mathrm{Alg}_\lambda, \mathrm{Alg}_\pi$, number of iterations $K \in \mathbb{N}$
for $k = 1, \dots, K$ do
    $\lambda_k = \mathrm{Alg}_\lambda(d_1^\pi, \dots, d_{k-1}^\pi)$        ▷ Cost player update
    $d_k^\pi = \mathrm{Alg}_\pi(-\lambda_k)$        ▷ Policy player update
end for
Return: $\bar{d}_K^\pi = \frac{1}{K}\sum_{k=1}^K d_k^\pi$, $\bar{\lambda}_K = \frac{1}{K}\sum_{k=1}^K \lambda_k$.
```

The authors then analyze Algorithm 2.3.3 under the lenses of online convex optimization: the learner is presented with a sequence of K convex loss functions $\ell_1, \dots, \ell_K : \mathcal{V} \to \mathbb{R}$ and at each round k must select a point $x_k \in \mathcal{V}$ after which it suffers a loss of $\ell_k(x_k)$.[8] In this context, the loss functions for the cost player are $\ell_k^\lambda = -\mathcal{L}(\cdot, \lambda_k)$, and for the policy player are $\ell_k^\pi = \mathcal{L}(d_k^\pi, \cdot)$. The learner then wants to minimize its *average regret*, defined as

$$\bar{R}_K := \frac{1}{K}\left(\sum_{k=1}^K \ell_k(x_k) - \min_{x\in\mathcal{K}} \sum_{k=1}^K \ell_k(x)\right).$$

Thanks to this formulation, they are able to prove the following result:

Theorem 2.3.4 (No-Regret of Algorithm 2.3.3 [Zahavy et al., 2021])**.** *Assume that Alg$_\lambda$ and Alg$_\lambda$ have guaranteed average regret bounded as $\bar{R}_K^\pi \leqslant \epsilon_K$ and $\bar{R}_K^\lambda \leqslant \delta_K$, respectively. Then Algorithm 2.3.3 outputs $\bar{d}_K^\pi$ and $\bar{\lambda}_K$ satisfying*

$$\min_{d_\pi\in\mathcal{V}} \mathcal{L}(d_\pi, \bar{\lambda}_K) \geqslant \mathcal{F}^* - \epsilon_K - \delta_K \quad \textit{and} \quad \max_{\lambda\in\Lambda} \mathcal{L}(\bar{d}_K^\pi, \lambda) \leqslant \mathcal{F}^* + \epsilon_K + \delta_K,$$

with $\mathcal{F}^* = \min_{d^\pi\in\mathcal{V}} \mathcal{F}(d^\pi)$.

In other words, this theorem states that as long as no-regret algorithms are employed for the sub-routine algorithms of both cost and policy players, then Algorithm 2.3.3

[7] For concave utilities, it is equivalent to optimizing against $R = \lambda$.

[8] The learner is assumed to have perfect knowledge of the loss functions.

will produce a solution to the cMDP problem to any desired tolerance. In other words, rewards are indeed enough to solve cMDPs, as long as the infinite trials objective is considered.

Finally, Zahavy et al. [2021] instantiates specific cost and policy players' algorithms, illustrating how indeed Algorithm 2.3.3 unifies several branches of RL problems, as summarized in Table 2.1.

Objective	**Alg**$_\lambda$, **Alg**$_\pi$	**Application**
$\lambda \cdot d_\pi$	FTL/RL	(Standard) RL
$\|d^\pi - d^E\|_2^2$	FTL/BR	Apprenticeship Learning (AL) [Abbeel and Ng, 2004, Zahavy et al., 2020]
$d^\pi \cdot \log(d^\pi)$	FTL/BR	State Entropy Maximization [Hazan et al., 2019]
$\|d^\pi - d^E\|_\infty$	OMD/BR	AL [Syed and Schapire, 2007, Syed et al., 2008]
$\mathbb{E}_c\big[\lambda c \cdot \big(d^\pi - d^E(c)\big)\big]^\dagger$	OMD/BR	Inverse RL in contextual MDPs [Belogolovsky et al., 2021]
$\lambda_1 \cdot d^\pi, s.t. \lambda_2 \cdot d^\pi \leqslant c$	OMD/RL	Constrained MDP [Borkar, 2005, Altman, 1999, Bhatnagar and Lakshmanan, 2012, Tessler et al., 2019, Efroni et al., 2020, Calian et al., 2021]
$\mathrm{dist}(d^\pi, \mathcal{C})^{\dagger\dagger}$	OMD/BR	Feasibility of constrained cMDPs [Miryoosefi et al., 2019]
$\min_{\lambda_1,\ldots,\lambda_k} d_k^\pi \cdot \lambda_k$	OMD/RL	Adversarial MDPs [Rosenberg and Mansour, 2019]
$\mathrm{KL}(d^\pi \|\| d^E)$	FTL/RL	GAIL [Huang et al., 2016], State Marginal Marching [Lee et al., 2020]
$-\mathbb{E}_z \mathrm{KL}(d_z^\pi \|\| E_k d_k^\pi)^\ddagger$	FTL/RL	Diverse skill discovery [Gregor et al., 2017, Achiam et al., 2018, Eysenbach et al., 2018, Hausman et al., 2018, Tirumala et al., 2022]

Table 2.1: *Instances of Algorithm 2.3.3 for various cMDPs. FTL: Follow the Leader [Hazan et al., 2006]; OMD: Online Mirror Descent [Beck and Teboulle, 2003]; RL: Reinforcement Learning; BR: Best Response; d^E: expert state(-action) distribution;† c is the context variable; †† C is a convex set.*

Non-Markovianity Matters with Single Trials

As discussed previously, the mismatch between the infinite-trials and finite-trials formulations becomes critical when the RL agent is evaluated over only a handful of trials, just one in the worst case. Interestingly, this mismatch is reflected in the nature of the policies that attain the optimal solution. In particular, Mutti et al. [2023] shows that while the optimal policy in the infinite-trials setting is a *Markovian* policy, single-trial problems always admit a deterministic non-Markovian optimal policy, whereas the best policy within the space of Markovian policies must be randomized. Crucially, they also show that this randomization degrades the single-trial performance of Markovian policies compared to the optimal non-Markovian policy.

Their analysis leverages the notion of a *value gap*, defined as the difference between the value of a policy and the value of an optimal policy over a certain horizon. For a horizon T, it is defined as

$$\mathcal{V}_T(\pi) = \mathcal{F}^* - \mathbb{E}_{d_1 \sim p_1^\pi} \left[\mathcal{F}(d_1)\right], \tag{2.66}$$

where $\mathcal{F}^* = \max_{\pi^* \in \Pi} \mathbb{E}_{d_1 \sim p_1^{\pi^*}} \left[\mathcal{F}(d_1)\right]$ is the value achieved by an optimal policy $\pi^* \in \Pi$ over T steps. Similarly, $\mathcal{V}_t(\pi, s)$ denotes the value gap induced by π over t steps starting from the state s, such that $\mathcal{V}_T(\pi) = \mathbb{E}_{s \sim \mu}[\mathcal{V}_T(\pi, s)]$ and $\mathcal{V}_0(\pi, s) = 0$ for all $s \in \mathcal{S}$.

Using this concept, it is possible to formalize how non-Markovian policies enjoy favorable properties. First of all, the following result holds:

Lemma 2.3.5. *For every convex MDP $\mathcal{M}^{\mathcal{F}}$, there exists a deterministic non-Markovian policy $\pi_{\mathrm{NM}} \in \Pi^{\mathrm{D,NM}}$ such that*

$$\pi_{\mathrm{NM}} \in \arg\max_{\pi \in \Pi^{\mathrm{NM}}} \mathbb{E}_{d_1 \sim p_1^\pi} \left[\mathcal{F}(d_1)\right], \tag{2.67}$$

which suffers zero value gap: $\mathcal{V}_T(\pi_{\mathrm{NM}}) = 0$.

Moreover, whenever the deterministic non-Markovian optimal policy must adapt its decisions based on the history leading to a state, an optimal Markovian policy for the same objective must necessarily be stochastic. This randomization is harmful to its performance:

Lemma 2.3.6. *Let π_{M} be an optimal Markovian policy for $\zeta_1(\pi)$ in the cMDP $\mathcal{M}^{\mathcal{F}}$. Then for any $\mathrm{s}_t \in \mathcal{T}_{\mathcal{S}}^t$, $\underline{\mathcal{V}}_{T-t}(\pi_{\mathrm{M}}) \leqslant \mathcal{V}_{T-t}(\pi_{\mathrm{M}}) \leqslant \overline{\mathcal{V}}_{T-t}(\pi_{\mathrm{M}})$ where*

$$\begin{aligned}
\underline{\mathcal{V}}_{T-t}(\pi_{\mathrm{M}}) &= \frac{\mathcal{F}^* - \mathcal{F}_2^*}{\pi_{\mathrm{M}}(a^* \mid s_t)} \mathrm{Var}_{\mathbf{s} \oplus s_t \sim p_{1,t}^{\pi_{\mathrm{NM}}}} \left[\mathbb{E}\left[\mathcal{B}(\pi_{\mathrm{NM}}(a^* \mid \mathbf{s} \oplus s_t))\right]\right], \\
\overline{\mathcal{V}}_{T-t}(\pi_{\mathrm{M}}) &= \frac{\mathcal{F}^* - \mathcal{F}_*}{\pi_{\mathrm{M}}(a^* \mid s_t)} \mathrm{Var}_{\mathbf{s} \oplus s_t \sim p_{1,t}^{\pi_{\mathrm{NM}}}} \left[\mathbb{E}\left[\mathcal{B}(\pi_{\mathrm{NM}}(a^* \mid \mathbf{s} \oplus s_t))\right]\right],
\end{aligned}$$

where $\pi_{\mathrm{NM}} \in \arg\max_{\pi \in \Pi^{\mathrm{D,NM}}} \mathbb{E}_{d_1 \sim p_1^\pi} \left[\mathcal{F}(d_1)\right]$, Var denotes the variance of a random variable, $\mathcal{B}(x)$ denotes a Bernoulli variable with parameter x, and

$$\begin{aligned}
\mathcal{F}_* &= \min_{\mathbf{s} \in \mathcal{T}_{\mathcal{S}}^{T-t}} \mathcal{F}(d(\cdot \mid \mathbf{s}_t \oplus \mathbf{s})), \\
\mathcal{F}_2^* &= \max_{\mathbf{s} \in \mathcal{T}_{\mathcal{S}}^{T-t} \setminus \mathcal{T}_{\mathcal{S}}^{T-t,\star}} \mathcal{F}(d(\cdot \mid \mathbf{s}_t \oplus \mathbf{s})) \quad \textit{s.t.} \quad \mathcal{T}_{\mathcal{S}}^{T-t,\star} = \arg\max_{\mathbf{s} \in \mathcal{T}_{\mathcal{S}}^{T-t}} \mathcal{F}(d(\cdot \mid \mathbf{s}_t \oplus \mathbf{s})).
\end{aligned}$$

These two results can be combined into the following theorem:

Theorem 2.3.7. *For every convex MDP $\mathcal{M}^{\mathcal{F}}$, the optimal policy π_{NM} is deterministic and non-Markovian, whereas the optimal Markovian policy π_{M} is randomized. The value gap of the optimal Markovian policy satisfies*

$$\mathcal{V}_T(\pi_{\mathrm{M}}) \geqslant \mathcal{V}_T(\pi_{\mathrm{NM}}) = 0. \tag{2.68}$$

This result highlights the importance of non-Markovianity in single-trial convex RL: the class of Markovian policies is dominated by that of non-Markovian policies. Most importantly, it shows that non-Markovian policies are strictly better than Markovian ones in many convex MDPs of practical interest, particularly those where the optimal

Markovian policy must be randomized to achieve optimality. The key insight is that this happens because of its inability to disambiguate the underlying history, whereas a non-Markovian policy can leverage the full trajectory and deterministically select the optimal action.

Unfortunately, despite the theoretical appeal of non-Markovian policies, they are notoriously difficult to learn. For this reason, the following sections explore the role of Markovian policies and discuss how one might circumvent the need for non-Markovian policies using alternative approaches.

CHAPTER 3

Unsupervised Pre-Training in Reinforcement Learning

In the previous chapter, we presented a broad overview of how RL provides a powerful framework for solving sequential decision-making problems. Due to its favorable properties, RL achieved impressive results even in complex domains, ranging from video-games [Mnih et al., 2013, 2015, Silver et al., 2016, Berner et al., 2019, Wurman et al., 2022] to nuclear-reactor control [Duval et al., 2024].

However, a closer examination of the learning pipelines in these success stories reveals a considerable degree of human supervision. In particular, RL has demonstrated its full potential primarily in settings where a clear and informative reward function is available, offering a direct link between task specification and agent feedback. Yet in practice, such reward functions are rarely innate to the environment. Instead, they are often painstakingly designed by human experts, with the intent of encouraging desirable behaviors. This design process is non-trivial and typically requires extensive domain knowledge, significantly limiting the autonomy of RL agents. As a result, RL's promise as a fully autonomous learning paradigm remains undercut: every new task demands a bespoke reward formulation, and the generalization capabilities across tasks, as achieved via unsupervised pre-training in other areas of machine learning, have yet to be fully realized in RL. Most contemporary RL methods still operate in a tabula rasa fashion, learning from scratch, which is not only inefficient but often impractical in real-world applications, where data collection is costly, time-consuming, or even risky [Agarwal et al., 2022].

In response to these challenges, the framework of unsupervised RL [Laskin et al., 2021] has recently emerged as a compelling alternative, offering the potential to overcome these limitations through pre-training. In this setting, learning is decomposed into two distinct phases, as visualized in Fig. 3.1. The first, known as the *unsupervised pre-*

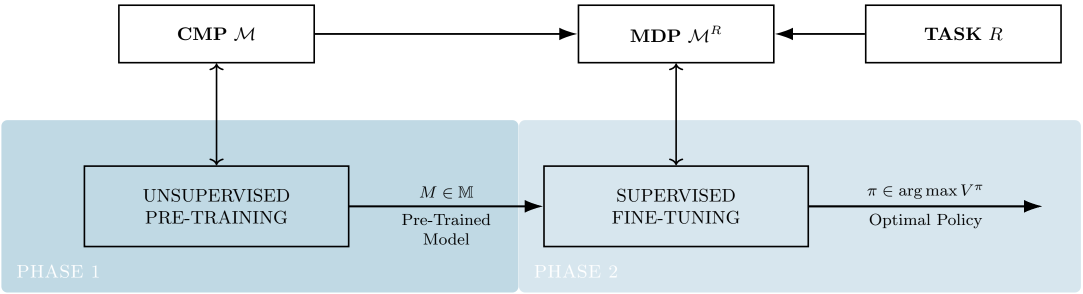


Figure 3.1: *Unsupervised RL as of Laskin et al. [2021], Mutti [2023].*

training phase, involves the agent interacting with a CMP to acquire general-purpose knowledge, which is distilled into a pre-trained model. This model, denoted by $\mathbb{M}$, can take various forms: a representation of transition dynamics, an abstract state representation, a single policy, a policy class, or simply a dataset of collected trajectories. The goal of this phase is not to solve a particular task, but to encode useful knowledge about the environment that can later facilitate learning.

The second phase, termed *supervised fine-tuning*, begins once a reward function R is revealed. At this point, the CMP becomes a standard MDP $\mathcal{M}^R$, and the agent leverages the pre-trained model $\mathbb{M}$ to efficiently solve the new objective. Depending on the nature of $\mathbb{M}$, this may involve direct planning (if $\mathbb{M}$ models transitions accurately) or further interaction with the environment (if $\mathbb{M}$ represents an exploratory policy). A key advantage of this approach is that the same model $\mathbb{M}$ can be reused across multiple tasks defined on the same CMP. Although the unsupervised pre-training phase may be computationally intensive, its benefits can amortize over a broad spectrum of downstream tasks, significantly reducing the learning burden during fine-tuning.

Let $\mathbb{M}$ denote the class of models to be pre-trained. We can then define the objective of unsupervised pre-training as follows:

$$\max_{M \in \mathbb{M}} \mathcal{F}_{\text{pre-train}}(M, \mathcal{M}), \tag{3.1}$$

where $\mathcal{F}_{\text{pre-train}}$ is a task-agnostic objective that scores the utility of a model $M \in \mathbb{M}$ in capturing relevant structure in the CMP $\mathcal{M}$. For example, in the context of this thesis, we focus on the case where $\mathbb{M}$ is the space of Markovian policies Π, the model M corresponds to a policy $\pi \in \Pi$, and the objective $\mathcal{F}_{\text{pre-train}}$ is the entropy $\mathcal{H}$ of the state distribution d^π induced by π over $\mathcal{M}$ [Hazan et al., 2019].

Beyond entropy maximization, the literature has explored a variety of model classes $\mathbb{M}$ and objectives $\mathcal{F}_{\text{pre-train}}$, some of which involve complex inner-loop computations such as planning [Jin et al., 2020]. The central theme is to construct models that yield downstream benefits when a reward function becomes available.

Once the reward R is introduced, the fine-tuning objective is given by:

$$\max_{\pi \in \Pi} \mathcal{J}_{\mathcal{M}^R}(\pi, M), \tag{3.2}$$

where the agent's goal is to optimize performance in the MDP $\mathcal{M}^R$, leveraging the pre-trained model $M \in \mathbb{M}$. For instance, M could be a policy π that initializes an RL algorithm, replacing the standard random initialization. While the nature of the objective in Eq. (3.2) remains that of standard RL, the presence of the pre-trained model may significantly accelerate learning, yielding significant empirical [Laskin et al., 2021] and theoretical [Jin et al., 2020, Xie et al., 2022] benefits, also in terms of regret minimization.

Formally, the regret of a learning algorithm $\mathbb{A}$ over K episodes in the MDP $\mathcal{M}^R$ is defined as:

$$\mathrm{Reg}_K(\mathcal{M}^R, \mathbb{A}) = \mathbb{E}_{s\sim\mu}\left[\sum_{k\in[K]} \left(V^\star(s) - V^{\pi_k}(s)\right)\right],$$

where $V^\star$ is the optimal value function in $\mathcal{M}^R$, and V^{π_k} is the value function of the policy π_k deployed at episode k by algorithm $\mathbb{A}$. The benefit of pre-training can thus be quantified by comparing the regret of $\mathbb{A}$ when initialized with a pre-trained model $\mathbb{M}$ versus a random initialization [Ye et al., 2023].

Importantly, it has been shown that any RL algorithm learning from scratch in a tabula rasa fashion must incur at least $\Omega(\sqrt{|\mathcal{S}||\mathcal{A}|HK})$ regret [Auer et al., 2008]. In contrast, if the pre-trained model $\mathbb{M}$ enables near-optimal zero-shot planning across all reward functions R, the regret can be reduced to a constant ϵ, or at the very least, improved significantly through better multiplicative constants when using carefully pre-trained policies [Ye et al., 2023].

3.1 Literature Overview of Pre-Training in Reinforcement Learning

This section presents an overview of the literature surrounding (unsupervised) pre-training in RL. While the field of unsupervised RL is still relatively new, it is rapidly evolving, with an increasing number of methods being proposed to overcome the limitations of learning from scratch. A central axis of classification within this body of work lies in identifying the type of model that is pre-trained during the unsupervised phase and subsequently transferred to the supervised phase. In what follows, we review representative approaches across different categories of pre-trained models, which are also summarized in Table 3.1 for convenience. Our analysis follows the taxonomy from Mutti [2023], updated to cover more recent results in the field. However, we note that Agarwal et al. [2025a] offers a different classification that includes a few works not covered by this framing.

3.1.1 Representations Pre-Training

Many RL tasks involve high-dimensional and complex inputs such as visual data, for which suitable representations are often critical to achieve good performance [Merckling, 2021]. In this context, substantial effort has been dedicated to learning useful representations of the environment independently of any specific task. This line of work naturally fits the unsupervised pre-training paradigm, where the aim is to learn a mapping from input pairs to a lower-dimensional representation space, denoted as $\mathbb{M}_\phi = \left\{\phi : \mathcal{O} \times \mathcal{A} \to \mathbb{R}^d\right\}$.

Several studies have shown that task-agnostic state-action representations can be used to factorize the transition dynamics of the environment, enabling efficient planning during the supervised fine-tuning phase. Notably, works such as Misra et al. [2020], Agarwal et al. [2020], Modi et al. [2024] develop theoretical frameworks in which the transition function of a reward-free POMDP admits a low-rank decomposition of

Approach	Pre-training	References
Low-rank or Block MDPs	Representations	Misra et al. [2020], Agarwal et al. [2020], Modi et al. [2024]
Contrastive Loss	Representations	Laskin et al. [2020], Luu et al. [2022], Yu et al. [2025]
Reconstruction Loss	Representations	Burda et al. [2019], Anand et al. [2019], Seo et al. [2022], Meng et al. [2023]
Supervised Learning Loss	Representations	Yuan et al. [2022], Yoon et al. [2023]
Reward-Free RL	Transition Model	Jin et al. [2020], Kaufmann et al. [2021], Ménard et al. [2021], Zhang et al. [2020d]
Task-Agnostic RL	Transition Model	Zhang et al. [2020c]
Forward-Backward & Successor Measures	Transition Model	Touati and Ollivier [2021], Touati et al. [2023], Agarwal et al. [2025a,b]
Behavioral Foundation Models	Transition Model	Tirinzoni et al. [2025], Sikchi et al. [2025]
World Models	Transition Model	Ha and Schmidhuber [2018], Hafner et al. [2019], Matsuo et al. [2022] Hafner et al. [2023], Pearce et al. [2024]
Curiosity	Transition Model	Schmidhuber [1991], Pathak et al. [2017], Burda et al. [2018]
Reward-Free Data Collection	Dataset	Wang et al. [2020], Zanette et al. [2020]
ExORL	Dataset	Yarats et al. [2022]
Explore2Offline	Dataset	Lambert et al. [2022]
Count-Based	Dataset	Bellemare et al. [2016]
Policy Space Compression	Policy Space	Mutti et al. [2022c], Tenedini et al. [2025]
Policy Collection-Elimination	Policy Space	Ye et al. [2023]
Mutual Information for Skill Discovery	Policy Space	Gregor et al. [2017], Eysenbach et al. [2018], Hansen et al. [2019], Sharma et al. [2019], Campos et al. [2020], Liu and Abbeel [2021a], He et al. [2022], Zahavy et al. [2022]
Entropy Maximization	Policy	see Table 3.2
High-Level Hierarchical Policies	Policy	Pertsch et al. [2021], Baker et al. [2022], Ramrakhya et al. [2023], Yuan et al. [2024]
Fine-Tuning Mechanisms	Policy	Campos et al. [2021], Pislar et al. [2021], Xie et al. [2021], Uchendu et al. [2023]

Table 3.1: *Overview of the literature in Unsupervised Pre-Training for RL. The **Approach** column reports either a specific work or a stream of similar works. The **Pre-training** column reports the model to be pre-trained. The **References** column reports a (non-exaustive) list of references.*

the form $\mathbb{P}(o' \mid o, a) = \phi(o, a) \cdot \psi(o')$, under structural assumptions. By planning directly in the space of such reduced representations, agents can often recover optimal or near-optimal strategies with minimal or no further interaction during fine-tuning. These works also provide analyses of the sample complexity involved in representation learning and the associated sub-optimality guarantees of downstream policies, in both model-based [Agarwal et al., 2020] and model-free [Modi et al., 2024] regimes.

In addition to these theoretically grounded approaches, a parallel line of research has focused on representation learning through neural networks trained with unsupervised objectives. These include contrastive losses [Laskin et al., 2020, Luu et al., 2022], reconstruction-based losses [Burda et al., 2019, Anand et al., 2019, Seo et al., 2022, Meng et al., 2023], and others. Although these methods provide weaker theoretical guarantees, they have demonstrated strong empirical performance on high-dimensional RL benchmarks [Laskin et al., 2021, Luu et al., 2022, Yoon et al., 2023]. Complementing this trend, Yuan et al. [2022] found that visual encoders pre-trained on unrelated domains (e.g., ImageNet) can provide general-purpose features useful for a wide range of RL tasks, showcasing the broader potential of transfer learning across domains. Finally, the work of Yu et al. [2025] extended representation pre-training to the multi-agent setting, showing that contrastively pre-trained communication networks between agents can generalize across different MARL tasks in a similar fashion.

3.1.2 Transition Models Pre-Training

Another prominent direction in unsupervised pre-training for RL is based on learning accurate models of the transition dynamics. Within this framework, the goal is often to estimate the function $p \in \mathbb{M}_p = \{p : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}\}$ that maps state-action pairs to distributions over next states. The rationale is that such models can be used for planning or as simulators during the supervised fine-tuning stage, especially when direct interaction with the environment becomes expensive or limited.

The typical pre-training objectives in this case either seek to minimise the discrepancy between the learned model and the true transition kernel, or to ensure near-optimality of the learned model across all possible reward functions. The first strategy leads to objectives of the form

$$\min_{p \in \mathbb{M}_p} \sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} \left\| P\left(s' \mid s, a\right) - p\left(s' \mid s, a\right) \right\|_2 , \tag{3.3}$$

while the second aims to minimize the performance gap between the optimal policy induced by the learned model and the optimal policy for the true model under a worst-case reward:

$$\min_{p \in \mathbb{M}_p} \sup_{R \in \{\mathcal{S} \times \mathcal{A} \to [0,1]\}} \mathbb{E}_{s \sim \mu} \left[\left| V_P^*(s) - V_p^*(s) \right| \right] . \tag{3.4}$$

The second formulation, in particular, has given rise to conceptually grounded approaches such as reward-free RL [Jin et al., 2020, Kaufmann et al., 2021, Ménard et al., 2021, Zhang et al., 2020d] and task-agnostic RL [Zhang et al., 2020c].

This formulation inspired practical implementations like the one of Touati and Ollivier [2021], in which a *Forward-Backward* (FB) representation is proposed, so as to make the fine-tuning phase extremely efficient. Later, Touati et al. [2023] noticed that FBs perform better and consistently across a variety of tasks compared to other representations such as *Successor Features* (SFs). More recently, Agarwal et al. [2025b]

introduced the concept of *Proto Successor Measures* (PSMs), demonstrating that any possible behavior can be represented using an affine combination of these functions, yet that their linearity and the fact that they are policy independent makes the fine-tuning phase extremely easier. Recently, Agarwal et al. [2025a] outlined a comprehensive treatment of the relationships among these objects, proposing a unified framework for FBs, PSMs, SFs and other representations.

Additionally, this line of research has recently led to the development of so-called *behavioral foundation models* [Tirinzoni et al., 2025, Sikchi et al., 2025], which aim to learn task embeddings alongside corresponding near-optimal behaviors and incorporating an inference procedure to directly retrieve the latent task embedding and associated policy for any given reward function.

In parallel, the estimation-oriented objective in Eq. (3.3) has been adopted in both theoretical studies [Tarbouriech et al., 2020] and empirical works targeting complex observation spaces. This line of research has led to the development of so-called *world models* [Ha and Schmidhuber, 2018, Hafner et al., 2019, Matsuo et al., 2022, Hafner et al., 2023], which aim to learn compact and expressive simulators of the environment.

Remarkably, recent results [Pearce et al., 2024] have shown that world model pre-training exhibits similar scaling laws to those observed in large language models, suggesting deep connections between predictive modeling across modalities. Additionally, alternate pre-training objectives have been proposed to accelerate transition model acquisition. Among these, curiosity-based intrinsic motivations [Schmidhuber, 1991, Pathak et al., 2017, Burda et al., 2018] reward agents for seeking out novel or surprising transitions, thereby guiding exploration in the absence of an external reward.

3.1.3 Datasets Pre-Training

A third and increasingly popular approach to unsupervised pre-training involves collecting a dataset of environment interactions rather than directly pre-training a model. In this paradigm, the unsupervised phase focuses on generating informative data, which can later be leveraged during the supervised fine-tuning phase through direct planning or via offline RL algorithms [Levine et al., 2020]. This strategy reframes the pre-training problem as one of data acquisition, where the objective is to gather a dataset that maximises the utility of future decision-making. Formally, the model class can be informally described as $\mathbb{M}_{\mathcal{D}}$, consisting of all datasets $\mathcal{D}$ made up of N transition tuples (s, a, s'). The central questions in this line of work pertain to how such a dataset should be constructed-specifically, what policies or exploration strategies should be employed during data collection, and how large the dataset needs to be to guarantee useful performance upon fine-tuning.

Some of the most influential theoretical work in this domain draws inspiration from reward-free RL. Rather than learning a parametric model or representation, these studies consider objectives that seek to minimize policy sub-optimality with respect to a fixed dataset. For example, Wang et al. [2020], Zanette et al. [2020] propose a dataset-based analogue of the objective in Eq. (3.4), where planning is performed using an offline RL algorithm acting on a fixed transition dataset:

$$\min_{\mathcal{D}\in\mathbb{M}_{\mathcal{D}}} \sup_{R\in\{\mathcal{S}\times\mathcal{A}\to[0,1]\}} \mathop{\mathbb{E}}_{s\sim\mu}\left[\left|V^*_P(s) - \widehat{V}_{\mathcal{D}}(s)\right|\right], \tag{3.5}$$

where $\widehat{V}_{\mathcal{D}}(s)$ denotes the value function under the policy obtained by an offline algorithm that uses $\mathcal{D}$ as its input. In the special case of linear MDPs-where the transition dynamics admit a linear decomposition with respect to some known representation-these works provide compelling theoretical guarantees on the effectiveness of the collected dataset.[1]

Beyond theoretical guarantees, recent empirical work has focused on designing practical strategies for data collection in complex and high-dimensional domains [Yarats et al., 2022, Lambert et al., 2022]. While these methods may lack the formal guarantees provided by reward-free RL, they impose no restrictive assumptions on the structure of the environment and remain compatible with a broad range of offline RL algorithms during the fine-tuning phase.

3.1.4 Policy Spaces Pre-Training

An alternative set of methodologies involves pre-training within the policy space using unsupervised interactions. The underlying intuition is to reduce the complexity of the search space for downstream tasks: by learning a structured or compressed policy space during the unsupervised phase, the supervised fine-tuning stage can more effectively identify a high-performing policy with reduced sample complexity. In this setting, the model class is given by $\mathbb{M}_{\Pi} = \{\Pi_{\text{red}} \in \mathscr{P}(\Pi)\}$, where $\mathscr{P}(\Pi)$ denotes the power set of the policy space Π.[2]

From a theoretical perspective, several works have focused on formal criteria for constructing the reduced policy space Π_{red}. A typical objective in this line of research involves maximizing the coverage of the state-action distribution induced by policies in Π_{red}, ensuring it approximates the distributional support of the full policy space Π [Mutti et al., 2022c, Ye et al., 2023, Tenedini et al., 2025]. When this coverage condition is satisfied, fine-tuning can be restricted to the smaller space Π_{red} without significant loss of optimality, often resulting in improved regret bounds [Ye et al., 2023] and more sample-efficient learning [Tenedini et al., 2025].

On the methodological side, many approaches have embraced diversity-driven objectives for constructing Π_{red}. Instead of attempting exhaustive coverage, these methods aim to pre-train a compact set of diverse skills or policies that span a wide range of behaviors. A common formalism in this context is the maximization of mutual information between latent variables and the induced trajectories or outcomes of the policy [Gregor et al., 2017, Eysenbach et al., 2018, Hansen et al., 2019, Sharma et al., 2019, Campos et al., 2020, Liu and Abbeel, 2021a, He et al., 2022, Zahavy et al., 2022]. These techniques, often framed under the umbrella of *unsupervised skill discovery*, produce a discrete or continuous set of skills that can serve as primitives for hierarchical RL or as an initial basis for fine-tuning on specific downstream tasks.

While many of these approaches are heuristic in nature and lack formal guarantees, they have demonstrated practical effectiveness in a range of complex environments. Moreover, the learned skill sets or policy subsets often encode reusable behavioral abstractions that can be efficiently recombined, making them attractive for general-purpose RL pipelines.

[1] In this context, the transition model is presumed to possess a linear representation, which facilitates theoretical examination.

[2] $\mathscr{P}(\Pi)$ represents the power set of the policy space Π.

3.1.5 Policies Pre-Training

Among the various potential targets for pre-training, this thesis focuses exclusively on the unsupervised pre-training of policies, where the model class $\mathbb{M}$ corresponds to a policy space Π. This objective can be formally expressed as:

$$\max_{\pi \in \Pi} \mathcal{F}\left(d^{\pi}\right), \tag{3.6}$$

where d^{π} denotes the state distribution induced by policy π under the CMP $\mathcal{M}$, and $\mathcal{F}$ is a functional mapping state distributions to real values, i.e., $\mathcal{F} : \Delta_{\mathcal{S}} \rightarrow \mathbb{R}$. The central aim of optimising (3.6) is to learn a policy that accelerates supervised fine-tuning [Uchendu et al., 2023]. In this thesis, we centre our discussion on the pre-training phase. Although we present fine-tuning results, we adopt simple strategies to deploy the pre-trained policy, intended primarily for evaluating pre-training effectiveness rather than advancing fine-tuning methodology.

The foundational idea of using entropy-based objectives for policy pre-training was introduced by Hazan et al. [2019], who proposed maximising the Shannon entropy of the discounted state distribution. Their algorithm constructs a mixture of policies through a gradient method, iteratively estimating the state distribution and solving a sequence of RL subproblems. A similar game-theoretic approach was later proposed by Lee et al. [2020], targeting the entropy of the marginal state distribution instead. Other gradient approaches include Tarbouriech and Lazaric [2019], which focuses on the stationary state-action distribution, though this technique may suffer from slow convergence of the policy mixture. A related method by Mutti and Restelli [2020] seeks to pre-train a single policy that simultaneously accounts for the entropy of the stationary distribution and the system's mixing time.

While some of these methods have been evaluated in continuous domains [Hazan et al., 2019, Lee et al., 2020], they typically rely on accurate estimation of either the state distribution [Hazan et al., 2019, Lee et al., 2020] or the transition dynamics [Tarbouriech and Lazaric, 2019, Mutti and Restelli, 2020], which limits their applicability in complex, high-dimensional environments. To address this, Mutti et al. [2021] proposed a non-parametric entropy estimator and optimised it via policy gradient, enabling single-policy pre-training in challenging continuous control domains. This approach was extended by Liu and Abbeel [2021a], who integrated entropy estimation with learned state embeddings for visual-input domains. Even random encodings, as shown by Seo et al. [2021], can suffice for entropy-driven pre-training. Similarly, Yarats et al. [2021] explored concurrent learning of state representations and latent prototypes to stabilise entropy estimates.

Building upon these advances, Zahavy et al. [2021] provided a theoretical formulation by framing entropy maximization as an instance of convex RL. Using Fenchel duality, they cast the problem as a two-player zero-sum game-between a policy player and a reward-generating adversary, and applied no-regret algorithms to minimise regret in this setting. The resulting MetaEnt algorithm offers strong sample complexity guarantees. This line of work was extended in Tiapkin et al. [2023], who examined both visitation and trajectory entropy. They proposed EntGame, a game-theoretic approach with improved sample complexity over prior methods, and *RL-Explore-Ent*, an algorithm that solves regularised Bellman equations using transition models learned from exploratory trajectories.

While prior methods primarily rely on Shannon entropy, recent work explores alternatives. Zhang et al. [2021a] argued that Rényi entropy provides better coverage incentives and introduced *MaxRenyi*, a method for directly optimising this objective. In parallel, Guo et al. [2021] proposed a geometry-aware entropy that incorporates structural properties of the space. Another perspective comes from Nedergaard and Cook [2022], who maximise a lower bound on state entropy, demonstrating superior policy quality in some settings. Their estimator, a k-means-based approximation, is also used by Yang and Spaan [2023], who incorporate safety constraints into the entropy maximisation problem and present a trust-region method with convergence guarantees. Finally, Jain et al. [2023] introduced $\eta\psi$-Learning, which combines predecessor (η) and successor (ψ) representations to estimate entropy from single trajectories. This allows the synthesis of deterministic, non-Markovian policies from trajectory-based learning. A comprehensive empirical evaluation by Zisselman et al. [2023] confirmed the generalisability of pre-trained policies across tasks.

Recent studies have begun extending entropy maximization principles to settings beyond MDPs. For POMDPs, Savas et al. [2022] designed finite-state controllers to maximise the entropy of observation trajectories under reward constraints, while Zamboni et al. [2024b,a] developed policy optimisation methods that operate on observations or latent state representations. These approaches focus on achieving theoretical guarantees relative to entropy objectives over the true latent state space. In the context of Markov games, Zamboni et al. [2025b] proposed a decentralised trust-region approach to maximise entropy over the state space, and Gemp et al. [2025] introduced a centralised projected-gradient algorithm with convergence guarantees, yet assuming model knowledge. Similarly, in the context of Parallel MDPs [Sucar, 2007], De Paola et al. [2025] proposed a policy gradient method to enhance exploration in parallel settings, showing that parallel exploration is more efficient than single-agent exploration when mixture distributions are properly exploited. A summary of these entropy-based policy pre-training algorithms is presented in Table 3.2.

As a final note, in parallel to these developments, another stream of research has explored the pre-training of high-level policies through imitation learning on pre-collected datasets. These approaches provide temporally abstract actions that facilitate downstream tasks within hierarchical RL frameworks [Pertsch et al., 2021, Baker et al., 2022, Ramrakhya et al., 2023, Yuan et al., 2024].

3.2 A Dive Into Policy Pre-Training via State Entropy Maximization

In the previous section, we provided a high-level overview of the diverse methods employed to address policy pre-training through the optimization of functionals over the induced state distribution. We now delve into a more detailed exposition of the foundational work by Hazan et al. [2019], which will serve as the theoretical basis for the remainder of this thesis. In their seminal contribution, the authors propose the entropy of the state distribution induced by a policy as an objective for exploration in the absence of extrinsic rewards. They formally define the entropy objective as:

$$\mathcal{H}(d_\gamma^\pi) := -\mathbb{E}_{s\sim d_\gamma^\pi}\left[\log d_\gamma^\pi(s)\right],$$

Algorithm	Distribution	Space	Reference
MaxEnt	Discounted	State	Hazan et al. [2019]
FW-AME	Stationary	State-Action	Tarbouriech and Lazaric [2019]
SMM	Marginal	State	Lee et al. [2020]
IDE^{3}AL	Stationary	State	Mutti and Restelli [2020]
MEPOL	Marginal	State	Mutti et al. [2021]
MaxRényi	Discounted	State-Action	Zhang et al. [2021a]
GEM	Marginal	State	Guo et al. [2021]
APT	Marginal	State	Liu and Abbeel [2021b]
RE3	Marginal	State	Seo et al. [2021]
Proto-RL	Marginal	State	Yarats et al. [2021]
MetaEnt	Discounted	State	Zahavy et al. [2021]
RL-Explore-Ent	Discounted	State Trajectories	Zahavy et al. [2021]
KME	Discounted	State	Nedergaard and Cook [2022]
FSC	Stationary	Observation Trajectories	Savas et al. [2022]
CEM	Marginal	State	Yang and Spaan [2023]
$\eta\psi$-Learning	Discounted	State	Jain et al. [2023]
ExpGen	Marginal	State	Zisselman et al. [2023]
MOE	Marginal	Observation	Zamboni et al. [2024b]
MBE	Marginal	Latent State	Zamboni et al. [2024a]
TRPE	Marginal	State	Zamboni et al. [2025b]
PGL	Marginal	State	Gemp et al. [2025]
PGPSE	Marginal	State	De Paola et al. [2025]

Table 3.2: *Overview of the literature in Unsupervised Pre-Training for RL via Maximum Entropy. For each algorithm, we report the nature of the objective, i.e., whether it considers stationary, discounted, or marginal distributions (**Distribution**), and which space it accounts for (**Space**).*

where $\mathcal{H}$ denotes the Shannon entropy and d_γ^π the discounted state distribution induced by policy π. Since $\mathcal{H}$ is a concave function of d_γ^π, this formulation aligns naturally with the framework of Convex RL [cRL, Hazan et al., 2019, Zhang et al., 2020c]. The infinite-horizon nature of the discounted formulation renders the problem analogous to a cRL setting with infinite trials in episodic environments.[3]

While algorithms with provable efficiency have been developed for this infinite-trial cRL setting [Zhang et al., 2020a, Zahavy et al., 2021], it is important to discuss the computational implications specific to the state entropy maximization problem. Before turning to those considerations, we briefly motivate why maximizing the entropy of the state distribution has emerged as a central objective in unsupervised RL.

[3] To make this connection explicit, consider a long trajectory generated by the Markov chain under policy π. As temporal correlations decay over time, one can treat distant segments of the trajectory as independent episodes. Averaging state visitations over such episodes yields an estimate of the discounted state distribution d_γ^π.

Motivation

The rise of state entropy maximization as a pre-training objective is primarily attributed to its strong empirical performance [Laskin et al., 2021], especially in practical, high-dimensional domains. However, there are also compelling informal arguments that offer insight into its effectiveness. In the context of offline RL [Levine et al., 2020], which focuses on learning near-optimal policies from fixed datasets, it is well established [Antos et al., 2008, Chen and Jiang, 2019, Jin et al., 2021b, Foster et al., 2021, Zhan et al., 2022] that the coverage of the state space in the dataset plays a pivotal role in determining the sample complexity. This requirement is often quantified through the *concentrability coefficient*:

$$C(\mathcal{D}) := \sup_{\pi \in \Pi, s \in \mathcal{S}} \frac{d_\gamma^\pi(s)}{\mathcal{D}(s)}, \tag{3.7}$$

where $\mathcal{D}(s)$ denotes the empirical state distribution in the dataset. The rationale behind this condition is that adequate coverage of the state space ensures that the dataset contains enough information to accurately evaluate the optimal action in each state with high probability.

Recent work by Xie et al. [2022] explicitly connects the initial policy's state coverage to the sample complexity of the fine-tuning task, extending the relevance of coverage conditions to the online RL setting. In this light, unsupervised policy pre-training can be viewed as the problem of finding a policy that induces an optimal data distribution $\mathcal{D}$. While directly minimising the concentrability coefficient over all policies is intractable, entropy maximization offers a tractable surrogate by encouraging uniform coverage of the state space.

Policies optimised for state entropy have also proven effective in related settings, such as the reward discovery problem Tarbouriech and Lazaric [2019], which seeks to minimise the number of interactions needed to visit all state-action pairs at least once (in high probability), and the reward-free RL formulation [Jin et al., 2020], which involves collecting information sufficient to compute near-optimal policies for any reward function (in high probability).

The Policy Viewpoint: Primal Problem

In its primal form, the objective of state entropy maximization entails directly searching for a policy that induces a high-entropy state distribution. This objective can be expressed as:

$$\max_{\pi \in \Pi} \mathcal{H}(d_\gamma^\pi), \tag{3.8}$$

where the policy $\pi \in \Pi$ is the optimisation variable. Although Eq. (3.8) is framed as the maximisation of a concave function, the relationship between the policy parameters $\pi(a \mid s)$ and the resulting state distribution d_γ^π is highly non-trivial. Specifically, d_γ^π is defined recursively as:

$$d_\gamma^\pi(s) = (1-\gamma)\mu(s) + \gamma \sum_{s' \in \mathcal{S}} \sum_{a' \in \mathcal{A}} d_\gamma^\pi(s')\pi(a' \mid s')\mathbb{P}(s \mid s', a'),$$

where μ is the initial state distribution, γ is the discount factor, and $\mathbb{P}$ is the transition kernel of the environment. As shown by Hazan et al. [2019], this recursive dependence implies that $\mathcal{H}(d_\gamma^\pi)$ is not a concave function of the policy parameters, which complicates the use of standard policy gradient methods [Sutton et al., 1999] for optimisation. In particular, gradient ascent on the entropy objective, i.e., computing $\nabla_\pi \mathcal{H}(d^\pi)$, does not guarantee convergence to a global optimum.

To address this challenge, Hazan et al. [2019] introduce a gradient (Frank-Wolfe) method that iteratively constructs a mixture of policies to approximate the entropy-optimal distribution. We discuss their approach in detail in the following section. In subsequent chapters, we examine alternative surrogate objectives that make the primal entropy maximization problem more tractable for modern optimisation techniques.

Frank-Wolfe for Maximum State Entropy

To address the non-concavity of the primal objective (3.8), Hazan et al. [2019] propose a conditional gradient method-commonly known as the Frank-Wolfe algorithm [Frank et al., 1956], in a method they call *MaxEnt*. Rather than directly optimizing the intractable primal objective, MaxEnt decomposes it into a sequence of more manageable sub-problems. The solution to each sub-problem contributes to constructing a mixture of policies, whose overall state distribution progressively maximizes entropy.

Each sub-problem is formulated as solving a MDP using a reward function defined by the gradient of the entropy at the current policy mixture. Specifically, the reward is given by

$$R(s) = \left(\nabla_\pi \mathcal{H}(d^\pi_{\text{mix}})\right)(s), \tag{3.9}$$

where d^π_{mix} denotes the state distribution induced by the current policy mixture. Algorithm 3.2 summarizes the MaxEnt procedure. Detailed algorithmic insights and implementation notes can be found in Hazan et al. [2019].

Assuming full knowledge of the environment, particularly the transition model $\mathbb{P}$-MaxEnt is guaranteed to output a policy π_{mix} such that

$$\mathcal{H}(d^{\pi_{\text{mix}}}) \geqslant \max_{\pi \in \Pi} \mathcal{H}(d^\pi) - \epsilon$$

in a number of iterations

$$T = \text{poly}(|\mathcal{S}|, |\mathcal{A}|, \tfrac{1}{\epsilon}, \tfrac{1}{1-\gamma}),$$

thereby establishing its *computational efficiency*.

When the transition model $\mathbb{P}$ is unknown, MaxEnt remains applicable in a model-free setting. In this case, the algorithm relies on two components: (i) a *density estimator* capable of approximating the induced state distribution $\hat{d}^{\pi_{\text{mix}}}$, and (ii) a *planning oracle* that computes a near-optimal policy for the reward defined by the entropy gradient. Provided both components are implemented in a provably efficient manner, MaxEnt can achieve near-optimality by using

$$\tilde{O}\left(\frac{|\mathcal{S}|^2|\mathcal{A}|}{\epsilon^3(1-\gamma)^2}\right)$$

samples from the environment-making the algorithm *statistically efficient* as well.

Algorithm 3.2: MaxEnt [Hazan et al., 2019]

```
**Input:** Step size $\alpha$, iterations $T$, tolerance $\epsilon$
Initialize $\pi_0$ arbitrarily, set $\gamma_0 = 1$, $C_0 = \{\pi_0\}$, and $\pi_{\text{mix}} = (\gamma_0, C_0)$
for $t = 0, \dots, T-1$ do
    Estimate $\hat{d}^{\pi_{\text{mix}}}$ up to error $\epsilon$
    Compute rewards $R(s) = \left(\nabla_\pi \mathcal{H}(d^{\pi_{\text{mix}}})\right)(s)$
    Compute $\epsilon$-optimal policy $\pi_t$ using planning with reward $R$
    Update mixture:
        $\gamma_{t+1} = ((1-\alpha)\gamma_t, \alpha), \quad C_{t+1} = (C_0, \dots, \pi_t)$
    Update $\pi_{\text{mix}} = (\gamma_{t+1}, C_{t+1})$
end for
Output: State-entropy-maximizing policy $\pi_{\text{mix}}$
```

While MaxEnt offers strong theoretical guarantees, a few practical limitations are worth noting. First, it outputs a *mixture of policies* rather than a single Markovian policy. Though this mixture can be projected onto a single policy that approximates the same state distribution, such projection may entail computational or performance trade-offs. Alternatively, one might consider using the mixture directly for fine-tuning, though most standard RL algorithms are not designed to operate with policy mixtures.

Second, although state density estimation is straightforward in tabular domains, extending this to *continuous or high-dimensional state spaces* is significantly more challenging. The effectiveness and scalability of MaxEnt in such settings remain open areas of research.

The Occupancy Measure Viewpoint: Dual Problem

While the objective in Eq. (3.8) is concave in the state distribution, its dependence on the policy parameters makes the overall optimization problem non-concave. This observation motivates a dual formulation, where instead of optimizing over policies, we optimize directly over state(-action) distributions subject to consistency constraints.

To this end, we define again the set of valid discounted state-action distributions as:

$$\mathcal{V} = \left\{ \nu \in \Delta_{\mathcal{S}\times\mathcal{A}} : \sum_{a\in\mathcal{A}} \nu(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s'\in\mathcal{S}, a'\in\mathcal{A}} P(s|s',a')\nu(s',a'),\ \forall s \in \mathcal{S} \right\}.$$

With a slight abuse of notation, let $\nu(s) := \sum_{a\in\mathcal{A}} \nu(s,a)$ denote the marginal state distribution induced by ν. The dual optimization problem then becomes:

$$\max_{\nu\in\mathcal{V}} \mathcal{H}(\nu), \tag{3.10}$$

where $\mathcal{H}(\nu) := -\sum_{s\in\mathcal{S}} \nu(s) \log \nu(s)$ is the entropy of the induced state distribution.

Because $\mathcal{N}$ is a convex polytope [Puterman, 2014] and $\mathcal{H}(\nu)$ is concave in ν, Eq. (3.10) is a linearly constrained concave program-making it amenable to standard convex optimization techniques. Specifically, this formulation involves $|\mathcal{S}||\mathcal{A}|$ optimization variables and $2|\mathcal{S}||\mathcal{A}| + |\mathcal{S}|$ linear constraints.

Once the optimal solution $\nu^* \in \arg\max_{\nu \in \mathcal{V}} \mathcal{H}(\nu)$ is obtained, a corresponding policy π_{ν^*} that achieves maximum entropy can be extracted via normalization:

$$\pi_{\nu^*}(a|s) = \frac{\nu^*(s,a)}{\sum_{a' \in \mathcal{A}} \nu^*(s,a')} \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}.$$

This dual perspective provides a tractable and theoretically elegant approach to the state entropy maximization problem. In tabular domains, the dual problem can be solved efficiently using off-the-shelf solvers. However, the approach faces notable limitations. The number of variables and constraints grows linearly with $|\mathcal{S}||\mathcal{A}|$, making it computationally burdensome in large-scale settings. More critically, extending this formulation to continuous or high-dimensional spaces is non-trivial, as the set $\mathcal{V}$ becomes infinite-dimensional and the entropy functional may lose tractability.

Despite these challenges, the occupancy measure view offers important conceptual insight and lays the foundation for further approximation-based methods in more complex settings.

CHAPTER 4

Unsupervised Pre-Training with Partial Observability

In Chapter 3, we discussed various approaches to unsupervised pre-training in RL. While most of the existing literature has focused on representation learning for partially observable settings, the idea of directly pre-training *policies* in POMDPs remained significantly underexplored-despite the demonstrated effectiveness of such strategies in fully observable environments and the ubiquity of partial observability in real-world applications. Consider, for instance, a financial trading scenario, where an agent observes only market indicators like prices and volumes, while the latent variables that truly drive market dynamics-such as sentiment or company health-remain hidden.

This chapter investigates how unsupervised pre-training techniques can be extended to handle partial observability, through the lens of state entropy maximization. Beyond the intellectual appeal of optimizing unobservable quantities, we argue that this line of work is essential for bridging the gap between recent theoretical insights and their practical deployment in real-world systems. In Section 4.1, we analyze the fundamental limitations of directly maximizing entropy over raw observations, and in Section 4.2, we introduce scalable solutions that sidestep these limitations and are better suited for practical implementation.

This chapter is based on two papers:*"The Limits of Pure Exploration in POMDPs: When the Observation Entropy is Enough"*, co-authored with D. Cirino, M. Restelli, and M. Mutti, published at RLC 2024 (Section 4.1); *"How to Explore with Belief: State Entropy Maximization in POMDPs"*, with the same authors, published at ICML 2024 (Section 4.2).[1]

[1] A complete reference can be found in the bibliography [Zamboni et al., 2024a,b]

Convex Formulation of Partially Observable MDPs

We begin by introducing a general framework that adapts the convex RL formalism to POMDPs, allowing for the optimization of complex functionals over the (unobserved) state distribution.

Convex Partially Observable Markov Decision Processes (cPOMDP).
A cPOMDP is defined as a tuple $\mathcal{M}^{\mathcal{F}} := (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathbb{P}, \mathbb{O}, \mathcal{F}, T, \mu)$, where $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathbb{P}, \mathbb{O}, T, \mu)$ constitutes a standard POMDP without rewards, and $\mathcal{F}$ is a concave utility function bounded by some finite constant F. Thus, a cPOMDP augments the classical POMDP formulation with a non-linear objective $\mathcal{F}(\cdot)$ defined over the latent state distribution.

In the following sections, we will formally define the domain over which the concave utility function $\mathcal{F}$ operates, and analyze how this choice influences the resulting unsupervised pre-training process.

4.1 The Intrinsic Limits of Observations

While investigating the nature and properties of state entropy maximization in cPOMDPs, first of all we aim to answer to the following question:

Can we maximize the entropy over states getting partial observations only?

4.1.1 Problem Formulation

In the MDP setting observations coincide with the true states of the state of the system, and Hazan et al. [2019] have formulated the *Maximum State Entropy* (MSE) objective as a special case of a Convex RL problem as follows

$$\max_{\pi \in \tilde{\Pi}} \left\{ \mathcal{F}(d_{\mathcal{S}}^{\pi}) := -\sum\nolimits_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s) \log d_{\mathcal{S}}^{\pi}(s) = \mathcal{H}(d_{\mathcal{S}}^{\pi}) \right\}, \tag{4.1}$$

where $\tilde{\Pi} \subseteq \Delta_{\mathcal{S}}^{\mathcal{A}}$ is the set of Markovian policies from states to distribution over actions, and $\mathcal{F}(d_{\mathcal{S}}^{\pi})$ is the convex objective supported on the state distribution "conditioned" on running the policy π in the MDP.

In principle, we aim to address the same objective (4.1) in cPOMDPs as well. However, in the cPOMDP setting, we cannot access the true states, which are latent, but we have to rely on partial observations generated from those states. Thus, a straightforward adaptation of Eq. (4.1) to cPOMDPs is to define an analogous objective on observations as a proxy for $\mathcal{F}(d_{\mathcal{S}}^{\pi})$, which we cannot access. We define the *Maximum Observation Entropy* (MOE) objective as follows

$$\max_{\pi \in \Pi} \left\{ \mathcal{F}(d_{\mathcal{O}}^{\pi}) := -\sum\nolimits_{o \in \mathcal{O}} d_{\mathcal{O}}^{\pi}(o) \log d_{\mathcal{O}}^{\pi}(o) = \mathcal{H}(d_{\mathcal{O}}^{\pi}) \right\}, \tag{4.2}$$

where $\Pi \subseteq \Delta_{\mathcal{O}}^{\mathcal{A}}$ is the set of Markovian policies from observations to distribution over actions, and $\mathcal{F}(d_{\mathcal{O}}^{\pi})$ is the convex objective supported on the observation distribution "conditioned" on running the policy π in the cPOMDP.

Similarly, as in MDPs, we aim to find a policy π that maximizes (4.2), but we are actually interested in achieving a good performance on (4.1). It is easy to see how the value of (4.2) can depart significantly from the true objective (4.1). Take, for instance, an observation matrix that maps every state to the same observation $\mathbb{O}(\bar{o}|s) = 1 \, \forall s \in \mathcal{S}$. It is clear that every policy is optimal for MOE in this setting, but the entropy on the true states can be arbitrarily bad. While those extreme cases are rather unrealistic, the observation matrix can be truly messed up in practice. We want to understand what are the settings that are worth addressing with MOE and what kind of guarantees we can get. As in cMDPs, again, we call the problem (4.2) the *infinite trials* RL formulation for cPOMDPs. Indeed, the objective $\mathcal{F}(d^\pi_\mathcal{O})$ considers the performance that we can achieve on the average of an infinite number of episodes drawn with π. However, in practice, we can never draw infinitely many episodes following a policy π. Instead, we draw a small batch of episodes and obtain an empirical distribution over observation $d_{n,\mathcal{O}} \sim p^\pi_{n,\mathcal{O}}$. Thus, we can instead conceive a *finite trials* RL formulation that is closer to what is optimized in practice:

$$\max_{\pi \in \Pi} \left\{ \mathop{\mathbb{E}}_{d_{n,\mathcal{O}} \sim p^\pi_\mathcal{O}} \mathcal{F}(d_{n,\mathcal{O}}) \right\} := \mathcal{J}_{n,\mathcal{O}}(\pi) \tag{4.3}$$

Remarkably, looking at the single trial formulation ($n = 1$), one should notice that it is compatible with a trajectory-based characterization:

$$\max_{\pi \in \Pi} \left\{ \mathop{\mathbb{E}}_{\mathbf{o} \sim p^\pi_{\mathcal{O},1}} \mathcal{F}(d_\mathcal{O}(\cdot|\mathbf{o})) \right\} := \mathcal{J}_{1,\mathcal{O}}(\pi), \tag{4.4}$$

where $d_\mathcal{O}(\cdot|\mathbf{o})$ is the empirical distribution induced by the observation trajectory $\mathbf{o}$. In the following, we will mostly focus on optimizing the latter objective, as it is the one that we are often asked to optimize in practice: reaching good performance over a single interaction with the environment is often a more realistic goal than optimizing the average performance over many episodes.

4.1.2 A Formal Characterization of Maximum Observation Entropy

In this section, we aim to characterize the gap $\mathcal{F}(d^\pi_\mathcal{S}) - \mathcal{F}(d^\pi_\mathcal{O})$ induced by a chosen policy π, e.g., the policy that maximizes the MOE objective (4.2), when the function is set to be the entropy function, namely when $\mathcal{F}(d^\pi_\mathcal{O}) = \mathcal{H}(d^\pi_\mathcal{O})$. Due to the POMDP nature, in which only partial information (if any) on the true states is leaked to the agent, we cannot provide any general guarantee on the latter gap, which can be as large as

$$|\mathcal{H}(d^\pi_\mathcal{S}) - \mathcal{H}(d^\pi_\mathcal{O})| \leqslant \max\{\log|\mathcal{S}|, \log|\mathcal{O}|\}. \tag{4.5}$$

Nonetheless, we can provide *instance-dependent* results that formally characterize the gap according to notable properties of the observation function in the given instance. First, we prove the following.

Theorem 4.1.1 (Spectral Approximation Bounds)**.** *Let $\mathcal{M}^{\mathcal{F}}$ a cPOMDP and let $\pi \in \Pi \subseteq \Delta_{\mathcal{O}}^{\mathcal{A}}$ a policy. Let the objective function be the entropy function, $\mathcal{F}(d_{\mathcal{O}}^{\pi}) = \mathcal{H}(d_{\mathcal{O}}^{\pi})$. Then, it holds*

$$\log\left(\frac{1}{\sigma_{\max}(\mathbb{O}^{\circ-1})}\right) \leqslant \mathcal{H}(d_{\mathcal{S}}^{\pi}) - \mathcal{H}(d_{\mathcal{O}}^{\pi}) \leqslant \log(\sigma_{\max}(\mathbb{O})).$$

Proof. First, we derive the upper bound. Starting from $\mathcal{H}(d_{\mathcal{O}}^{\pi})$, we have

$$\begin{aligned}
\mathcal{H}(d_{\mathcal{O}}^{\pi}) \geqslant &\mathcal{H}_2(d_{\mathcal{O}}^{\pi}) = \log\left(\frac{1}{\|d_{\mathcal{O}}^{\pi}\|_2}\right) = \log\left(\frac{1}{\|\mathbb{O} \cdot d_{\mathcal{S}}^{\pi}\|_2}\right) && (4.6)\\
&\geqslant \log\left(\frac{1}{\|\mathbb{O}\|_2 \, \|d_{\mathcal{S}}^{\pi}\|_2}\right) = \log\left(\frac{1}{\|d_{\mathcal{S}}^{\pi}\|_2}\right) + \log\left(\frac{1}{\|\mathbb{O}\|_2}\right) && (4.7)\\
&= \mathcal{H}(d_{\mathcal{S}}^{\pi}) - \log\left(\sigma_{\max}(\mathbb{O})\right) && (4.8)
\end{aligned}$$

where the first inequality comes from $\mathcal{H}(V) \geqslant \mathcal{H}_2(V)$ for every variable V and the second inequality from $\|\mathbb{V} \cdot v\|_2 \leqslant \|\mathbb{V}\|_2 \, \|v\|_2$ for every matrix $\mathbb{V}$ and vector v. Then, starting from $\mathcal{H}(d_{\mathcal{S}}^{\pi})$, we get

$$\begin{aligned}
\mathcal{H}(d_{\mathcal{S}}^{\pi}) &= \|d_{\mathcal{S}}^{\pi}\|_\infty \log\left(\frac{1}{\|d_{\mathcal{S}}^{\pi}\|_\infty}\right) + \sum_{s : d_{\mathcal{S}}^{\pi}(s) < \|d_{\mathcal{S}}^{\pi}\|_\infty} d_{\mathcal{S}}^{\pi}(s) \log\left(\frac{1}{d_{\mathcal{S}}^{\pi}(s)}\right) && (4.9)\\
&\leqslant \|d_{\mathcal{S}}^{\pi}\|_\infty \mathcal{H}_\infty(d_{\mathcal{S}}^{\pi}) + (1 - \|d_{\mathcal{S}}^{\pi}\|_\infty) \log\left(\frac{|\mathcal{S}| - 1}{1 - \|d_{\mathcal{S}}^{\pi}\|_\infty}\right) && (4.10)
\end{aligned}$$

where the inequality is obtained by letting $d_{\mathcal{S}}^{\pi}$ be uniformly distributed outside of the entry $\|d_{\mathcal{S}}^{\pi}\|_\infty$. By noting $\mathcal{H}_\infty(V) \leqslant \mathcal{H}_2(V)$ and plugging (4.9) back to (4.6) we get

$$\mathcal{H}(d_{\mathcal{O}}^{\pi}) \geqslant \frac{\mathcal{H}(d_{\mathcal{S}}^{\pi})}{\|d_{\mathcal{S}}^{\pi}\|_\infty} + \frac{\|d_{\mathcal{S}}^{\pi}\|_\infty - 1}{\|d_{\mathcal{S}}^{\pi}\|_\infty} \log\left(\frac{|\mathcal{S}| - 1}{1 - \|d_{\mathcal{S}}^{\pi}\|_\infty}\right) + \log\left(\frac{1}{\sigma_{\max}(\mathbb{O})}\right) \qquad (4.11)$$

which gives the result for $\|d_{\mathcal{S}}^{\pi}\|_\infty \to 1$.[2]

To derive the lower bound, we proceed as follows. We start from the $\mathcal{H}(d_{\mathcal{O}}^{\pi})$ definition to write

$$\begin{aligned}
\mathcal{H}(d_{\mathcal{O}}^{\pi}) &= \sum_{o \in \mathcal{O}} d_{\mathcal{O}}^{\pi}(o) \log\left(\frac{\sum_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s)}{\sum_s d_{\mathcal{S}}^{\pi}(s) \mathbb{O}(o|s)}\right) \sum_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s) && (4.12)\\
&\leqslant \sum_{o \in \mathcal{O}} d_{\mathcal{O}}^{\pi}(o) \sum_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s) \log\left(\frac{d_{\mathcal{S}}^{\pi}(s)}{d_{\mathcal{S}}^{\pi}(s) \mathbb{O}(o|s)}\right) && (4.13)\\
&= \mathcal{H}(d_{\mathcal{S}}^{\pi}) + \sum_{o \in \mathcal{O}} d_{\mathcal{O}}^{\pi}(o) \sum_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s) \log\left(\frac{d_{\mathcal{S}}^{\pi}(s)}{\mathbb{O}(o|s)}\right) && (4.14)\\
&\leqslant \mathcal{H}(d_{\mathcal{S}}^{\pi}) + \mathop{\mathbb{E}}_{o \sim d_{\mathcal{O}}^{\pi}} \mathop{\mathbb{E}}_{s \sim d_{\mathcal{S}}^{\pi}} \left[\log(\mathbb{O}^{\circ-1}(o|s))\right] && (4.15)\\
&\leqslant \mathcal{H}(d_{\mathcal{S}}^{\pi}) + \log\left(\max_{o \in \mathcal{O}} \max_{s \in \mathcal{S}} \mathbb{O}^{\circ-1}(o|s)\right) && (4.16)\\
&\leqslant \mathcal{H}(d_{\mathcal{S}}^{\pi}) + \log\left(\sigma_{\max}(\mathbb{O}^{\circ-1})\right) && (4.17)
\end{aligned}$$

[2] Note that (4.11) is a tighter version of the upper bound than the one provided in the theorem statement, although it directly depends on the state distribution $d_{\mathcal{S}}^{\pi}$ beyond spectral properties of $\mathbb{O}$.

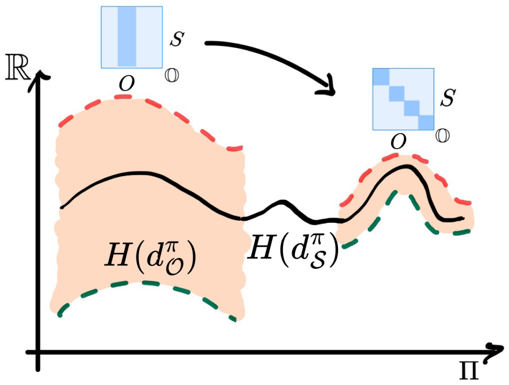


Figure 4.1: *Spectral Bound behavior for two different observation matrices $\mathbb{O}$. MOE values compatible with MSE values are in orange.*

where we exploit $d_{\mathcal{O}}^{\pi}(o) = \sum_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s)\mathbb{O}(o|s)$ and $\sum_{s \in \mathcal{S}} d_{\mathcal{S}}^{\pi}(s) = 1$ to write (4.12), we first apply the log-sum inequality and we split the logarithm to get (4.14). Then, in (4.16), we write the first inequality through the definition of the Hadamard inverse of $\mathbb{O}$ and noting that $d_{\mathcal{S}}^{\pi}(s) \leqslant 1 \ \forall s \in \mathcal{S}$, we get the second inequality from $\mathbb{E}[V] \leqslant \max(V)$ for any random variable V and the monotonicity of the logarithm. Finally, we obtain the result (4.17) by $\|\mathbb{V}\|_{\infty} \leqslant \|\mathbb{V}\|_2 = \sigma_{\max}(\mathbb{V})$ for any matrix $\mathbb{V}$. □

Theorem 4.1.1 gives bounds on the approximation gap that can be much tighter than the worst-case gap in Eq. (4.5). The bounds relate the gap to the scale of the transformation induced by the observation matrix on the distribution of the latent states, which is captured by the maximum singular value of $\mathbb{O}$ and $\mathbb{O}^{\circ -1}$, respectively. For instance, an observation matrix that maps every state to the same observation $\mathbb{O}(\bar{o}|s) = 1 \ \forall s \in \mathcal{S}$ can lead to a larger gap between MOE and MSE, as visualized in the left-hand side of Figure 4.1.

On the other hand, when the observation matrix maps with high probability each state to a different observation, the gap is necessarily smaller (see the right-hand side of Figure 4.1). Notably, both sides of the bound collapse to zero when the observation matrix is an identity matrix, i.e., when the states are fully observed.

The bounds in Theorem 4.1.1 only focus on spectral properties of the observation matrix $\mathbb{O}$. In a similar vein, we can provide an analogous characterization based on information properties of $\mathbb{O}$.

Theorem 4.1.2 (Information Approximation Bound)**.** *Let $\mathcal{M}^{\mathcal{F}}$ a cPOMDP, let $\pi \in \Pi \subseteq \Delta_{\mathcal{O}}^{\mathcal{A}}$ a policy, and let $\mathcal{H}(d_{\mathcal{O}}^{\pi}|d_{\mathcal{S}}^{\pi}) = \mathbb{E}_{s \sim d_{\mathcal{S}}^{\pi}}[\mathcal{H}(\mathbb{O}(\cdot|s))]$. Let the objective function be the entropy function, $\mathcal{F}(d_{\mathcal{O}}^{\pi}) = \mathcal{H}(d_{\mathcal{O}}^{\pi})$. Then, it holds*

$$\mathcal{H}(d_{\mathcal{S}}^{\pi}) \geqslant \mathcal{H}(d_{\mathcal{O}}^{\pi}) - \mathcal{H}(d_{\mathcal{O}}^{\pi}|d_{\mathcal{S}}^{\pi}).$$

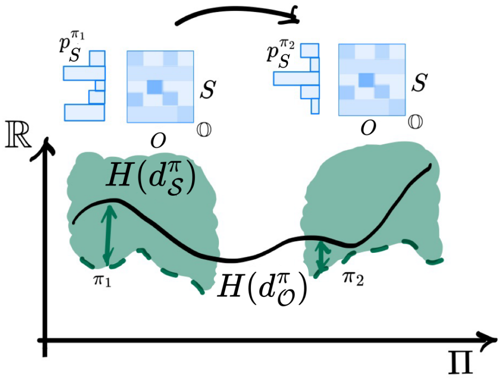


Figure 4.2: *Information Approximation Bound behavior for two different $d^\pi_\mathcal{S}$. MSE values compatible with MOE values are in green.*

Proof. Starting from $\mathcal{F}(d^\pi_\mathcal{O})$ definition, we can write

$$\mathcal{F}(d^\pi_\mathcal{O}) = \sum_{o\in\mathcal{O}} d^\pi_\mathcal{O}(o)\log\frac{1}{d^\pi_\mathcal{O}(o)} = \sum_{o\in\mathcal{O}}\sum_{s\in\mathcal{S}} \mathbb{O}(o|s)d^\pi_\mathcal{S}(s)\log\frac{1}{\sum_{s'\in\mathcal{S}}\mathbb{O}(o|s')d^\pi_\mathcal{S}(s')} \tag{4.18}$$

$$\leqslant \sum_{o\in\mathcal{O}}\sum_{s\in\mathcal{S}} \mathbb{O}(o|s)d^\pi_\mathcal{S}(s)\log\frac{1}{\mathbb{O}(o|s)d^\pi_\mathcal{S}(s)} \tag{4.19}$$

$$= \sum_{o\in\mathcal{O}}\sum_{s\in\mathcal{S}} \mathbb{O}(o|s)d^\pi_\mathcal{S}(s)\log\frac{1}{d^\pi_\mathcal{S}(s)} + \sum_{o\in\mathcal{O}}\sum_{s\in\mathcal{S}} \mathbb{O}(o|s)d^\pi_\mathcal{S}(s)\log\frac{1}{\mathbb{O}(o|s)} \tag{4.20}$$

$$= \mathcal{H}(d^\pi_\mathcal{S}) + \sum_{s\in\mathcal{S}} d^\pi_\mathcal{S}(s)H(\mathbb{O}(\cdot|s)) = \mathcal{H}(d^\pi_\mathcal{S}) + \mathcal{H}(d^\pi_\mathcal{O}|d^\pi_\mathcal{S}) \tag{4.21}$$

where we get (4.19) by noting $\sum_{s'\in\mathcal{S}}\mathbb{O}(o|s')d^\pi_\mathcal{S}(s') \geqslant \mathbb{O}(o|s)d^\pi_\mathcal{S}(s)$, we split the logarithm to write (4.20), we let $\sum_{o\in\mathcal{O}}\mathbb{O}(o|s) = 1$ and $\sum_{s\in\mathcal{S}} d^\pi_\mathcal{S}(s)H(\mathbb{O}(\cdot|s)) = \mathcal{H}(d^\pi_\mathcal{O}|d^\pi_\mathcal{S})$ to obtain the result in (4.21). □

Theorem 4.1.2 essentially states that the gap between the entropy on observations and true states is small as long as the policy π induces visits to states where the observation function has low entropy, which is captured by the term $\mathcal{H}(d^\pi_\mathcal{O}|d^\pi_\mathcal{S}) = \mathbb{E}_{s\sim d^\pi_\mathcal{S}}[\mathcal{H}(\mathbb{O}(\cdot|s))]$. When a policy visits states emitting observations with high entropy, the bound on the gap will be loose, as visualized in the left-hand side of Figure 4.2. Instead, when the most visited states emit almost deterministic observations, then the bound on the gap is tighter (see the right-hand side in Figure 4.2). This latter bound is tight when the true states are fully observed, collapsing the gap to zero.

The combination of Theorems 4.1.1, 4.1.2 yield a nice description of the instances that is reasonable to address with a MOE approach, i.e., those for which the gap between the resulting policy and the optimal MSE policy is small thanks to the properties of the observation matrix. Unfortunately, policies in POMDPs have control over neither the spectral properties of the observation function nor whether the visited states have low-entropy observation distributions. In other words, while being descriptive, these results do not provide any further tool to actively address MSE in cPOMDPs. In the next section, we reformulate the bound in Theorem 4.1.2 around quantities that can be actively controlled by a policy conditioned on observations and we provide a family of policy gradient algorithms to learn a MOE policy in those relevant instances.

Before diving into algorithmic solutions, it is interesting to confront the properties making a state entropy maximization problem on cPOMDPs easy and analogous requirements for RL in POMDPs. In the latter setting, we generally ask for an observation function that leaks significant information on the latent state. For instance, this is captured by a lower bound on the minimum singular value of $\mathbb{O}$ in the *revealing* POMDP assumption [Liu et al., 2022a]. Instead, in state entropy maximization, we care less about identifying the latent state, and we can just focus on observations as long as $\mathbb{O}$ does not dramatically jeopardize the underlying state distribution.

4.1.3 Towards Principled Policy Gradients

In the previous section, we analyzed the theoretical guarantees we get on the state entropy maximization problem by optimizing the MOE objective (4.2), but we did not yet describe how the latter optimization can be performed. Here we propose a family of Policy Gradient algorithms [Kober and Peters, 2008] to learn a MOE policy from sampled interactions with the cPOMDP.

First, we define a space of *parametric* policies $\pi_\theta \in \Pi_\Theta \subseteq \Pi$ where $\theta \in \Theta \subseteq \mathbb{R}^{|\mathcal{O}||\mathcal{A}|}$ are differentiable policy parameters.[3] The expression of the MOE objective does not allow for an easy computation of policy gradients. However, if we take into account the single-trial formulation of MOE, namely as in Eq. (4.3) by setting $n = 1$, we can easily derive a policy gradient formulation:

Proposition 4.1.3 (Policy Gradient for single-trial cPOMDPs)**.** *Let* $\pi_\theta \in \Pi_\Theta$ *a parametric policy and let the* policy scores $\nabla_\theta \log \pi_\theta(\mathbf{o}, \mathbf{a}) = \sum_{t\in[T]} \nabla_\theta \log \pi_\theta(\mathbf{a}[t]|\mathbf{o}[t])$. *We can compute the* policy gradient *of* π_θ *as*

$$\nabla_\theta \mathcal{J}_{1,\mathcal{O}}(\pi_\theta) = \mathop{\mathbb{E}}_{\mathbf{oa}\sim p^{\pi_\theta}_{\mathcal{OA},1}} \Big[\nabla_\theta \log \pi_\theta(\mathbf{o}, \mathbf{a}) \mathcal{F}(d_\mathcal{O}(\cdot|\mathbf{o}))\Big], \tag{4.22}$$

where $d_\mathcal{O}(\cdot|\mathbf{o})$ *is the empirical distribution induced by the observation trajectory* $\mathbf{o}$.

Notably, the trajectory-based objective (4.3) is a lower bound to the MOE objective (4.2), due to the concavity of the entropy function and the Jensen's inequality [Mutti et al., 2022a]. Thus, optimizing for (4.3) guarantees a non-degradation of our initial objective function (4.2), while it allows for an easy derivation of the gradient ∇_θ w.r.t. the policy parameters, as reported here:

Proof. We write

$$\begin{aligned}
\nabla_\theta \mathcal{J}_{1,\mathcal{O}}(\pi_\theta) &= \nabla_\theta \sum\nolimits_{(\mathbf{oa})\in\mathcal{O}^T\times\mathcal{A}^T} p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa}) \mathcal{F}(d_\mathcal{O}(\cdot|\mathbf{o})) && (4.23)\\
&= \sum\nolimits_{(\mathbf{oa})\in\mathcal{O}^T\times\mathcal{A}^T} \Big(\nabla_\theta p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa})\Big) \mathcal{F}(d_\mathcal{O}(\cdot|\mathbf{o})) && (4.24)\\
&= \sum\nolimits_{(\mathbf{oa})\in\mathcal{O}^T\times\mathcal{A}^T} p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa}) \nabla_\theta \log p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa}) \mathcal{F}(d_\mathcal{O}(\cdot|\mathbf{o})) && (4.25)\\
&= \mathop{\mathbb{E}}_{\mathbf{oa}\sim p^{\pi_\theta}_{\mathcal{OA},1}} \big[\nabla_\theta \log p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa}) \mathcal{F}(d_\mathcal{O}(\cdot|\mathbf{o}))\big] && (4.26)
\end{aligned}$$

[3] See [Deisenroth et al., 2013, Section 1.3] for common choices of parametric policy spaces.

by exploiting the linearity of the expectation to go from the first to the second equality, then applying the common log-trick [Kober and Peters, 2008] and finally recognizing the sum as an expectation again.

To derive the gradient we then have to provide the calculation of the *policy scores* $\nabla_\theta \log p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa})$. For every $\pi \in \Pi_\Theta$, we notice that $p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa}) = \prod_{t\in[T]} Pr(o_t = \mathbf{o}[t])\pi_\theta(a_t = \mathbf{a}[t]|o_t = \mathbf{o}[t])$ and that the only term depending on θ is the policy itself. By exploiting the properties of the logarithm we have

$$\nabla_\theta \log p^{\pi_\theta}_{\mathcal{OA},1}(\mathbf{oa}) = \sum_{t\in[T]} \nabla_\theta \log \pi_\theta(a_t = \mathbf{a}[t]|o_t = \mathbf{o}[t]) = \nabla_\theta \log \pi_\theta(\mathbf{oa}) \tag{4.27}$$

which leads to the standard REINFORCE formulation [Williams, 1992]. □

With the latter result, we can design a policy gradient algorithm based on REINFORCE [Williams, 1992]. The procedure, described in Algorithm 4.1.3, initializes the policy parameters and then performs several iterations of gradient ascent updates. As we shall see in the next section, Algorithm 4.1.3 can be a simple yet effective solution to MOE optimization in various settings. However, the resulting policy can be underwhelming in domains where the observation matrix is particularly challenging. While we cannot overcome the barriers established in Theorems 4.1.1, 4.1.2, we can still exploit additional information on the observation function to further improve the performance.

Algorithm 4.1.3: PG for MOE (**Reg-MOE**)

```
Input: learning rate $\alpha$, number of iterations $K$, batch size $N$
Initialize the policy parameters $\theta_1$
for $k = 1, \dots, K$ do
    Sample $N$ trajectories $\{(\mathbf{o}_i, \mathbf{a}_i)\}_{i\in[N]}$ with the policy $\pi_{\theta_k}$
    Compute $\{\mathcal{H}(d_{\mathcal{O}}(\cdot|\mathbf{o}_i))\}_{i\in[N]}$ and $\{\nabla_\theta \log \pi_\theta(\mathbf{o}_i\mathbf{a}_i)\}_{i\in[N]}$
    Update the policy parameters in the gradient direction:

$\theta_{k+1} \leftarrow \theta_k + \alpha\frac{1}{N}\sum_{i\in[N]} \nabla_\theta \log \pi_\theta(\mathbf{o}_i\mathbf{a}_i)(\mathcal{H}(d_{\mathcal{O}}(\cdot|\mathbf{o}_i)) - \beta \sum_{o\in\mathcal{O}} d_{1,\mathcal{O}}(o)\mathcal{H}(\mathbb{O}(o|\cdot)))$

end for
Output: the final policy $\pi_{\theta_K}$
```

Known Observation Matrix

With the knowledge of $\mathbb{O}$, we are tempted to directly optimize the lower bound to $\mathcal{H}(d^\pi_{\mathcal{S}})$ provided in Theorem 4.1.2 by trading-off high entropy on observations ($\mathcal{H}(d^\pi_{\mathcal{O}})$) with the entropy of their emission ($\mathcal{H}(d^\pi_{\mathcal{O}}|d^\pi_{\mathcal{S}})$). Unfortunately, we do not have access to the state distribution $d^\pi_{\mathcal{S}}$ to compute the expectation $\mathcal{H}(d^\pi_{\mathcal{O}}|d^\pi_{\mathcal{S}}) = \mathbb{E}_{s\sim d^\pi_{\mathcal{S}}}[\mathcal{H}(\mathbb{O}(\cdot|s))]$. Nonetheless, we can rework the lower bound into an alternative form where all of the terms are known and can be controlled by a policy conditioned on observations only, as it demonstrates the following corollary to Theorem 4.1.2.

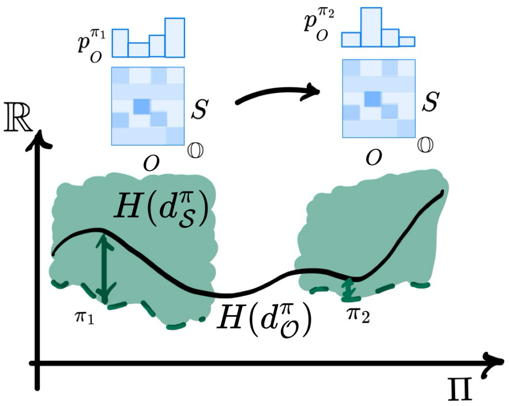


Figure 4.3: *Actionable Lower Bound behavior for two different* $d_{\mathcal{O}}^\pi$. *MSE values compatible with MOE values are in green.*

Corollary 4.1.4 (Actionable Lower Bound)**.** *Let* $\mathcal{M}^{\mathcal{F}}$ *a cPOMDP, let* $\pi \in \Pi \subseteq \Delta_{\mathcal{O}}^{\mathcal{A}}$ *a policy, and let* $\mathcal{H}(d_{\mathcal{S}}^\pi|d_{\mathcal{O}}^\pi) = \mathbb{E}_{o\sim d_{\mathcal{O}}^\pi}[\mathcal{H}(\mathbb{O}(o|\cdot))]$. *Let the objective function be the entropy function,* $\mathcal{F}(d_{\mathcal{O}}^\pi) = \mathcal{H}(d_{\mathcal{O}}^\pi)$. *Then, it holds*

$$\mathcal{H}(d_{\mathcal{S}}^\pi) \geqslant \mathcal{H}(d_{\mathcal{O}}^\pi) - \mathcal{H}(d_{\mathcal{S}}^\pi|d_{\mathcal{O}}^\pi) + \log(\sigma_{\max}(\mathbb{O})). \tag{4.28}$$

Proof. The result follows through further manipulation of Theorem 4.1.2. We have,

$$\mathcal{H}(d_{\mathcal{S}}^\pi) \geqslant \mathcal{H}(d_{\mathcal{O}}^\pi) - \mathcal{H}(d_{\mathcal{O}}^\pi|d_{\mathcal{S}}^\pi) = \mathcal{H}(d_{\mathcal{O}}^\pi) - \mathcal{H}(d_{\mathcal{S}}^\pi|d_{\mathcal{O}}^\pi) + \mathcal{H}(d_{\mathcal{S}}^\pi) - \mathcal{H}(d_{\mathcal{O}}^\pi) \tag{4.29}$$

$$\geqslant \mathcal{H}(d_{\mathcal{O}}^\pi) - \sum_{o\in\mathcal{O}} d_{\mathcal{O}}^\pi(o)\mathcal{H}(\mathbb{O}(o|\cdot)) + \log(\sigma_{\max}(\mathbb{O})) \tag{4.30}$$

where (4.29) is the result of the application of the Bayes rule to the conditional entropy $\mathcal{H}(d_{\mathcal{O}}^\pi|d_{\mathcal{S}}^\pi)$ and (4.30) follows from the fact that $\mathcal{H}(d_{\mathcal{O}}^\pi) - \mathcal{H}(d_{\mathcal{S}}^\pi) \geqslant -\log(\sigma_{\max}(\mathbb{O}))$ due to Theorem 4.1.1. □

From the latter result, we get a lower bound to $\mathcal{H}(d_{\mathcal{S}}^\pi)$ that can be controlled, as we flipped the conditioning from $\mathcal{H}(d_{\mathcal{O}}^\pi|d_{\mathcal{S}}^\pi)$ to $\mathcal{H}(d_{\mathcal{S}}^\pi|d_{\mathcal{O}}^\pi)$, which we can compute by taking an expectation with the observation distribution. Visually, when a policy visits observations that can be emitted by many states, the bound on the gap will be looser (Figure 4.3, left-hand side). When the visited observations are emitted by specific states with high probability, then the bound on the gap is tighter (Figure 4.3, right-hand side).

Inspired by the rationale provided by this bound, it is then possible to explicitly account for the effect of dealing with observation only: for every $\beta \in (0, 1)$, we can write a regularized version of (4.3) as

$$\max_{\pi\in\Pi} \left\{ \mathop{\mathbb{E}}_{d_{n,\mathcal{O}}\sim p_{\mathcal{O}}^\pi} \mathcal{H}(d_{n,\mathcal{O}}) - \beta \sum\nolimits_{o\in\mathcal{O}} d_{n,\mathcal{O}}(o)\mathcal{H}(\mathbb{O}(o|\cdot))\Big) \right\} := \mathcal{J}_{n,\mathcal{O}}^\beta(\pi) \tag{4.31}$$

which we call *Regularized MOE* (Reg-MOE), and a slight variation of Alg. 4.1.3 (highlighted in the pseudocode) to optimize the regularized objective. In the next section, we provide an empirical validation of the proposed PG algorithms to describe their respective strengths and weaknesses. Note that the presented algorithms can be further

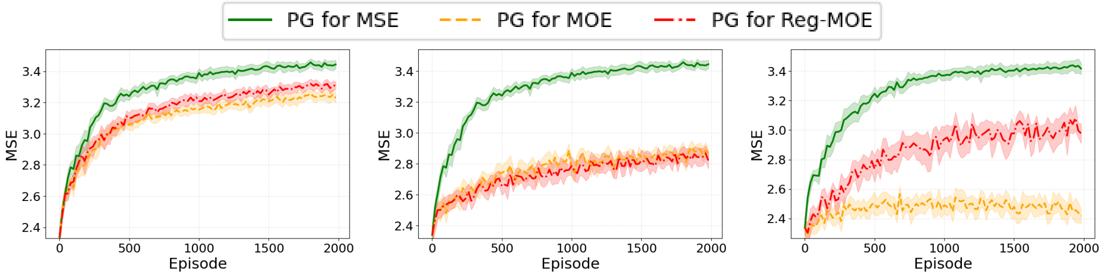


Figure 4.4: *Well-behaved observations with* $\mathbb{E}[H(\mathbb{O})] \approx 1$

Figure 4.5: *Challenging observations with* $\mathbb{E}[\mathcal{H}(\mathbb{O})] \approx 2.2$

Figure 4.6: *Challenging observations with structure* $\mathbb{E}[\mathcal{H}(\mathbb{O})] \approx 1.85$

Entropy on latent states (MSE) achieved by PG for MSE, PG for MOE, *and* PG for Reg-MOE *in gridworlds with various* $\mathbb{O}$. *We report the average and 95% c.i. over 16 runs.*

enhanced with the same technical solutions of advanced policy optimization algorithms for the MSE objective [Mutti et al., 2021, Liu and Abbeel, 2021b, Seo et al., 2021, Yarats et al., 2021] to address continuous and high-dimensional domains.

4.1.4 Numerical Validation

Here we provide a brief numerical validation of the theoretical results provided in Section 4.1.2 and the algorithmic solutions proposed in Section 5.3. Especially, we aim to show that

(a) Optimizing MOE is particularly effective when the observation matrix is "well-behaved";

(b) Optimizing MOE is bound to fail when the observation matrix is not "well-behaved";

(c) Additional knowledge of the observation structure can be sometimes exploited to improve the performance in the latter challenging cases by optimizing the regularized MOE.

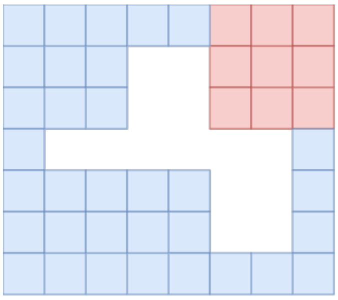
Intuitively, an observation matrix is "well-behaved" when it does not induce a significant transformation of the state distribution, keeping the approximation gap between MOE and MSE small. Thanks to Theorems 4.1.1, 4.1.2 we can provide a formal characterization of this property. In the experiments below, we measure the latter through the average entropy of the observation function $\mathbb{E}[\mathcal{H}(\mathbb{O})] = \sum_{s \in \mathcal{S}} \mathcal{H}(\mathbb{O}(\cdot|s))/|\mathcal{S}|$ on the lines of the information bound in Theorem 4.1.2.

In Figure 4.4 we test (a) by showing that the performance of the algorithms accessing observations only, i.e., *PG for MOE* and *PG for Reg-MOE*, is remarkably close to the ideal baseline having access to the true states, i.e., *PG for MSE*. This is due to the low average entropy of the observation function: Although the agent cannot know its exact position, maximizing the entropy of observations still leads to a large entropy over the latent states.

This is not the case in the experiment in Figure 4.5, where the gridworld configuration is the same, but the observation function is now more challenging, i.e., more entropic on average. The significant gap between the algorithms optimizing MOE and the ideal baseline is a testament of (b) and a corroboration of the theoretical limits of the MOE approach, which are formally provided in Theorems 4.1.1, 4.1.2. *PG for MOE* and *PG for Reg-MOE* can still successfully maximize the entropy over observations, but cannot avoid a significant mismatch with the resulting entropy over latent states.

However, not all the domains with challenging (i.e., entropic) observations are hopeless for the MOE approach, especially when we can exploit knowledge on how the observations are themselves generated. In Figure 4.6, we report a further experiment in which the observation matrix has a block with very high entropy (in which observations are almost random) and a block with nearly deterministic observations. *PG for MOE* does not exploit the structure of $\mathbb{O}$ and cannot distinguish between observations that are *reliable* from those that are not.

Instead, the regularization term in *PG for Reg-MOE* leads to more visitations of reliable observations (i.e., generated with lower entropy) effectively reducing the gap with the ideal baseline (*PG for MSE*), which corroborates both (c) and the result in Corollary 4.1.4.

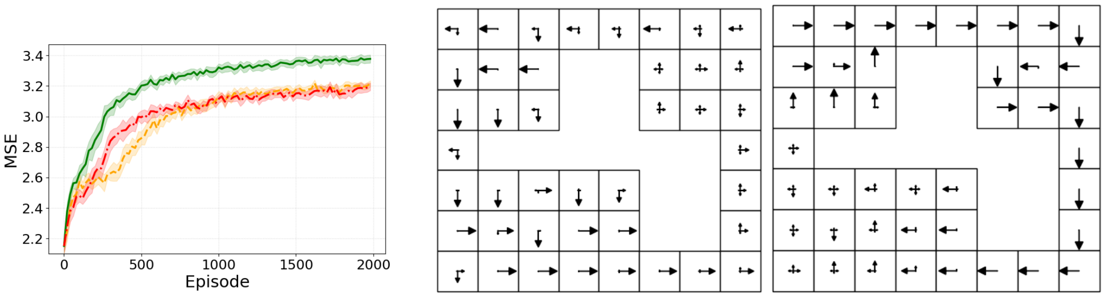


Finally, in order to further investigate the effects of the regularization term, we considered a different grid-world, whose visualization is reported on the right. The observation matrix is designed as a Gaussian $\mathcal{G}(0, \sigma^2)$ over the Manhattan distance in the blue rooms, while it is deterministic (and thus fully revealing) in the red room. For this experiment, we set the variance to $\sigma^2 = 1$, the regularization term to $\beta = 0.3$, and the horizon $T = 40$. As for the remaining parameters, they are kept as in the previous experiments. Figure 4.7 shows how the two learned policies have indeed similar performances. Yet, while the MOE-based policy tries to explore the environment uniformly, the Reg-Moe one successfully explored the portion of the grid with lower entropy in the observations, to later address a deeper exploration of the remaining rooms. This behavior exactly aligns with the role of the regularization term, which should indeed make the agent prefer observations that are emitted with lower entropy by the observation function.

As a bottom line, this numerical validation shows that the MOE approach, while not being a solution to every cPOMDP instance, can still provide a remarkable performance on domains where the observation matrix is not too challenging or when its knowledge can be exploited.

Concluding Remarks

In this section, we made a step forward into generalizing state entropy maximization in cPOMDPs. Specifically, we addressed the problem of learning a policy conditioned only by observations that target the entropy over the latent states. We proposed the simple approach of optimizing the entropy over observations in place of latent states and we formally characterized the instances where it is effective by deriving approximation bounds of the latent objective that depend on the structure of the observation matrix. Finally, we designed a family of policy gradient algorithms to optimize the



MSE performances. *MOE-based Policy.* *Reg-MOE-based Policy.*

Figure 4.7: *Comparison of the policies learned by PG for MOE and PG for Reg-MOE over 2000 episodes. The magnitude of each arrow is proportional to the probability of the policy to choose that action, after marginalizing over all the possible observations emitted in that state.*

observation entropy in practice and to exploit knowledge of the observation structure when available.

It is worth mentioning that state entropy maximization can find further motivation in POMDPs beyond its common use in MDP settings. While how those methods can benefit offline data collection and transition model estimation is less obvious under partial observability, it is worth noting that the reward in a POMDP is usually defined over the true states, such that pre-training a policy to explore over them is still relevant [Eysenbach et al., 2021].

4.2 Explore with Belief

In the previous Section, we noticed how optimizing for objectives over observations only might be problematic when these are not well behaved, and the mismatch between the entropy over observations and true states can be significant in relevant domains (e.g., the rescue operation setting we described above). Additionally, we showed how in order to recover good performances, the specification of the cPOMDP is needed.

This scenario is motivated by domains in which we can train the agent's policy on a simulator of the environment and then deploy the optimal policy in the real world. However, a simulator is not available in all the relevant applications. Can we still learn a reasonable policy in those settings? To overcome this limitation, we can instead compute approximate beliefs from observations [Subramanian et al., 2022] and then optimize the entropy of the states sampled from the beliefs as a *proxy* objective that incorporates all of the information available about the entropy on the true states. In the following section, we show how this option provides a more scalable alternative than the trivial use of entropy of observations in general.

4.2.1 Problem Formulation

As stated before, convex RL in general, and state entropy maximization in particular, is particularly challenging in cPOMDPs as the objective function is defined on a space to which the agent has no direct access. It is clear that the ideal goal of maximizing the objective in Eq. 4.1 as in (fully observable) MDPs is far-fetched under these premises. Addressing convex RL in cPOMDPs includes the following additional and intertwined challenges: **(a)** Defining a proxy objective function compatible with the setting, i.e., on quantities the agent can observe; **(b)** Defining a compact policy class such that policies can be efficiently stored.

In this section, we will build upon the single-trial formulation of the objective, which is closer to the need for practical applications [Mutti et al., 2022b]. Additionally, we notice that the common infinite-trials relaxation considered in previous works [Hazan et al., 2019] is still intractable in cPOMDPs, which leaves minimal benefit over the single-trial formulation for details.

(a) Proxy Objective Functions. Optimizing Eq. (4.1) is ill-posed in cPOMDPs without further assumptions because states are not observed. We then seek to design proxy objectives whose maximization leads to policies with good performance on the (ideal) original objective as well. The first and most intuitive choice is to formulate an analogous objective over observations instead of states. Previously, the (single-trial) *Maximum Observation Entropy* (MOE) objective of Eq. (4.3) was defined as

$$\max_{\pi \in \Pi} \left\{ \mathop{\mathbb{E}}_{\mathbf{o} \sim p^{\pi}_{\mathcal{O},1}} \mathcal{H}(d(\cdot|\mathbf{o})) \right\} := \mathcal{J}_{1,\mathcal{O}}(\pi).$$

While being rather intuitive, this objective is intrinsically problematic. There can be significant mismatches between observation and state spaces. When the POMDPs are under (respectively over) complete [Liu et al., 2022a], i.e., when the number of observations is less (respectively more) than the number of states, it may be hard to link entropy over observations to entropy over states. Moreover, even when $\mathcal{O} = \mathcal{S}$, a random emission function $\mathbb{O}$ could jeopardize any estimate of the state entropy that is based on the entropy of observations. Here, we introduce more reliable proxy objectives in Section 4.2.2, 4.2.3 along with corresponding assumptions on the information available to the agent.

(b) Deployable Policy Classes. So far, we denoted the policy class as $\Pi_{\mathcal{I}}$ for a generic set $\mathcal{I}$ of the available information. An essential point to be addressed in POMDPs is which policy class to use [Cassandra, 1998]. We say a policy class is *deployable* if its policies are conditioned on the information set $\mathcal{I}$ that is available to the agent *at deployment*.[4] We follow a similar definition of deployable policies as for centralized training and decentralized executions in multi-agent settings [Albrecht et al., 2024]. It follows that any policy class over true states is not deployable, and this is the case for deterministic non-Markovian policies as well [Mutti et al., 2022b]. Yet, other policy classes are deployable, e.g., over observations, trajectories of observations, and trajectories of beliefs. Ideally, we want to employ the richer deployable policy class, which is the space of non-Markovian policies over observations (or, equivalently, over beliefs). Unfortunately, a policy in this class cannot be efficiently stored in general, so we will look for restricted classes with more reasonable memory requirements.

4.2.2 Primer: accessing a Simulator

First, we consider a simplified setting where:

Assumption 4.2.1 (Known Model)**.** $\mathbb{P}, \mathbb{O}$ *are fully known in training.*

[4] Even in the case a simulator is available to optimize the policy, we still want to deploy the policy in unknown partially observable environments in general.

This setting encompasses the best-case scenario, in which a (white-box) simulator of the environment is available and the true state of the cPOMDP can be accessed. Even in this simplified setting, the problem is non-trivial. First, it does not reduce to the MDP problem, as we need to learn a deployable policy. Secondly, the best deployable policy class is problematic in terms of memory complexity. Finally, as the theory demonstrates [Papadimitriou and Tsitsiklis, 1987, Mundhenk et al., 2000], even solving a known cPOMDP is computationally intractable. These issues drive the algorithmic choices in the following sense:

1. **Memory complexity.** The policy class will be restricted to memory-efficient policies, such that the policy parameters are polynomial in the size of $\mathcal{M}^{\mathcal{F}}$.

2. **Computational complexity.** A first-order method will be employed, i.e., policy gradient [Williams, 1992, Sutton et al., 1999], to overcome computational hardness.

(1) Unfortunately, the size of $\mathcal{T}_{\mathcal{O}}, \mathcal{T}_{\mathcal{B}}$ is exponential in T, which means that policies over such spaces would require an exponential number of parameters. This leaves the information sets $\mathcal{O}, \mathcal{B}$ as viable options. Similarly, the set $\mathcal{B}$ of belief states reachable in T steps can be extremely large even in simple POMDPs.[5] Policy classes that are efficient to store are $\Pi_{\mathcal{O}}$ and $\Pi_{\tilde{\mathcal{S}}}$, i.e., the set of Markovian policies over observations or believed states. It is known, however, that non-Markovian policies are needed to optimize the single-trial convex objectives in general [Mutti et al., 2022b]. An option is to consider the belief, which is a function of the trajectory over states and actions, as a succinct representation of the history, and then to employ a careful parametrization of the policy to get memory efficiency. Formally, we introduce the *Belief-Averaged* (BA) policy class as $\bar{\Pi}_{\mathcal{B}} := \{\pi \in \bar{\Pi}_{\mathcal{B}} : \pi_\theta(\cdot|\boldsymbol{b}) = \langle \theta, \boldsymbol{b} \rangle\} \subseteq \Delta(\mathcal{A})$.

(2) The optimization problem over the latter policy class will be addressed via first-order methods [Williams, 1992], in order to overcome computational hardness. Previous works have considered policy gradient for MSE in MDPs [Liu and Abbeel, 2021b]. Here, we derive a specialized gradient for the cPOMDP setting when the information set $\mathcal{I}$ is not defined.[6]

Proposition 4.2.1 ((General) Policy Gradient for single-trial cPOMDPs). *Let* $\pi_\theta \in \Pi_{\mathcal{I}}$ *a policy parametrized by* $\theta \in \Theta \subseteq \mathbb{R}^{IA}$*, and let the* policy scores $\nabla_\theta \log \pi_\theta(\mathbf{ia}) = \sum_{t \in [T]} \nabla_\theta \log \pi_\theta(\mathbf{a}[t] | \mathbf{i}[t])$. *We can compute the* policy gradient *of* π_θ *as*

$$\nabla_\theta \mathcal{J}_{1,\mathcal{O}}(\pi_\theta) = \mathop{\mathbb{E}}_{\mathbf{ia} \sim p^{\pi_\theta}_{\mathcal{IA},1}} \Big[\nabla_\theta \log \pi_\theta(\mathbf{ia}) \mathcal{F}(d_{\mathcal{I}}(\cdot|\mathbf{i})) \Big], \tag{4.32}$$

where $\mathcal{I} \in \{\mathcal{S}, \mathcal{O}\}$.

[5] We can compute $\mathcal{B}$ by means of Algorithm B.1 in Appendix B.1.

[6] The full derivation can be found in Appendix B.1.

Algorithm 4.2.2: General **Reg-**PG for cPOMDPs

```
Input: learning rate $\alpha$, initial parameters $\theta_1$, number of episodes $K$, batch size $N$, information set $\mathcal{I} \in \{\mathcal{S}, \mathcal{O}, \tilde{\mathcal{S}}\}$, regularization parameter $\rho$
for $k = 1$ to $K$ do
    Sample $N$ trajectories $\{\mathbf{i}^n \sim p^{\pi_{\theta_k}}\}_{n\in[N]}$
    Compute the feedbacks $\{\mathcal{H}(d_{\mathcal{I}}(\cdot|\mathbf{i}^n))\}_{n\in[N]}$
    Compute $\{\log \pi(\mathbf{i}^n)\}_{n\in[N]}$
    Perform a gradient step:
        $\theta_{k+1} \leftarrow \theta_k + \frac{\alpha}{N} \sum_{n\in[N]} \log \pi(\mathbf{i}^n)[\mathcal{H}(d_{\mathcal{I}}(\cdot|\mathbf{i}^n)) - \rho \sum_t \mathcal{H}(b_t^n)]$
end for
Output: the last-iterate policy $\pi_\theta^K$
```

Algorithmic Architecture

It can be seen that optimizing for different objectives, the policy gradient differs only on the second factor of the product, which we refer to as *feedback*. Thus, we propose a general algorithmic framework, which works for any objective, and mimics the structure of REINFORCE [Williams, 1992]. The pseudocode is reported in Algorithm 4.2.2.[7] The main loop of the algorithm (**2**-**7**) is composed of the main steps: (**3**) sample N trajectories with the current policy, (**4**) extract the feedbacks coherently to the objective being optimized, (**5**) compute the log-policy term and (**6**) perform a gradient ascent step over the parameters space.

Smoothness of the Optimization Landscape

We can prove that the considered objectives are locally smooth, making first-order approaches of the kind described above well-suited for the problem.[8]

Theorem 4.2.2 (Local Lipschitz Constants)**.** *Let* $\pi_1, \pi_2 \in \Pi_{\mathcal{I}}$*, let* $\mathcal{T}_{\mathcal{I}}(\pi_1, \pi_2) = \{\mathbf{i} \in \mathcal{T}_{\mathcal{I}} : p^{\pi_1}(\mathbf{i}) > 0 \vee p^{\pi_2}(\mathbf{i}) > 0\}$ *be the set of realizable trajectories over* $\mathcal{I} \in \{\mathcal{S}, \mathcal{O}\}$*, and let* $\mathbf{i}^\star = \arg\max_{\mathbf{i}\in\mathcal{T}_{\mathcal{I}}(\pi_1,\pi_2)} \mathcal{F}(d_{\mathcal{I}}(\cdot|\mathbf{i}))$*. It holds*

$$|\mathcal{J}_{1,\mathcal{I}}(\pi_1) - \mathcal{J}_{1,\mathcal{I}}(\pi_2)| \leqslant T\mathcal{F}(d_{\mathcal{I}}(\cdot|\mathbf{i}))d^{TV}(\pi_1, \pi_2).$$

A global (but looser) upper bound of the Lipschitz constant can be derived as $TH_{\max}$, where $H_{\max}$ is the maximum entropy that can be obtained over the support. These results provide an interesting insight into how (a bound on) the smoothness constant behaves, as both the objectives defined over true states or observations have Lipschitz constants that are not directly dependent on the policies themselves.

[7] Note that the meaning and role of the regularization parameters and corresponding regularization term, color-highlighted in the algorithm, will be clarified in the next section.

[8] The full derivation of the result is in Appendix B.1.

4.2.3 Scalable Solutions without accessing a Simulator

The Assumption 4.2.1 of having access to the cPOMDP specification is rather restrictive and arguably unreasonable in domains where a (white-box) simulator is not available. To overcome this assumption, we aim to refine the design of our algorithmic solution to work with quantities related to observations only. Luckily, beliefs can still be computed approximately well without access to the cPOMDP model. Belief approximation techniques have been extensively studied in the literature (e.g., see Subramanian et al. [2022] for a summary). Here, we do not delve into the technicalities of the latter, which are out of the scope of this work, and we instead assume to have access to an *approximated* oracle to compute beliefs.

Assumption 4.2.2 (Belief Oracle). *Let* $a \in \mathcal{A}$ *and* $o \in \mathcal{O}$. *Given an approximate belief* $\hat{b}_t \in \Delta_{\mathcal{S}}$ *of the true belief* b_t, *an* oracle belief approximator *gives* $\hat{b}_{t+1}$ *such that* $\|T^{ao}(\hat{b}_t) - \hat{b}_{t+1}\|_1 \leqslant \epsilon$.

With the latter, we can follow it as is, computing approximate beliefs instead of the true beliefs. Yet, we have to change the feedback as we cannot compute the entropy on the true states. Luckily, the trivial MOE feedback (4.3) is *not* the only option we have. We can use the approximate beliefs to reconstruct *believed trajectories* over states and then compute the feedbacks as their entropy. We call the latter the *Maximum Believed Entropy* (MBE):

$$\max_{\pi \in \Pi_{\mathcal{I}}} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{b} \sim p^{\pi}_{\mathcal{B}}} \mathop{\mathbb{E}}_{\tilde{\mathbf{s}} \sim p(\cdot|\boldsymbol{b})} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \right\} := \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi), \tag{4.33}$$

where the update of the belief in p^{π} is now given by the approximate belief oracle. Notably, we can extend both Theorem 4.2.1, 4.2.2 to the MBE objective.

Theorem 4.2.3. *For a policy* $\pi_\theta \in \Pi_{\mathcal{I}}$ *parametrized by* $\theta \in \Theta \subseteq \mathbb{R}^{SA}$, *we have*

$$\nabla_\theta \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_\theta) = \mathop{\mathbb{E}}_{\boldsymbol{b} \sim p^{\pi}_{\mathcal{B}}} \mathop{\mathbb{E}}_{\tilde{\mathbf{s}} \sim p(\cdot|\boldsymbol{b})} \Big[\nabla_\theta \log \pi_\theta(\tilde{\mathbf{s}}) \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \Big], \tag{4.34}$$

where $\nabla_\theta \log \pi_\theta(\tilde{\mathbf{s}})$ *are defined as in 4.2.1. Additionally, let* $\mathcal{T}_{\mathcal{B}}(\pi_1, \pi_2) = \{\boldsymbol{b} \in \mathcal{T}_{\mathcal{B}} : p^{\pi_1}(\boldsymbol{b}) > 0 \vee p^{\pi_2}(\boldsymbol{b}) > 0\}$, $\boldsymbol{b}^\star = \arg\max_{\boldsymbol{b} \in \mathcal{T}_{\mathcal{B}}(\pi_1,\pi_2)} \mathbb{E}_{\tilde{\mathbf{s}} \sim p(\cdot|\boldsymbol{b})} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$, *and* $\bar{\mathcal{F}}(\boldsymbol{b}^\star) = \mathbb{E}_{\tilde{\mathbf{s}} \sim p(\cdot|\boldsymbol{b})} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$, *we have*

$$|\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_1) - \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_2)| \leqslant T \bar{\mathcal{F}}(\boldsymbol{b}^\star) d^{TV}(\pi_1, \pi_2). \tag{4.35}$$

Interestingly, compared to the other results in Theorem 4.2.2, MBE displays an upper bound of the Lipschitz constant that depends on the policies π_1, π_2 directly (through $\boldsymbol{b}^\star$). Additionally, $\bar{\mathcal{F}}(\boldsymbol{b}^\star)$ consists in the best *expected* believed entropy, which is generally smaller than $\mathcal{F}(d(\cdot|\mathbf{i}^\star)), i \in \{\mathcal{S}, \mathcal{O}\}$ of Theorem 4.2.2.

Objectives Gaps and Hallucinatory Effect

Without 4.2.1, we cannot know the value of the MSE objective anymore. Thus, it is hard to keep track of the mismatch between what the agent expects the (latent) performance to be and what it truly is once it is evaluated on the true states of the environment. However, it is possible to show that the true objective lies in a space explicitly encircled by the proxies. First, we provide the following instrumental definitions:

Definition 4.2.1. *We define* $\mathcal{T}_{\mathcal{O}}(\mathbf{s}) = \{\mathbf{o} \in \mathcal{T}_{\mathcal{O}} : \mathcal{F}(d(\cdot|\mathbf{o})) \geqslant \mathcal{F}(d(\cdot|\mathbf{s}))\}, \mathcal{T}(\mathbf{s}) = \{\tilde{\mathbf{s}} \in \mathcal{T}_{\tilde{\mathcal{S}}} : \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \geqslant \mathcal{F}(d(\cdot|\mathbf{s}))\}$ *as the set of trajectories over observations and believed states, respectively, for which their entropy is higher than the entropy of a fixed trajectory over true states. We let* $\mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s}) = \sum_{\mathbf{o} \in \mathcal{T}_{\mathcal{O}}(\mathbf{s})} p^{\pi}(\mathbf{o}|\mathbf{s}), \mathbb{P}(\mathcal{T}|\boldsymbol{b}) = \sum_{\tilde{\mathbf{s}} \in \mathcal{T}(\mathbf{s})} \boldsymbol{b}(\mathbf{s})$ *the cumulative probability of drawing a trajectory form the above sets and* $\bar{p}_{\mathcal{S}}(\mathbf{s}) = \mathbb{E}_{\boldsymbol{b} \sim p^{\pi}(\cdot|\mathbf{s})} \mathbb{P}(\mathcal{T}|\boldsymbol{b})$ *the expected probability of the believed set. Finally,* $\mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) = \mathbb{E}_{\mathbf{o} \sim p^{\pi}(\cdot|\mathbf{s})}[\mathcal{F}(d(\cdot|\mathbf{o}))], \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s}) = \mathbb{E}_{\boldsymbol{b} \sim p^{\pi}(\cdot|\mathbf{s})} \mathbb{E}_{\tilde{\mathbf{s}} \sim \boldsymbol{b}}[\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))]$ *the MOE (MBE) objective for a fixed trajectory on the states.*

Then, the following theorem holds:

Theorem 4.2.4 (Proxy Gaps)**.** *For a fixed policy* $\pi \in \Pi_{\mathcal{I}}$*, the MSE objective* $\mathcal{J}_{1,\mathcal{S}}(\pi)$ *is bounded by the MOE objective according to*

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \leqslant \mathop{\mathbb{E}}_{\mathbf{s} \sim \bar{p}_{\mathcal{S}}} \Big[\frac{1}{\mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})} \mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) \Big]$$

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \geqslant \mathop{\mathbb{E}}_{\mathbf{s} \sim \bar{p}_{\mathcal{S}}} \Big[\frac{1}{1 - \mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})} \mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) \Big] - \mathop{\mathbb{E}}_{\mathbf{s} \sim \bar{p}_{\mathcal{S}}} \Big[\frac{\mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})}{1 - \mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})} \Big] \log O$$

Analogously, $\mathcal{J}_{1,\mathcal{S}}(\pi)$ *is bounded by the MBE objective according to*

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \leqslant \mathop{\mathbb{E}}_{\mathbf{s} \sim \bar{p}} \Big[\frac{1}{\bar{p}_{\mathcal{S}}(\mathbf{s})} \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s}) \Big]$$

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \geqslant \mathop{\mathbb{E}}_{\mathbf{s} \sim \bar{p}} \Big[\frac{1}{1 - \bar{p}_{\mathcal{S}}(\mathbf{s})} \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s}) \Big] - \mathop{\mathbb{E}}_{\mathbf{s} \sim \bar{p}} \Big[\frac{\bar{p}_{\mathcal{S}}(\mathbf{s})}{1 - \bar{p}_{\mathcal{S}}(\mathbf{s})} \Big] \log S$$

These results show thaxt the true objective (MSE) is upper/lower bounded by the proxies depending on the probability to generate trajectories (over observations or believed states, respectively) with entropy higher than the one of the trajectory that generated them. We refer to this probability as *hallucination probability* and to the resulting phenomenon as **hallucinatory effect**. We show in Figure 4.8 a visual representation of the MBE gaps in 4.2.4. It is evident that for low over-estimation probabilities ($\bar{p}_{\mathcal{S}} = 0.02$), the MBE objective is a good lower bound for the MSE objective. Indeed, one may notice that the MOE gap is potentially looser: In many scenarios $\log(O) \gg \log(S)$ while on the other hand $\bar{p}_{\mathcal{S}}(\mathbf{s})$ is the result of an additional expectation with respect to $\mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})$. On the other hand, it is less so as the hallucination probability increases. The full derivation of these results can be found in B.1.

The role of hallucinatory effects is crucial. Indeed, when the effect of hallucinations is negligible, the proxy objectives are reasonable lower bounds to the true MSE objective, and optimizing them guarantees at least a non-degradation of the MSE objective. The hallucinatory effect, i.e., generating over-entropic trajectories due to the randomness of the generating process, on either observations or beliefs, can be controlled by reducing the randomness of the generating process itself. Unfortunately, under Assumption 4.2.2, we cannot control the observation model as done in the previous section. However, we have partial control over the trajectory of beliefs that are generated, as they are (approximately) learned and the belief update is conditioned on the taken action. Thus, we can follow the same rationale and derive a regularized objective built upon $\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi)$. In particular, we can maintain a valid lower bound to

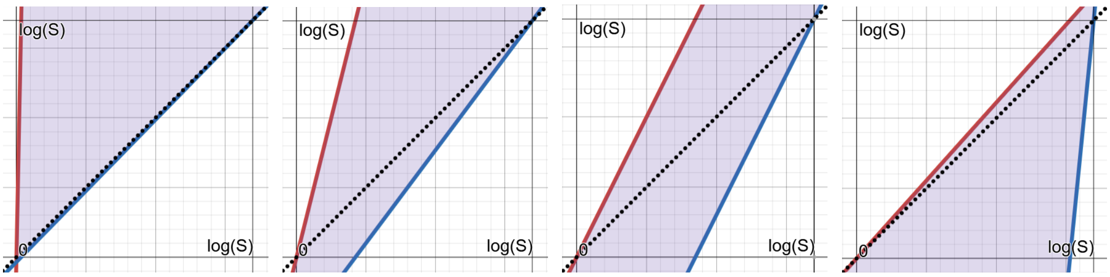


Figure 4.8: *MBE Proxy gaps: for different hallucination probabilities $\bar{p}_{\mathcal{S}}$ and a fixed trajectory* $\mathbf{s}$*, the y-axis represents the* ***possible MSE values*** *contained between the* ***upper bound*** *and* ***lower bound*** *as* $\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi)(\pi|\mathbf{s})$ *varies between* 0 *and the maximum value* $\log(S)$ *(the corresponding* ***MBE values*** *are plotted over the diagonal to allow a comparisons).*

the MBE objective while enforcing the generation of a sequence of low-entropy belief states $\boldsymbol{b} = (b_1, \cdots, b_T)$ with the following:

$$\begin{aligned}\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi) &\geqslant \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi) - \rho \mathop{\mathbb{E}}_{\boldsymbol{b}\sim p^{\pi}}[\mathcal{H}(\boldsymbol{b})] \\ &\geqslant \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi) - \rho \mathop{\mathbb{E}}_{\boldsymbol{b}\sim p^{\pi}}\left[\sum\nolimits_{t\in[T]} \mathcal{H}(b_t)\right] =: \tilde{\mathcal{J}}^{\rho}_{1,\mathcal{S}}(\pi)\end{aligned}$$

where the second inequality is due to the sub-additivity of the entropy. We call the obtained $\tilde{\mathcal{J}}^{\rho}_{1,\mathcal{S}}(\pi)$ *MBE with belief regularization* (Reg-MBE for short). Then, the policy gradient for parametrized policies $\nabla_\theta \tilde{\mathcal{J}}^{\rho}_{1,\mathcal{S}}(\pi_\theta)$ for this objective is

$$\nabla_\theta \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_\theta) - \rho \mathop{\mathbb{E}}_{\boldsymbol{b},\tilde{\mathbf{s}}\mathbf{a}\sim p^{\pi_\theta}}\left[\nabla_\theta \log \pi_\theta(\tilde{\mathbf{s}}\mathbf{a}) \sum\nolimits_{t\in[T]} \mathcal{H}(b_t)\right].$$

It is easy to see that whenever $\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi)$ is a good proxy (i.e., a tight lower bound) of the true MSE objective, then the regularized objective $\tilde{\mathcal{J}}^{\rho}_{1,\mathcal{S}}(\pi)$ will be a reasonable lower bound as well. Most importantly, the regularization term incentives lower-entropy beliefs, which keeps $\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi)$ in a region where it approximates MSE well. From these considerations, a *belief-regularized* version of the Algorithm 4.2.2 is proposed by simply modifying how the gradient step in $(\mathbf{6})$ is computed, as can be seen in the regularized version of Algorithm 4.2.2.

4.2.4 Numerical Validation

In this section, we provide an empirical corroboration of the proposed methods and reported claims. The section is organized as follows: first, we describe the experimental set-up; then, we compare the performance driven by the proxy objectives (MOE, MBE, MBE with belief regularization) against the ideal objective (MSE); finally, we study the impact of belief approximation on MBE-based algorithms (with and without regularization).

Experimental Set-Up

We consider the following set of finite domains:

(i) A 5×5-Gridworld with a single room, where $\mathcal{O} = \mathcal{S}$ and the emission matrix $\mathbb{O}$ is such that every row is a (discretized) Gaussian $\mathbb{O}(o|s) = \mathcal{N}(s, \sigma^2)$;

(ii) A 6×6-Gridworld with 4 identical rooms, where $\mathcal{O} = \mathcal{S}$ and the emission matrix $\mathbb{O}$ is such that every row is a (discretized) Gaussian $\mathbb{O}(o|s) = \mathcal{N}(s, \sigma^2)$;

(iii) A 6×6-Gridworld with 4 identical rooms, where $\mathcal{O} = \{1, 2, 3, 4\}$ and the deterministic emission matrix $\mathbb{O}$ such that for every state $\mathbb{O}(s)$ is the id of the room in which the state lies;

(iv) A 6×6-Gridworld with 4 identical rooms, where $\mathcal{O} = \{1, 2\}$ and the deterministic emission matrix $\mathbb{O}$ such that for every state $\mathbb{O}(s)$ is the side of the grid (left rooms or right rooms) the state lies in.

In all the environments described above, the agent has four actions to take, one for moving to the adjacent grid cell in each of the coordinate directions. Moving against a wall undoes the effect of an action. When we say an environment is *deterministic* we mean that the agent actions never fail. In a *stochastic* environment each action has 0.1 failure probability, which has the equivalent effect of taking one of the other three actions at random. Finally, we compare the algorithms designed for the MSE, MOE, MBE objectives presented in previous sections.[9] Irrespective of the optimized objective, their performance is evaluated on the **true state entropy** (Equation 5.2), which is the ultimate target of state entropy maximization in cPOMDPs. All of the algorithms optimize a policy within the BA class $\overline{\Pi}_{\mathcal{B}}$. A visualization of the described environment is provided in C.2, while the choice of the experimental parameters is discussed in C.2. C.2 provides a finer analysis of the choice of the policy class.

MSE in cPOMDP with the Proxy Objectives

In this section, we compare the performance obtained by Algorithm 4.2.2 specialized for the different proxy objectives. For the sake of clarity, here we assume the belief updates to be computed exactly, while we study the impact of the belief approximation in the next section. 4.9 shows that all of the objectives works equally well in easy settings, e.g., deterministic transitions and small observation noise. However, major differences arise when considering harder settings. The MOE objective is sensitive to the quality of the observations, which is evident from the performance degradation in Figures 4.10, 4.11, 4.12. Instead, MBE objectives are remarkably robust to their (diminishing) quality. MBE with belief regularization (Reg-MBE) always performed better than the non-regularized version, showing faster convergence or better final performance. In Figure 4.13, we see that stochastic transitions are also arduous for MOE and MBE. MBE with belief regularization proved to be better. Interestingly, the true state entropy improvement happens concurrently with the optimization of the regularization term (4.14). Unsurprisingly, optimizing the MSE objective leads to the best performance in most cases, as a testament that whenever the cPOMDP specification is available in simulation, it is worth training the policy we seek to deploy on the true state entropy. Interestingly, in some limit cases with extreme disentanglement between the observations and the true MSE objective (Figure 4.15), the belief-regularized MBE proxy performed slightly better. Finally, Figure 4.14 shows how the MBE is severely hallucinated, an effect that is mitigated with belief regularization.

[9] While we only compare algorithms of our design, we note that we could not find any previous algorithm addressing state entropy maximization in cPOMDPs.

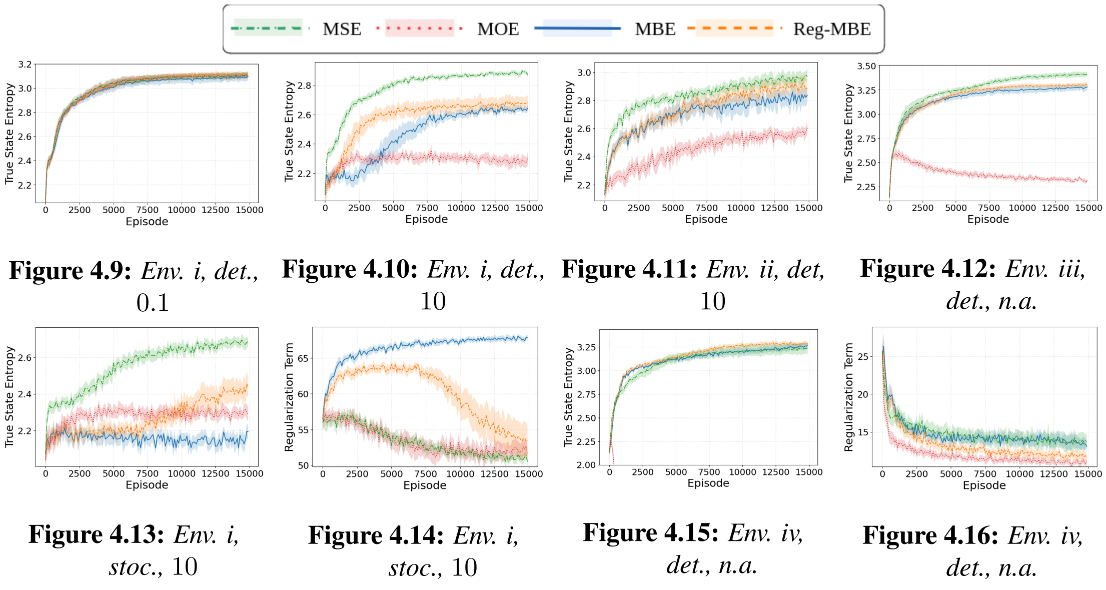


Figure 4.9: *Env. i, det.,* 0.1

Figure 4.10: *Env. i, det.,* 10

Figure 4.11: *Env. ii, det,* 10

Figure 4.12: *Env. iii, det., n.a.*

Figure 4.13: *Env. i, stoc.,* 10

Figure 4.14: *Env. i, stoc.,* 10

Figure 4.15: *Env. iv, det., n.a.*

Figure 4.16: *Env. iv, det., n.a.*

True state entropy (or regularization term) obtained by Algorithm 4.2.2 specialized for the feedbacks MSE, MOE, MBE, MBE with belief regularization *(Reg-MBE). For each plot, we report a tuple (environment, transition noise, observation variance) where the latter is* not available *(n.a.) when observations are deterministic. For each curve, we report the average and 95% c.i. over 16 runs.*

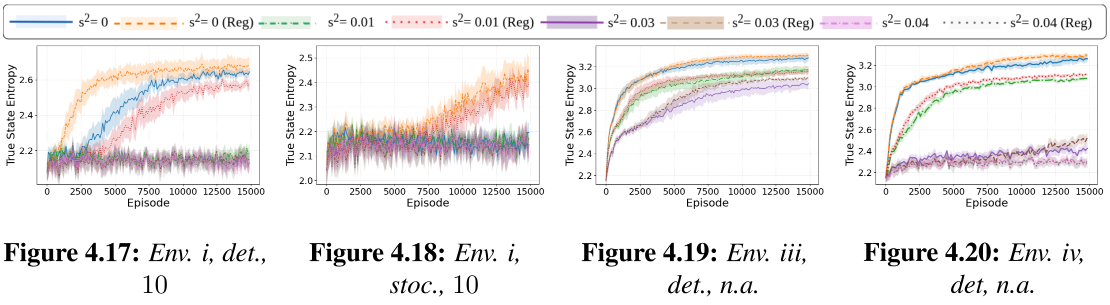


Figure 4.17: *Env. i, det.,* 10

Figure 4.18: *Env. i, stoc.,* 10

Figure 4.19: *Env. iii, det., n.a.*

Figure 4.20: *Env. iv, det, n.a.*

True state entropy obtained by Algorithm 4.2.2 with MBE, MBE with belief regularization *(MBE with Reg) feedbacks under different levels of approximation noise s^2. For each plot, we report a tuple (environment, transition noise, observation variance) where the latter is* not available *(n.a.) when observations are deterministic. For each curve, we report the average and 95% c.i. over 16 runs.*

The Impact of Belief Approximation

In the previous section, we compared the algorithms in an ideal setting in which the belief is approximated exactly. Here we instead consider the effect of the belief approximation on the same experiments. Especially, to keep full generality of our results, we perturb the exact beliefs with an entry-wise Gaussian noise (with variance $s^2 = \{0, 0.01, 0.03, 0.04\}$ respectively), so that our results do not apply to a single belief approximator but any approximator with a bounded error.[10] All Figures from 4.17 to 4.20 provide two important evidences. First, when good belief approximators are available, the resulting performance is strikingly similar to the ideal setting with exact beliefs. Secondly, MBE with belief regularization is significantly more robust to perturbations, hinting that mitigating hallucination also alleviates the impact of the approximation error to some extent.

[10] For the sake of clarity, here we report the variance of the perturbation instead of the approximation.

Concluding Remarks

In this section, we expanded over the topic of state entropy maximization problem in cPOMDPs towards more scalable solutions. In particular, we choose a convenient subclass of non-Markovian policies that retain compressed information of history without incurring unreasonable memory requirements, namely belief-conditioned policies. Additionally, we showed how such scalable solutions might risk to incur into hallucinations, while trying to optimize thought a learnt belief-representation. Finally, we designed practical first-order algorithms, which are based on policy gradient, to overcome the inherent non-convexity of the considered objective functions.

CHAPTER 5

Unsupervised Pre-Training with Multiple Agents

Multi-Agent Reinforcement Learning [MARL, Albrecht et al., 2024] has recently demonstrated promising results in learning complex behaviors in the presence of multiple agents, spanning from coordination [Samvelyan et al., 2019], strategic planning under imperfect information [Perolat et al., 2022], up to even emergent economic behavior like trading [Johanson et al., 2022]. However, as with single-agent RL, much of the research in MARL still focuses on tabula rasa learning-starting from scratch without leveraging any prior knowledge, offline data, or policy pre-training.

While this approach is general, it poses significant limitations when applied to real-world scenarios, where training from scratch is often slow, expensive, and largely unnecessary [Agarwal et al., 2022]. Some progress has been made in the multi-agent domain, particularly in areas like ad hoc teamwork [Mirsky et al., 2022] and zero-shot coordination [Hu et al., 2020], which aim to build more adaptable agents. Yet, the role of unsupervised pre-training in MARL remains largely unexplored. The only notable exception is Jiang et al. [2022], which proposes initializing policies with strong inter-agent interaction, though without offering a formal theoretical framework for doing so.

This chapter investigates how unsupervised pre-training techniques can be extended to handle multiple agents, through the lens of state entropy maximization. First, we introduce a new framework for multi-agent state entropy maximization, which generalizes the concept of state entropy maximization in single-agent settings to the multi-agent contexts. Them we address the framework by fist investigating alternative formulations, theoretically characterizing what are the positives and negatives and highlighting how the problem, even if seeming rather intuitive in theory, is actually challenging in practice. Then, we present a scalable, decentralized, trust-region policy search algorithm to address the problem in practical settings. Finally, we provide numerical validations to both corroborate the theoretical findings and pave the way for unsupervised MARL via state entropy maximization in challenging multi-agent settings, showing that optimizing for a specific objective, namely *mixture entropy*, provides an excellent trade-off between tractability and performances.

This chapter is based on the paper *"Towards Principled Unsupervised Multi-Agent Reinforcement Learning"*, co-authored with M. Mutti and M. Restelli and published at NeurIPS 2025. In order to strengthen some intuitions, we included some results (namely, Th. 5.1.1 and Fact 5.2.2) about parallel non-interacting agents from *"Enhancing Diversity in Parallel Agents: A Maximum State Entropy Exploration Story"*, co-authored with V. De Paola, M. Mutti, and M. Restelli, published at ICML 2025.[1]

Convex Formulation of Markov Games

As a first step, we introduce a generalization of MGs that allows the optimization of more expressive objectives over the state distribution, analogous to what was introduced for MDPs in Section 2. This generalized framework forms the foundation for our study of unsupervised pre-training in multi-agent systems.

Convex Markov Games (cMGs).
A cMG is defined as a tuple $\mathcal{M}^{\mathcal{F}} := (\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{F}, \mu, T)$, where $(\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathbb{P}, \mu, T)$ represents a Markov Game without rewards, and $\mathcal{F}$ is an F-bounded *concave* utility function with $F < \infty$. In other words, a cMG is a Markov Game equipped with a (potentially non-linear) utility function $\mathcal{F}(\cdot)$.

Gemp et al. [2025] recently introduced such a convex generalization of MGs that consists in a MG equipped with (non-linear) functions of the *stationary joint state* distribution $\mathcal{F}(d^\pi)$. We expand over this definition, by noticing that state entropy maximization can be casted as solving a cMG equipped with an entropy functional, namely $\mathcal{F}(\cdot) := \mathcal{H}(\cdot)$.

Additionally, when per-agent transitions are independent from the presence of other agents, namely the transition function $\mathbb{P}$ is such that for any agent $i \in \mathcal{N}$, $\mathbb{P}(s' \mid s, a) = \mathbb{P}(s'_i \mid s_i, a_i)\mathbb{P}(s'_{-i} \mid s_{-i}, a_{-i})$, we refer to the cMG as a *convex Parallel MDP* (cPMDP), a convex generalization of PMDPs [Sucar, 2007].

In the sections that follow, we discuss how to properly define the domain (support) of the concave utility function and explore how this design choice impacts the effectiveness and structure of the resulting pre-training phase.

5.1 Problem Formulation

This section addresses the first of the research questions:

> *Can we formulate in a principled way unsupervised pre-training via state entropy maximization in MARL as well?*

In fact, when a reward function is not available, the core of the problem resides in finding a well-behaved problem formulation coherent with the task. How much should

[1] A complete reference can be found in the bibliography [Zamboni et al., 2025b, De Paola et al., 2025]

the agents coordinate? How much information should they have access to? Different answers depict different objectives.

Joint Objectives. The first and most straightforward way to formulate the problem is to define it as in the MDP setting, with the joint state distribution simply taking the place of the single-agent state distribution. In this case, we define a *Joint* objective, consisting of

$$\max_{\pi=(\pi^i\in\Pi^i)_{i\in[|\mathcal{N}|]}} \Big\{\zeta_\infty(\pi) := \mathcal{F}(d^\pi)\Big\} \tag{5.1}$$

$$\max_{\pi=(\pi^i\in\Pi^i)_{i\in[|\mathcal{N}|]}} \Big\{\zeta_K(\pi) := \mathop{\mathbb{E}}_{d_K\sim p_K^\pi} \mathcal{F}(d_K)\Big\} \tag{5.2}$$

In state entropy maximization tasks, i.e. by setting $\mathcal{F}(\cdot) := \mathcal{H}(\cdot)$, an optimal (joint) policy will try to cover the joint state space as uniformly as possible, either in expectation or over a finite number of trials respectively. In this, the joint formulation is rather intuitive as it describes the most general case of multi-agent exploration. Moreover, as each agent sees a difference in performance explicitly linked to others, this objective should be able to foster coordinated exploration. As we will see, this comes at a price.

Disjoint Objectives. One might look for formulations more coherent with a multi-agent setting. The most trivial option is to design a disjoint counterpart of the objectives, that means to define a set of functions supported on per-agent state distributions rather than joint distributions. This intuition leads to *Disjoint* objectives:

$$\Big\{\max_{\pi^i\in\Pi^i} \zeta^i_\infty(\pi^i,\pi^{-i}) := \mathcal{F}(d_i^{\pi^i,\pi^{-i}})\Big\}_{i\in[|\mathcal{N}|]} \tag{5.3}$$

$$\Big\{\max_{\pi^i\in\Pi^i} \zeta^i_K(\pi^i,\pi^{-i}) := \mathop{\mathbb{E}}_{d_K\sim p_K^{\pi^i,\pi^{-i}}} \mathcal{F}(d_{K,i})\Big\}_{i\in[|\mathcal{N}|]} \tag{5.4}$$

According to these objectives, each agent will try to maximize her own marginal state entropy separately, neglecting the effect of her actions over others performances. In other words, we expect this objective to hinder the potential coordinated exploration, where one has to take as step down as so allow a better performance overall.

Mixture Objectives. At last, we introduce a problem formulation that will be later prove capable of uniquely taking advantage of the structure of the problem. In order to do so, we first introduce the following:

Assumption 5.1.1 (Uniformity)**.** *The agents have the same state spaces, namely* $\mathcal{S}_i = \mathcal{S}_j = \tilde{\mathcal{S}},\ \forall(i,j)\in\mathcal{N}\times\mathcal{N}$. [2]

Under this assumption, from now on we will drop the agent subscript when referring to the per-agent states, and use $\tilde{\mathcal{S}}$ instead. Interestingly, this assumptions allows us to define a particular distribution, namely:

$$\tilde{d}^\pi(\tilde{s}) := \frac{1}{|\mathcal{N}|}\sum_{i\in[|\mathcal{N}|]} d_i^\pi(\tilde{s}) \in \Delta_{\tilde{\mathcal{S}}}. \tag{5.5}$$

[2] One should notice that even in cMGs where this is not (even partially) the case, the assumption can be enforced by padding together the per-agent states.

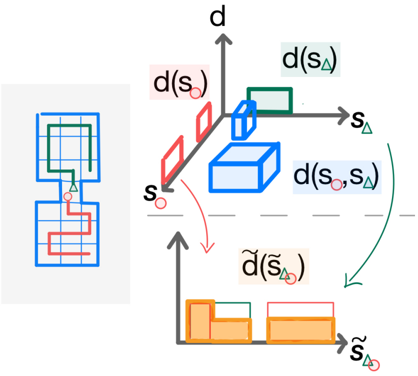


Figure 5.1: *The interaction on the* left *induces different (empirical) distributions: marginal distributions for* ***agent 1*** *and* ***agent 2*** *over their respective states; a* ***joint distribution*** *over the product space; a* ***mixture distribution*** *over a common space, defined as the average. The mixture distribution is usually* less sparse.

We refer to this distribution as *mixture* distribution, given that it is defined as a uniform mixture of the per-agent marginal distributions. Intuitively, it describes the average probability over all the agents to be in a common state $\tilde{s} \in \tilde{\mathcal{S}}$, in contrast with the joint distribution that describes the probability for them to be in a joint state s, or the marginals that describes the probability of each one of them separately. In Figure 5.1 we provide a visual representation of these concepts.

Similarly to what happens for the joint distribution, one can define the empirical distribution induced by K episodes as $\tilde{d}_K(\tilde{s}) = \frac{1}{|\mathcal{N}|}\sum_{i\in[|\mathcal{N}|]} d_{K,i}(\tilde{s})$ and $\tilde{d}^\pi = \mathbb{E}_{\tilde{d}_K \sim p_K^\pi}[\tilde{d}_K]$. The mixture distribution allows for the definition of the *Mixture* objectives, in their infinite and finite trials formulations respectively:

$$\max_{\pi=(\pi^i\in\Pi^i)_{i\in[|\mathcal{N}|]}} \left\{ \tilde{\zeta}_\infty(\pi) := \mathcal{F}(\tilde{d}^\pi) \right\} \tag{5.6}$$

$$\max_{\pi=(\pi^i\in\Pi^i)_{i\in[|\mathcal{N}|]}} \left\{ \tilde{\zeta}_K(\pi) := \mathop{\mathbb{E}}_{\tilde{d}_K \sim p_K^\pi} \mathcal{F}(\tilde{d}_K) \right\} \tag{5.7}$$

When this kind of objectives is employed in state entropy maximization, the entropy of the mixture distribution decomposes as

$$H(\tilde{d}^\pi) = \frac{1}{|\mathcal{N}|}\sum_{i\in[|\mathcal{N}|]} H(d_i^\pi) + \frac{1}{|\mathcal{N}|}\sum_{i\in[|\mathcal{N}|]} D_{\mathrm{KL}}(d_i^\pi || \tilde{d}^\pi)$$

and one remarkable scenario arises: Agents follow policies possibly inducing lower disjoint entropies, but their induced marginal distributions are maximally different. Thus, the average entropy remains low, but the overall mixture entropy is high due to diversity (i.e., high values of the KL divergences). This scenario has been referred to in Kolchinsky and Tracey [2017] as the *clustering* scenario and, in the following, we will provide additional evidences why this scenario is particularly relevant.

Further Intuitions on the Advantages of Mixture Distributions

The advantages of employing mixture distribution in scenarios involving multiple interacting agents will be the main object of the remaining of the chapter. Yet, it is possible to derive an intuitive yet grounded justifications by looking at the concentration properties of entropy functionals when agents act over the same environment without interacting with each other, a scenario encompassed by cPMDPs:

Theorem 5.1.1. *Let d^π be the (categorical) distribution induced by π over the finite set $\mathcal{S}$ with $|\mathcal{S}| = S$, and let d_K be the empirical distribution obtained from K independent samples drawn from d^π. Then, for any $\epsilon > 0$, the following bound holds:*

$$\mathbb{P}\left(\mathcal{H}(d_\pi) - \mathcal{H}(d_K) > \epsilon\right) \leqslant 2S \exp\left(-K \cdot \frac{\epsilon^2 \mathrm{Var}(d_\pi)}{2S^3\mathcal{H}^2(d_\pi)}\right),$$

where $\mathcal{H}(d_K)$ and $\mathcal{H}(d_\pi)$ denote the entropy of the empirical and true distributions, respectively, and $\mathrm{Var}(d_\pi) = \sum_{s\in[\mathcal{S}]} d_\pi(s)(1 - d_\pi(s))$ is the variance of a random variable associated with the categorical distribution d_π. Furthermore, to ensure this concentration with confidence $1 - \delta$, the number of samples n must satisfy the following lower bound:

$$K \geqslant \frac{2S^3\mathcal{H}^2(d_\pi)}{\epsilon^2\mathrm{Var}(d_\pi)} \cdot \ln\frac{2S}{\delta}.$$

This theorem establishes an upper bound on the probability that the entropy difference between the true and empirical distributions exceeds ϵ. Specifically, the probability of large deviations between these two entropies decreases exponentially with K, with the rate of convergence influenced by the entropy of the true distribution d^π. Notably, as $\lim_{\mathcal{H}(d_\pi)\to 0} \frac{\mathcal{H}^2(d_\pi)}{\mathrm{Var}(d_\pi)} = 0$, distributions with lower entropy require fewer samples for concentration, implying they are easier to approximate empirically. This result suggests a key advantage of state entropy maximization: when multiple agents explore the environment simultaneously, each can focus on different regions of the state space. As a result, they induce distributions with lower entropy compared to a single policy covering the entire space.

5.2 A Formal Characterization of Multi-Agent State Entropy Maximization

In the previous section, we provided a principled problem formulation of multi-agent state entropy maximization through an array of different objectives. In this section, we address the second research question:

How are different formulations related? Do crucial theoretical differences emerge?

First of all, we show that if we look at state entropy maximization tasks, i.e. the ones defined by setting the functional $\mathcal{F}(\cdot) := \mathcal{H}(\cdot)$, all the objectives in infinite-trials formulation can be elegantly linked one to the other though the following result:

Lemma 5.2.1 (Entropy Mismatch)**.** *For every cMG $\mathcal{M}^{\mathcal{H}}$ equipped with an entropy functional, for a fixed (joint) policy $\pi = (\pi^i)_{i \in \mathcal{N}}$ the infinite-trials objectives are ordered according to:*

$$\frac{\mathcal{H}(d^{\pi})}{|\mathcal{N}|} \leqslant \frac{1}{|\mathcal{N}|} \sum_{i \in [|\mathcal{N}|]} \mathcal{H}(d_i^{\pi}) \leqslant \mathcal{H}(\tilde{d}^{\pi})$$

$$\mathcal{H}(\tilde{d}^{\pi}) \leqslant \sup_{i \in [|\mathcal{N}|]} \mathcal{H}(d_i^{\pi}) + \log(|\mathcal{N}|) \leqslant \mathcal{H}(d^{\pi}) + \log(|\mathcal{N}|)$$

The full derivation of these bounds is reported in Appendix B.2. This set of bounds prescribe that the difference in performances over infinite-trials objective for the same policy can be generally bounded as a function of the number of agents. In particular, disjoint objectives generally provides poor approximations of the joint objective from the point of view of the single-agent, while the mixture objective is guaranteed to be a rather good lower bound to the joint entropy as well, since its over-estimation scales logarithmically with the number of agents.

It is still an open question how hard it is to actually optimize for these objectives. Now, while cMGs are a novel interaction framework, whose general properties are far from being well-understood, they surely enjoy some nice properties. In particular, as commonly done in Potential Markov Games [Leonardos et al., 2022], it is possible to exploit the fact that performing Policy Gradient [PG, Sutton et al., 1999, Peters and Schaal, 2008b] independently among the agents is equivalent to running PG jointly, when this is done over the same common objective (see Appendix B.2). This allows us to provide a rather positive answer, here stated informally and extensively discussed in Appendix B.2:

Fact 5.2.1 ((Informal) Efficiency of Independent Policy Gradient)**.** *Under proper assumptions, for every cMG $\mathcal{M}^{\mathcal{F}}$, independent Policy Gradient over infinite trials non-disjoint objectives via centralized-information policies of the form $\pi = (\pi^i \in \Delta_{\mathcal{S}}^{\mathcal{A}^i})_{i \in [|\mathcal{N}|]}$ converges* fast.

This result suggests that PG should be generally enough for the infinite-trials optimization, and thus, from a certain point of view, these problems might not be of so much interest.

Interestingly, a similar result can be indeed derived for cPMDPs as well, in which a parallel formulation of the MaxEnt Algorithm [Hazan et al., 2019] called Parallel MaxEnt can be shown not only to be efficient, but also to enjoy an acceleration in convergence due to the presence of multiple and parallel agents. Here, we state the result informally, while an extensive discussion can be found in Appendix B.2:

Fact 5.2.2 ((Informal) Efficiency of Parallel MaxEnt)**.** *Under proper assumptions, for every cPMDP $\mathcal{M}^{\mathcal{F}}$, Parallel MaxEnt over infinite trials objectives achieves near optimality by using a number of samples from the environment that scales inversely with the number of agents.*

However, convex MDP theory has outlined that optimizing for infinite-trials objectives might actually lead to extremely poor performances as soon as the policies are deployed over just a handful of trials, i.e. in almost any practical scenario [Mutti et al., 2023]. We show that this property transfers almost seamlessly to cMGs as well, with interesting additional takeaways:

Theorem 5.2.2 (Objectives Mismatch in cMGs)**.** *For every cMG $\mathcal{M}^{\mathcal{F}}$ equipped with a L-Lipschitz function $\mathcal{F}$ (see Ass. 2.3.1), let $K \in \mathbb{N}^+$ be a number of evaluation episodes/trials, and let $\delta \in (0, 1]$ be a confidence level, then for any (joint) policy $\pi = (\pi^i \in \Pi^i)_{i\in[|\mathcal{N}|]}$, it holds that*

$$|\zeta_K(\pi) - \zeta_\infty(\pi)| \leqslant LT\sqrt{\frac{2|\mathcal{S}|\log(2T/\delta)}{K}},$$

$$|\zeta^i_K(\pi) - \zeta^i_\infty(\pi)| \leqslant LT\sqrt{\frac{2|\tilde{\mathcal{S}}|\log(2T/\delta)}{K}},$$

$$|\tilde{\zeta}_K(\pi) - \tilde{\zeta}_\infty(\pi)| \leqslant LT\sqrt{\frac{2|\tilde{\mathcal{S}}|\log(2T/\delta)}{|\mathcal{N}|K}}.$$

The full derivation of these bounds is reported in Appendix B.2. In general, this set of bounds confirms that infinite and finite trials objectives might be extremely different, and thus optimizing the infinite-trials objective might lead to unpredictable performance at deployment, whenever this is done over a handful of trials. This property is inherently linked to the *convex* nature of convex MDPs, and Mutti et al. [2023] introduces it to highlight that the concentration properties of empirical state-distributions [Weissman et al., 2003] allow for a nice dependency on the number of trials in controlling the mismatch. In multi-agent settings, the result portraits a more nuanced scene:

(i) The mismatch still scales with the cardinality of the support of the state distribution, yet, for joint objectives, this quantity scales very poorly in the number of agents.[3] Thus, even though optimizing infinite-trials joint objectives might be rather easy *in theory* as Fact 5.2.1 suggests, it might result in poor performances *in practice*. On the other hand, the quantity is independent of the number of agents for disjoint and mixture objectives.

(ii) Looking at mixture objectives, the mismatch scales sub-linearly with the number of agents $\mathcal{N}$. Thus, in some sense, the number of agents has the same role as the number of trials: the more the agents the less the deployment mismatch, and at the limit, with $\mathcal{N} \to \infty$, the mismatch vanishes completely.[4] In other words, this result portraits a striking difference with respect to joint objectives: when facing state entropy maximization over mixtures, a reasonably high number of agents compared to the size of the state-space actually helps, and simple policy gradient over mixture objectives might be enough.

Remarks. One should notice that the results of Fact 5.2.1 are valid only for specific classes of policies, namely *centralized-information* policies of the form $\pi = (\pi^i \in \Delta_{\mathcal{S}}^{\mathcal{A}^i})_{i\in[|\mathcal{N}|]}$. To our knowledge, no guarantees are known for *decentralized-information*

[3] Indeed, in the case of product state-spaces $\mathcal{S} = \times_{i\in[|\mathcal{N}|]}\mathcal{S}_i$ the cardinality scales exponentially with the number of agents $|\mathcal{N}|$

[4] One should note that in this scenario, though, all the bounds of Lemma 5.2.1 linking different objectives become vacuous.

policies even in linear MGs. Interestingly though, the finite-trials formulation do offer additional insights on the behavior of optimal decentralized-information policies, a striking difference with respect to both the infinite-trial objectives and the linear MG interaction model in general. The interested reader can learn more about this in Appendix B.2.

5.3 Trust Region for Exploration in Practice

As stated before, a core drive of this work is addressing multi-agent state entropy maximization in practical scenarios. Yet, these cases are also the ones in which performing PG of infinite-trials objectives provide poor performance guarantees at deployment. In other words, here we address the third research question, that is:

> *Can we explicitly pre-train a policy for state entropy maximization in practical multi-agent scenarios?*

To do so, our attention will focus on the finite trials objectives explicitly, more specifically on the single-trial case with $K = 1$. Remarkably, it is possible to directly optimize the single-trial objective in multi-agent cases with decentralized algorithms: we introduce *Trust Region Pure Exploration* (TRPE), the first decentralized algorithm that explicitly addresses single-trial objectives in cMGs, with state entropy maximization as a special case. TRPE takes direct inspiration from trust-region based methods as TRPO [Schulman et al., 2015] for various reasons: a small change into the policy parameters of each agent may drastically change the value of the objective function, i.e., the optimization landscape is often brittle. The use of the trust region, like in TRPE, allows for accounting for this effect, as previous works have connected the trust region with the natural gradient [Pajarinen et al., 2019]; Additionally, trust-region methods recently enjoyed an ubiquitous success and interest for their surprising effectiveness in multi-agent problems [Yu et al., 2022].

In fact, trust-region analysis nicely align with the properties of finite-trials formulations and allow for an elegant extension to cMGs through the following.

Definition 5.3.1 (Surrogate Function over a Single Trial). *For every cMG $\mathcal{M}^{\mathcal{F}}$ equipped with a L-Lipschitz function $\mathcal{F}$ (see Ass. 2.3.1), let d_1 be a general single-trial distribution $d_1 = \{d_1, d_{1,i}, \tilde{d}_1\}$, then for any per-agent deviation over policies $\pi = (\pi^i, \pi^{-i})$, $\tilde{\pi} = (\tilde{\pi}^i, \pi^{-i})$, it is possible to define a per-agent* Surrogate Function $\mathcal{L}^i(\tilde{\pi}/\pi)$ *of the form*

$$\mathcal{L}^i(\tilde{\pi}/\pi) = \mathop{\mathbb{E}}_{d_1 \sim p_1^{\pi}} \rho^i_{\tilde{\pi}/\pi} \mathcal{F}(d_1),$$

where ρ^i is the per-agent importance-weight coefficient $\rho^i_{\tilde{\pi}/\pi} = p_1^{\tilde{\pi}}/p_1^{\pi} = \prod_{t\in[T]} \frac{\tilde{\pi}^i(\mathbf{a}^i[t]|\mathbf{s}^i[t])}{\pi^i(\mathbf{a}^i[t]|\mathbf{s}^i[t])}$, such that for $\zeta_1 \in \{\zeta_1, \zeta_1^i, \tilde{\zeta}_1\}$.

Algorithm 5.3: Trust-Region Pure Exploration (**TRPE**)

```
Input: exploration horizon $T$, number of trajectories $N$, trust-region threshold $\delta$, learning rate $\eta$.
initialize $\boldsymbol{\theta} = (\theta^i)_{i\in[|\mathcal{N}|]}$
for epoch = $1, 2, \dots,$ until convergence do
    Collect $N$ trajectories with $\pi_{\boldsymbol{\theta}} = (\pi^i_{\theta^i})_{i\in[|\mathcal{N}|]}$.
    for agent $i = 1, 2, \dots,$ concurrently do
        Construct datasets $\mathcal{D}^i = \{(\mathbf{s}^i_n, \mathbf{a}^i_n), \zeta^n_1\}_{n\in[N]}$
        $\theta^i \leftarrow$ IS-Optimizer($\mathcal{D}^i, \theta^i$)
    end for
end for
Output: (joint) policy $\pi_{\boldsymbol{\theta}} = (\pi^i_{\theta^i})_{i\in[|\mathcal{N}|]}$
```

IS-Optimizer

```
Input: Dataset $\mathcal{D}^i$, sampling parameter $\theta^i$.
Initialize $h = 0$ and $\theta^i_h = \theta^i$
while $D_{\mathrm{KL}}(\pi^i_{\theta^i_h} \| \pi^i_{\theta^i_0}) \leqslant \delta$ do
    Compute $\hat{\mathcal{L}}^i(\theta^i_h/\theta^i_0)$ via IS
    Perform Gradient step $\theta^i_{h+1} = \theta^i_h + \eta\nabla_{\theta^i_h}\hat{\mathcal{L}}^i(\theta^i_h/\theta^i_0)$
    $h \leftarrow h + 1$
end while
Output: parameters $\boldsymbol{\theta}_h$
```

From this definition, it follows that the trust-region algorithmic blueprint of Schulman et al. [2015] can be directly applied to single-trial formulations, with per-agent policies within a parametric space of stochastic differentiable policies $\Theta = \{\pi^i_{\theta^i} : \theta^i \in \Theta^i \subseteq \mathbb{R}^q\}$. In practice, KL-divergence is employed for greater scalability provided a trust-region threshold δ, we address the following optimization problem for each agent:

$$\max_{\tilde{\theta}^i\in\Theta^i} \mathcal{L}^i(\tilde{\theta}^i/\theta^i),$$
$$\text{s.t. } D_{\mathrm{KL}}(\pi^i_{\tilde{\theta}^i} \| \pi^i_{\theta^i}) \leqslant \delta$$

where we simplified the notation by letting $\mathcal{L}^i(\tilde{\theta}^i/\theta^i) := \mathcal{L}^i(\pi^i_{\tilde{\theta}^i}, \pi^{-i}_{\theta^{-i}}/\pi_\theta)$.[5]

The main idea then follows from noticing that the surrogate function in Definition 5.3.1 consists of an Importance Sampling (IS) estimator [Owen, 2013], and it is then possible to optimize it in a fully decentralized and off-policy manner, similarly to what was done in Metelli et al. [2020] for MDPs and in Mutti and Restelli [2020] for convex MDPs. More specifically, given a pre-specified objective of interest $\zeta_1 \in \{\zeta_1, \zeta^i_1, \tilde{\zeta}_1\}$, agents sample N trajectories $\{(\mathbf{s}_n, \mathbf{a}_n)\}_{n\in[N]}$ from the environment by following a (joint) policy with parameters $\boldsymbol{\theta}_0 = (\theta^i_0, \theta^{-i}_0)$. They then compute the values of the objective for each trajectory, building separate datasets $\mathcal{D}^i = \{(\mathbf{s}^i_n, \mathbf{a}^i_n), \zeta^n_1\}_{n\in[N]}$. Each agent uses her dataset to compute the Monte-Carlo approximation of the Surro-

[5] More precisely, $\mathcal{L}^i(\pi^i_{\tilde{\theta}^i}, \pi^{-i}_{\theta^{-i}}/\pi_\theta) = \mathbb{E}_{d_1\sim p_1^{\pi_\theta}} p_1^{\pi^i_{\tilde{\theta}^i}, \pi^{-i}_{\theta^{-i}}}/p_1^{\pi^i_{\theta^i}, \pi^{-i}_{\theta^{-i}}} \mathcal{F}(d_1)$.

gate Function, namely:

$$\hat{\mathcal{L}}^i(\theta^i_h/\theta^i_0) = \frac{1}{N}\sum_{n\in[N]} \rho^{i,n}_{\theta^i_h/\theta^i_0}\zeta^n_1,$$

where $\rho^{i,n}_{\theta^i_h/\theta^i_0} = \prod_{t\in[T]} \pi^i_{\theta^i_h}(\mathbf{a}^i_n[t]|\mathbf{s}^i_n[t])/\pi^i_{\theta^i_0}(\mathbf{a}^i_n[t]|\mathbf{s}^i_n[t])$ and ζ^n_1 is the plug-in estimator of the entropy based on the empirical measure d_1 [Paninski, 2003]. Finally, at each off-policy iteration h, each agent updates its parameter via gradient ascent $\theta^i_{h+1} \leftarrow \theta^i_h + \eta\nabla_{\theta^i_h}\hat{\mathcal{L}}^i(\theta^i_h/\theta^i_0)$ until the trust-region boundary is reached, i.e., when it holds $D_{\mathrm{KL}}(\pi^i_{\tilde{\theta}^i}\|\pi^i_{\theta^i}) > \delta$. The pseudo-code of TRPE is reported in Algorithm 5.3.

Limitations. The main limitations of the proposed methods are two. First, the Monte-Carlo estimation of single-trial objectives might be sample-inefficient in high-dimensional tasks. However, more efficient estimators of single-trial objectives remain an open question in single-agent convex RL as well, as the convex nature of the problem hinders the applicability of Bellman operators. Secondly, the plug-in estimator of the entropy is applicable to discrete spaces only, but designing scalable estimators of the entropy in continuous domains is usually a contribution *per se* [Mutti et al., 2021].

5.4 Numerical Validation

In this section, we address the last research question, that is:

Do crucial differences emerge in practice? Does this have an impact on downstream tasks learning?

by providing empirical corroboration of the findings discussed so far. Especially, we aim to answer the following questions: (**a**) Is Algorithm 5.3 actually capable of optimizing finite-trials objectives? (**b**) Do different objectives enforce different behaviors, as expected from Section 5.1? (**c**) Does the *clustering* behavior of mixture objectives play a crucial role? If yes, when and why?

Throughout the experiments, we will compare the result of optimizing finite-trial objectives, either joint, disjoint, mixture ones, through Algorithm 5.3 via fully decentralized-information policies. The experiments will be performed with different values of the exploration horizon T, so as to test their capabilities in different exploration efficiency regimes.[6]

Experimental Domains

The experiments were performed on two domains. The first is a notoriously difficult multi-agent exploration task called *secret room* [MPE, Liu et al., 2021],[7] referred to as Env. (**i**). In such task, two agents are required to reach a target while navigating over two rooms divided by a door. In order to keep the door open, at least one agent have to remain on a switch. Two switches are located at the corners of the two rooms. The hardness of the task then comes from the need of coordinated exploration, where one

[6] The exploration horizon T, rather than being a given trajectory length, has to be seen as a parameter of the exploration phase which allows to tradeoff exploration quality with exploration efficiency.

[7] We highlight that all previous efforts in this task employed centralized-information policies. We are interested on the role of the entropic feedback in fostering coordination rather than full-state conditioning, then maintaining fully decentralized-information policies instead.

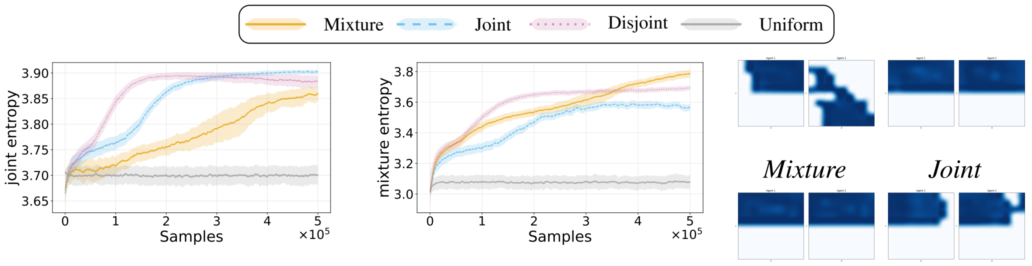


Figure 5.2: *Joint Entropy.*

Figure 5.3: *Mixture Entropy.*

Disjoint *Uniform*

*Single-trial Joint and Mixture Entropy induced by mixture, joint or disjoint objective optimization along a $T = 50$ horizon. (*Right*) State Distributions of two agents induced by different learned policies. We report the average and 95% c.i. over 4 runs.*

agent allows for the exploration of the other. The second is a simpler exploration task yet over a high dimensional state-space, namely a 2-agent instantiation of *Reacher* [MaMuJoco, Peng et al., 2021], referred to as Env. (**ii**). Each agent corresponds to one joint and equipped with decentralized-information policies conditioned on her own states. In order to allow for the use of plug-in estimator of the entropy [Paninski, 2003], each state dimension was discretized over 10 bins.

State-Entropy Maximization

As common for the unsupervised RL framework [Hazan et al., 2019, Laskin et al., 2021, Liu and Abbeel, 2021b, Mutti et al., 2021], Algorithm 5.3 was first tested in her ability to optimize for state entropy maximization objectives, thus in environments *without rewards*. First, we report the results for a short, and thus more challenging, exploration horizon $(T = 50)$ over Env. (**i**), as it is far more interpretable. Other experiments with longer horizons or over Env. (**ii**) can be found in Appendix C.3. Interestingly, at this challenging exploration regime, when looking at the joint entropy in Figure 5.2, joint and disjoint objectives perform rather well compared to mixture ones in terms of induced joint entropy, while they fail to address mixture entropy explicitly, as seen in Figure 5.3. On the other hand mixture-based objectives result in optimizing both mixture *and* joint entropy effectively, as one would expect by the bounds in Th. 5.2.1. By looking at the actual state visitation induced by the trained policies, the difference between the objectives is apparent. While optimizing joint objectives, agents exploit the high-dimensionality of the joint space to induce highly entropic distributions even without exploring the space uniformly via coordination; the same outcome happens in disjoint objectives, with which agents focus on over-optimizing over a restricted space loosing any incentive for coordinated exploration. On the other hand, mixture objectives enforce a clustering behavior and result in a better efficient exploration.

Policy Pre-Training via State-Entropy Maximization

More interestingly, we tested the effect of pre-training policies via different objectives as a way to alleviate the well-known hardness of sparse-reward settings, either throught faster learning or zero-short generalization. In order to do so, we employed a multi-agent counterpart of the TRPO algorithm [Schulman et al., 2015] with different pre-trained policies. First, we investigated the effect on the learning curve in the

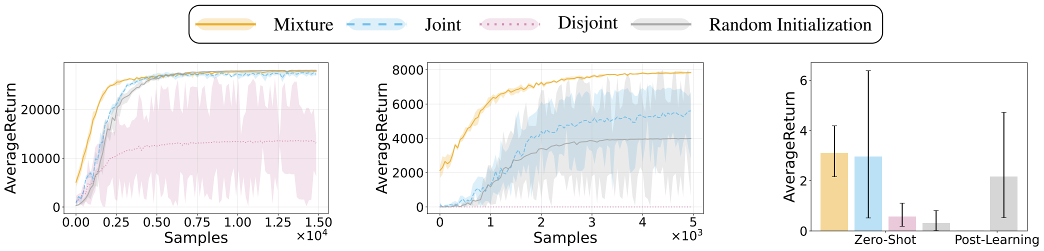


Figure 5.4: *MA-TRPO with TRPE Pre-Training (Env. (**i**), $T = 150$).*

Figure 5.5: *MA-TRPO with TRPE Pre-Training (Env. (**i**), $T = 50$).*

Figure 5.6: *MA-TRPO with TRPE Pre-Training (Env. (**ii**), $T = 100$).*

*Effect of pre-training in sparse-reward settings.(*left*) Policies initialized with either Uniform or TRPE pre-trained policies over 4 runs over a worst-case goal. (*rigth*) Policies initialized with either Zero-Mean or TRPE pre-trained policies over 4 runs over 3 possible goal state. We report the average and 95% c.i.*

hard-exploration task of Env. (**i**) under long horizons ($T = 150$), with a worst-case goal set on the the opposite corner of the closed room. Pre-training via mixture objectives still lead to a faster learning compared to initializing the policy with a uniform distribution. On the other hand, joint objective pre-training did not lead to substantial improvements over standard initializations. More interestingly, when extremely short horizons were taken into account ($T = 50$) the difference became appalling, as shown in Fig. 5.4: pre-training via mixture-based objectives lead to faster learning and higher performances, while pre-training via disjoint objectives turned out to be even *harmful* (Fig. 5.5). This was motivated by the fact that the disjoint objective overfitted the task over the states reachable without coordinated exploration, resulting in almost deterministic policies, as shown in in Appendix C.3. Finally, we tested the zero-shot capabilities of policy pre-training on the simpler but high dimensional exploration task of Env. (**ii**), where the goal was sampled randomly between worst-case positions at the boundaries of the region reachable by the arm. As shown in Fig. C.41, both joint and mixture were able to guarantee zero-shot performances via pre-training compatible with MA-TRPO after learning over $2e4$ samples, while disjoint objectives were not. On the other hand, pre-training with joint objectives showed an extremely high-variance, leading to worst-case performances not better than the ones of random initialization. Mixture objectives on the other hand showed higher stability in guaranteeing compelling zero-shot performance.

TakeAways

Overall, the proposed experiments managed to answer to all of the experimental questions: (**a**) Algorithm 5.3 is indeed able to optimize for finite-trial objectives; (**b**) **Mixture objectives enforce coordination**, essential when high efficiency is required, while joint or disjoint objectives may fail to lead to relevant solutions because of under or over optimization; (**c**) **The efficient coordination** through mixture objectives enforces the ability of **pre-training via state entropy maximization** to lead to **faster and better training** and even **zero-shot generalization**.

Concluding Remarks

In this chapter, we extended the state entropy maximization problem to Markov Games via a novel framework called Convex Markov Games. First of all, we showed that the task can be defined in several different ways: one can look at the joint distribution among all the agents, the marginals which are agent-specific, or the mixture which is a tradeoff of the two. Thus, we linked these three options via performance bounds and we show that while the first might enjoy nice theoretical guarantees, the others are more promising at working in practice, the latter in particular. Then, we designed a practical trust-region algorithm addressing more practical scenarios and we use it to confirm in a set of experiments the expected superiority of mixture objectives, due to its ability to enforce efficient but coordinated exploration over short horizons.

CHAPTER 6

Conclusions and Perspectives

In this thesis, we investigated the role of *unsupervised Reinforcement Learning* through the lens of *State Entropy Maximization* in settings that go beyond the classical single-agent, fully observable framework. In particular, we explored how this objective can be meaningfully extended to more realistic and challenging domains, such as partially observable environments and multi-agent systems. To this end, we introduced two novel classes of decision-making problems, *convex* Partially Observable Markov Decision Processes (cPOMDPs) and *convex* Markov Games (cMGs), and provided a thorough analysis of their theoretical foundations, as well as the practical implications of adopting entropy-based objectives within them.

Alongside these theoretical contributions, we proposed concrete pre-training methodologies that optimize more practical relaxations of the maximum state entropy objective. We showed how such approaches can yield broadly exploratory behaviors and significantly improve downstream performance when compared to training from scratch.

Importantly, these contributions were not straightforward extensions of prior work in simplified domains; rather, we uncovered that entropy maximization behaves in substantially different ways in partially observable and multi-agent settings, thus demanding new formulations and insights. While our results pave the way for scalable, general-purpose pre-training in complex Reinforcement Learning scenarios, several compelling research questions remain open. Below, we outline a few of these directions.

When are cPOMDPs tractable?
In Chapter 4, we highlighted a fundamental discrepancy between state and observation entropy in cPOMDPs, which complicates the direct use of entropy-based objectives. It remains an open question under which conditions, such as assumptions on the observation emission process, the form of the utility function, or the structure of the policy

class, cPOMDPs can be solved efficiently, either from a computational or statistical standpoint.

Can we devise more scalable algorithms for entropy maximization in cPOMDPs?
Also in Chapter 4, we showed that state entropy maximization can in principle be addressed via belief-based policies. However, devising practical algorithms for approximating belief states and optimizing such policies remains an open challenge. It would be of great interest to investigate whether approximate belief-learning techniques can capture the theoretical advantages promised by the entropy maximization objective.

What are the theoretical properties and practical benefits of cMGs?
In Chapter 5, we introduced cMGs as a framework for unsupervised pre-training in multi-agent settings. While we demonstrated practical ways to optimize entropy in these games, their general computational and statistical complexity is still poorly understood. Future work could explore under which structural assumptions cMGs become tractable. Additionally, extending the analysis to other convex objectives beyond entropy may help address fundamental multi-agent challenges such as coordination, imitation, competition, and non-stationarity.

In summary, this thesis contributes both foundational insights and algorithmic techniques for unsupervised policy pre-training in more realistic Reinforcement Learning scenarios. We hope that the introduced frameworks, results, and open questions will inspire further research into principled unsupervised pre-training for general-purpose Reinforcement Learning systems.

Bibliography

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse Reinforcement Learning. In *International Conference on Machine learning*, 2004.

David Abel, Will Dabney, Anna Harutyunyan, Mark K Ho, Michael Littman, Doina Precup, and Satinder Singh. On the expressivity of Markov Reward. In *Advances in Neural Information Processing Systems*, 2021.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, 2017.

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement Learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. FLAMBE: Structural complexity and representation learning of low rank MDPs. *Advances in Neural Information Processing Systems*, 33:20095–20107, 2020.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Reincarnating Reinforcement Learning: Reusing prior computation to accelerate progress. *Advances in Neural Information Processing Systems*, 35:28955–28971, 2022.

Siddhant Agarwal, Caleb Chuck, Harshit Sikchi, Jiaheng Hu, Max Rudolph, Scott Niekum, Peter Stone, and Amy Zhang. A unified framework for unsupervised Reinforcement Learning algorithms. In *Workshop on Reinforcement Learning Beyond Rewards@ Reinforcement Learning Conference 2025*, 2025a.

Siddhant Agarwal, Harshit Sikchi, Peter Stone, and Amy Zhang. Proto Successor Measure: Representing the behavior space of an RL agent. *Proceedings of the International Conference on Machine Learning (ICML)*, 2025b.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Pragnya Alatur, Giorgia Ramponi, Niao He, and Andreas Krause. Provably learning Nash policies in constrained Markov Potential Games. *CoRR*, abs/2306.07749, 2023.

Jean Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. FLAMINGO: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and modern approaches*. MIT Press, 2024.

Bibliography

Eitan Altman. *Constrained Markov Decision Processes*, volume 7. CRC Press, 1999.

ShunIchi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Christopher Amato, Daniel S Bernstein, and Shlomo Zilberstein. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 21(3):293–320, 2010.

Christopher Amato, George Konidaris, Leslie P Kaelbling, and Jonathan P How. Modeling and planning with macro-actions in decentralized POMDPs. *Journal of Artificial Intelligence Research*, 64:817–859, 2019.

Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in Atari. *Advances in Neural Information Processing Systems*, 32, 2019.

OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71:89–129, 2008.

Yuksel Arslantas, Ege Yuceel, Yigit Yalin, and Muhammed O. Sayin. Convergence of heterogeneous learning dynamics in zero-sum stochastic Games. *CoRR*, abs/2311.00778, 2023.

Karl Johan Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for Reinforcement Learning. *Advances in Neural Information Processing Systems*, 21, 2008.

Sarper Aydin and Ceyhun Eksin. Policy gradient play with networked agents in Markov Potential Games. In *Learning for Dynamics and Control Conference, L4DC 2023*, volume 211 of *Proceedings of Machine Learning Research*, pages 184–195. PMLR, 2023.

Qinbo Bai, Amrit Singh Bedi, Mridul Agarwal, Alec Koppel, and Vaneet Aggarwal. Achieving zero constraint violation for constrained Reinforcement Learning via primal-dual approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3682–3689, 2022.

Andrea Baisero and Christopher Amato. Learning internal state models in partially observable environments;. *Reinforcement Learning under Partial Observability, NeurIPS Workshop*, 2018.

Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (VPT): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022.

Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mo jtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sandra Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David J. Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378:1067 – 1074, 2022. doi: 10.1126/science.ade9097. URL `https://www.science.org/doi/abs/10.1126/science.ade9097`.

Bram Bakker. Reinforcement Learning with long short-term memory. In *Neural Information Processing Systems (NIPS)*, 2002.

Andrew G Barto, Satinder Singh, Nuttapong Chentanez, et al. Intrinsically motivated learning of hierarchical collections of skills. In *Proceedings of the 3rd International Conference on Development and Learning*, volume 112, page 19. Piscataway, NJ, 2004.

Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in Neural Information Processing Systems*, 29, 2016.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A Distributional Perspective on Reinforcement Learning. *arXiv:1707.06887 [cs, stat]*, July 2017. arXiv: 1707.06887.

Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023.

Richard Bellman. On the theory of dynamic programming. *Proceedings of the national Academy of Sciences*, 38 (8):716–719, 1952.

Stav Belogolovsky, Philip Korsunsky, Shie Mannor, Chen Tessler, and Tom Zahavy. Inverse Reinforcement Learning in contextual MDPs. *Machine Learning*, 110(9):2295–2334, 2021.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680*, 2019.

Daniel S Bernstein, Eric A Hansen, and Shlomo Zilberstein. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the nineteenth international joint conference on artificial intelligence (IJCAI)*, pages 52–57, 2005.

Dimitri P Bertsekas and John N Tsitsiklis. Introduction to probability. *EKLER Ek A: Sıralı Istatistik Ek B: Integrallerin Sayısal Hesabı Ek B*, 1, 2002.

Shalabh Bhatnagar and K Lakshmanan. An online actor-critic algorithm with function approximation for constrained Markov Decision Processes. *Journal of Optimization Theory and Applications*, 153(3):688–708, 2012.

L Bisi, L Sabbioni, E Vittori, M Papini, and M Restelli. Risk-averse trust region optimization for reward-volatility reduction. In *International Joint Conference on Artificial Intelligence*, 2020.

Byron Boots, Sajid M Siddiqi, and Geoffrey J Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research*, 30(7):954–966, 2011.

Vivek S Borkar. An actor-critic algorithm for constrained Markov Ddecision Processes. *Systems & control letters*, 54(3):207–213, 2005.

Joseph L Bower and Clark G Gilbert. *From Resource Allocation to Strategy*. Oxford University Press, 2005.

Michael Bowling, John D Martin, David Abel, and Will Dabney. Settling the reward hypothesis. In *International Conference on Machine Learning*, pages 3003–3020. PMLR, 2023.

Kianté Brantley, Miro Dudik, Thodoris Lykouris, Sobhan Miryoosefi, Max Simchowitz, Aleksandrs Slivkins, and Wen Sun. Constrained episodic Reinforcement Learning in concave-convex and knapsack settings. In *Advances in Neural Information Processing Systems*, 2020.

Daniel Brown, Scott Niekum, and Marek Petrik. Bayesian robust optimization for imitation learning. *Advances in Neural Information Processing Systems*, 33:2479–2491, 2020a.

Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020b.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *ICLR*, 2019.

Yang Cai, Haipeng Luo, Chen-Yu Wei, and Weiqiang Zheng. Uncoupled and convergent learning in two-player zero-sum Markov Games. *CoRR*, abs/2303.02738, 2023.

Bibliography

Dan A Calian, Daniel J Mankowitz, Tom Zahavy, Zhongwen Xu, Junhyuk Oh, Nir Levine, and Timothy Mann. Balancing constraints and rewards with meta-gradient. *International Conference on Learning Representations*, 2021.

Víctor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró-i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pages 1317–1327. PMLR, 2020.

Víctor Campos, Pablo Sprechmann, Steven Stenberg Hansen, André Barreto, Charles Blundell, Alex Vitvitskyi, Steven Kapturowski, and Adria Puigdomenech Badia. Coverage as a principle for discovering transferable behavior in Reinforcement Learning. *arXiv preprint arXiv:2102.13515*, 2021.

Diogo Carvalho, Francisco S Melo, and Pedro Santos. A new convergent variant of Q-Learning with linear function approximation. *Advances in Neural Information Processing Systems*, 33:19412–19421, 2020.

Anthony Cassandra, Michael L. Littman, and Nevin L. Zhang. Incremental pruning: a simple, fast, exact method for partially observable Markov Decision Processes. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI'97, pages 54–61, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1558604855.

Anthony R Cassandra, Leslie Pack Kaelbling, and Michael L Littman. Acting optimally in partially observable stochastic domains. In *Aaai*, volume 94, pages 1023–1028, 1994.

Anthony R Cassandra, Leslie Pack Kaelbling, and James A Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96*, volume 2, pages 963–972. IEEE, 1996.

Anthony Rocco Cassandra. *Exact and approximate algorithms for partially observable Markov Decision Processes*. PhD Thesis, Brown University, 1998.

Shicong Cen, Yuejie Chi, Simon Shaolei Du, and Lin Xiao. Faster last-iterate convergence of policy optimization in zero-sum Markov Games. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Dingyang Chen, Qi Zhang, and Thinh T. Doan. Convergence and price of anarchy guarantees of the softmax policy gradient in Markov Potential Games. *CoRR*, abs/2206.07642, 2022.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch Reinforcement Learning. In *International Conference on Machine Learning*, pages 1042–1051. PMLR, 2019.

Zaiwei Chen, Kaiqing Zhang, Eric Mazumdar, Asuman E. Ozdaglar, and Adam Wierman. A finite-sample analysis of payoff-based independent learning in zero-sum stochastic Games. *CoRR*, abs/2303.03100, 2023.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained Reinforcement Learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

Tianshu Chu, Sandeep Chinchali, and Sachin Katti. Multi-Agent Reinforcement Learning for networked system control. In *8th International Conference on Learning Representations, ICLR 2020*, 2020.

Scott Clayton. Lecture notes (topic 10) of eecs 598: Statistical learning theory. 2014.

Qiwen Cui and Simon S Du. When are offline two-player zero-sum Markov Games solvable? In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 25779–25791, 2022.

Marco Cusumano-Towner, David Hafner, Alex Hertzberg, Brody Huval, Aleksei Petrenko, Eugene Vinitsky, Erik Wijmans, Taylor Killian, Stuart Bowers, Ozan Sener, Philipp Kr̈henbühl, and Vladlen Koltun. Robust autonomy emerges from self-play. *arXiv preprint arXiv:2502.03349*, 2025. URL `https://arxiv.org/abs/2502.03349`.

Robert Dadashi, Leonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal Wasserstein imitation learning. In *International Conference on Learning Representations*, 2020.

Constantinos Daskalakis, Dylan J. Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Constantinos Daskalakis, Noah Golowich, and Kaiqing Zhang. The complexity of Markov equilibrium in stochastic Games. In *Conference on Learning Theory (COLT)*, volume 195 of *Proceedings of Machine Learning Research*, pages 4180–4234. PMLR, 2023.

Marisa P De Brito and Erwin A Van Der Laan. Inventory control with product returns: The impact of imperfect information. *European journal of operational research*, 194(1):85–101, 2009.

Vincenzo De Paola, Riccardo Zamboni, Mirco Mutti, and Marcello Restelli. Enhancing diversity in parallel agents: A maximum state entropy exploration story. *Proceedings of the International Conference on Machine Learning (ICML)*, 2025. URL `https://arxiv.org/abs/2505.01336`.

Marc Peter Deisenroth, Gerhard Neumann, Jan Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Jilles Steeve Dibangoye, Christopher Amato, Olivier Buffet, and François Charpillet. Optimally solving Dec-POMDPs as continuous-state MDPs. *Journal of Artificial Intelligence Research*, 55:443–497.

Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo R. Jovanovic. Independent policy gradient for large-scale Markov Potential Games: Sharper rates, function approximation, and game-agnostic convergence. In *International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 5166–5220. PMLR, 2022.

Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. Provably Efficient Reinforcement Learning with Aggregated States. February 2020. arXiv:1912.06366 [cs, math, stat].

Simon S Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudík, and John Langford. Provably efficient RL with Rich Observations via Latent State Decoding. 2019.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep Reinforcement Learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338. PMLR, 2016.

Miroslav Dudík and Robert E. Schapire. Maximum Entropy Distribution Estimation with Generalized Regularization. In *Learning Theory*, volume 4005, pages 123–138. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-35294-5 978-3-540-35296-9. doi: 10.1007/11776420_12. Series Title: Lecture Notes in Computer Science.

R. M. Dudley. *Uniform Central Limit Theorems*. Cambridge University Press, 1 edition, July 1999. ISBN 978-0-521-46102-3 978-0-521-05221-4 978-0-511-66562-2. doi: 10.1017/CBO9780511665622.

B.P. Duval, A. Abdolmaleki, M. Agostini, C.J. Ajay, S. Alberti, E. Alessi, G. Anastasiou, Y. Andrebe, G.M. Apruzzese, F. Auriemma, J. et. al. Ayllon-Guerola, F. Bagnato, A. Baillod, F. Bairaktaris, L. Balbinot, A. Balestri, M. Baquero-Ruiz, C. Barcellona, M. Bernert, W. Bin, P. Blanchard, J. Boedo, T. Bolzonella, F. Bombarda, L. Boncagni, M. Bonotto, and T.O.S.J. Bosman et Al. Experimental research on the TCV tokamak. *Nuclear Fusion*, 64(11):112023, oct 2024. doi: 10.1088/1741-4326/ad8361. URL `https://dx.doi.org/10.1088/1741-4326/ad8361`.

Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained MDPs. *arXiv preprint arXiv:2003.02189*, 2020.

Liad Erez, Tal Lancewicki, Uri Sherman, Tomer Koren, and Yishay Mansour. Regret minimization and convergence to equilibria in general-sum Markov Games. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 9343–9373. PMLR, 2023.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2018.

Benjamin Eysenbach, R. Salakhutdinov, and S. Levine. The Information Geometry of unsupervised Reinforcement Learning. *International Conference on Learning Representations*, 2021.

Bibliography

Dylan J Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. Offline Reinforcement Learning: Fundamental barriers for value function approximation. *arXiv preprint arXiv:2111.10919*, 2021.

Dylan J. Foster, Noah Golowich, and Sham M. Kakade. Hardness of independent learning and sparse equilibrium computation in Markov Games. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 10188–10221. PMLR, 2023.

Roy Fox, Stephen M Mcaleer, Will Overman, and Ioannis Panageas. Independent natural policy gradient always converges in Markov Potential Games. In *International Conference on Artificial Intelligence and Statistics*, pages 4414–4425. PMLR, 2022.

Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe Reinforcement Learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Oliver Bachem, Rémi Munos, and Olivier Pietquin. Concave utility Reinforcement Learning: The mean-field game viewpoint. In *International Conference on Autonomous Agents and Multiagent Systems*, 2022.

Ian Gemp, Andreas Haupt, Luke Marris, Siqi Liu, and Georgios Piliouras. Convex Markov Games: A framework for creativity, imitation, fairness, and safety in multiagent learning. *arXiv preprint arXiv:2410.16600*, 2025. URL `https://arxiv.org/abs/2410.16600`.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, 2020.

Angeliki Giannou, Kyriakos Lotidis, Panayotis Mertikopoulos, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. On the convergence of policy gradient methods to Nash equilibria in general stochastic games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Karol Gregor, Danilo Rezende, and Daan Wierstra. Variational intrinsic control. *International Conference on Learning Representations, Workshop Track*, 2017.

Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep Reinforcement Learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.

Zhaohan Daniel Guo, Mohammad Gheshlagi Azar, Alaa Saade, Shantanu Thakoor, Bilal Piot, Bernardo Avila Pires, Michal Valko, Thomas Mesnard, Tor Lattimore, and Rémi Munos. Geometric entropic exploration. *arXiv preprint arXiv:2101.02055*, 2021.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep Reinforcement Learning. *arXiv preprint arXiv:1812.11103*, 2018a.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-policy maximum entropy deep Reinforcement Learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. Pmlr, 2018b.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Steven Hansen, Will Dabney, Andre Barreto, David Warde-Farley, Tom Van de Wiele, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*, 2019.

Hado Hasselt. Double Q-Learning. *Advances in Neural Information Processing Systems*, 23, 2010.

Milos Hauskrecht and Hamish Fraser. Planning treatment of ischemic heart disease with partially observable Markov Decision Processes. *Artificial intelligence in medicine*, 18(3):221–244, 2000.

Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations*, 2018.

Elad Hazan, Adam Kalai, Satyen Kale, and Amit Agarwal. Logarithmic regret algorithms for online convex optimization. In *International Conference on Computational Learning Theory*, pages 499–513. Springer, 2006.

Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pages 2681–2691. PMLR, 2019.

Shuncheng He, Yuhang Jiang, Hongchang Zhang, Jianzhun Shao, and Xiangyang Ji. Wasserstein unsupervised Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6884–6892, 2022.

Nicolas Heess, Jonathan J Hunt, Timothy P Lillicrap, and David Silver. Memory-based control with recurrent neural networks. 2015. arXiv:1512.04455.

Verena Heidrich-Meisner and Christian Igel. Neuroevolution strategies for episodic Reinforcement Learning. *Journal of Algorithms*, 64(4):152–168, 2009.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep Reinforcement Learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, 2016.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pages 4399–4410. PMLR, 2020.

Baihe Huang, Jason D. Lee, Zhaoran Wang, and Zhuoran Yang. Towards general function approximation in zero-sum Markov Games. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári. Following the leader and fast rates in linear prediction: Curved constraint sets and other regularities. *Advances in Neural Information Processing Systems*, 29, 2016.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Maximilian Igl, Luisa M. Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational Reinforcement Learning for POMDPs. In *International Conference on Machine Learning (ICML)*, 2018.

Tommi Jaakkola, Satinder Singh, and Michael Jordan. Reinforcement Learning algorithm for partially observable Markov Decision Problems. *Advances in Neural Information Processing Systems*, 7, 1994.

Arnav Kumar Jain, Lucas Lehnert, Irina Rish, and Glen Berseth. Maximum state entropy exploration using predecessor and successor representations. *Advances in Neural Information Processing Systems*, 36:49991–50019, 2023.

Nan Jiang, Alex Kulesza, and Satinder P Singh. Improving predictive state representations via gradient descent. In *AAAI Conference on Artificial Intelligence*, pages 1709–1715, 2016.

Yuhang Jiang, Jianzhun Shao, Shuncheng He, Hongchang Zhang, and Xiangyang Ji. Spd: Synergy pattern diversifying oriented unsupervised multi-agent Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35:20661–20674, 2022.

Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for Reinforcement Learning. In *International Conference on Machine Learning*, pages 4870–4879. PMLR, 2020.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning - A simple, efficient, decentralized algorithm for multiagent RL. *CoRR*, abs/2110.14555, 2021a.

Bibliography

Ruiyang Jin, Zaiwei Chen, Yiheng Lin, Jie Song, and Adam Wierman. Approximate global convergence of independent learning in multi-agent systems. *arXiv preprint arXiv:2405.19811*, 2024.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offlineRL? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021b.

Yujia Jin, Vidya Muthukumar, and Aaron Sidford. The complexity of infinite-horizon general-sum stochastic Games. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023*, volume 251 of *LIPIcs*, pages 76:1–76:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

Michael Bradley Johanson, Edward Hughes, Finbarr Timbers, and Joel Z. Leibo. Emergent bartering behaviour in multi-agent Reinforcement Learning. *arXiv preprint arXiv:2205.06760*, 2022. URL `https://arxiv.org/abs/2205.06760`.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Fivos Kalogiannis and Ioannis Panageas. Zero-sum polymatrix Markov Games: Equilibrium collapse and efficient computation of Nash equilibria. *CoRR*, abs/2305.14329, 2023.

Fivos Kalogiannis, Ioannis Anagnostides, Ioannis Panageas, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Vaggos Chatziafratis, and Stelios Andrew Stavroulakis. Efficiently computing Nash equilibria in adversarial team Markov Games. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Emilien Kaufmann, Olivier Cappe, and Aurélien Garivier. Reward-free exploration in Reinforcement Learning. *NeurIPS*, 2021.

Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4436–4443, April 2020. doi: 10.1609/aaai.v34i04.5870.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep Reinforcement Learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4909–4926, 2021.

Jonathan Ko and Dieter Fox. Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models. *Autonomous Robots*, 27:75–90, 2009.

Jens Kober and Jan Peters. Policy search for motor primitives in robotics. *Advances in Neural Information Processing Systems*, 21, 2008.

Artemy Kolchinsky and Brendan Tracey. Estimating mixture entropy with pairwise distances. *Entropy*, 19(7): 361, July 2017. ISSN 1099-4300. doi: 10.3390/e19070361. URL `http://dx.doi.org/10.3390/e19070361`.

V. Koltchinskii and D. Panchenko. Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers. *The Annals of Statistics*, 30(1):1 – 50, 2002. doi: 10.1214/aos/1015362183.

Vijay Konda and John Tsitsiklis. Actor-Critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation learning via off-policy distribution matching. In *International Conference on Learning Representations*, 2019.

Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac Reinforcement Learning with rich observations. *Advances in Neural Information Processing Systems*, 29, 2016.

Alex Kulesza, Nan Jiang, and Satinder P Singh. Spectral learning of predictive state representations with insufficient statistics. In *AAAI Conference on Artificial Intelligence*, pages 2715–2721, 2015.

Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008, 2008.

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. Reinforcement Learning in reward-mixing MDPs. *Advances in Neural Information Processing Systems*, 34:2253–2264, 2021a.

Jeongyeol Kwon, Yonathan Efroni, Constantine Caramanis, and Shie Mannor. Rl for latent MDPs: Regret guarantees and a lower bound. *Advances in Neural Information Processing Systems*, 34:24523–24534, 2021b.

Nathan Lambert, Markus Wulfmeier, William Whitney, Arunkumar Byravan, Michael Bloesch, Vibhavari Dasagi, Tim Hertweck, and Martin Riedmiller. The challenges of exploration for offline Reinforcement Learning. *arXiv preprint arXiv:2201.11861*, 2022.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for Reinforcement Learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised Reinforcement Learning benchmark. 2021.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2020. URL `https://arxiv.org/abs/1906.05274`.

Stefanos Leonardos, Will Overman, Ioannis Panageas, and Georgios Piliouras. Global convergence of multi-agent policy gradient in Markov Potential Games. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

Yanjie Li, Baoqun Yin, and Hongsheng Xi. Finding optimal memoryless policies of POMDPs under the expected average reward criterion. *European Journal of Operational Research*, 211(3):556–567, 2011.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep Reinforcement Learning. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1509.02971`.

Yiheng Lin, Guannan Qu, Longbo Huang, and Adam Wierman. Multi-Agent Reinforcement Learning in stochastic networked systems. *Advances in Neural Information Processing Systems*, 34:7825–7837, 2021.

Daniel Y Little and Friedrich T Sommer. Learning and exploration in action-perception loops. *Frontiers in neural circuits*, 7:37, 2013.

Michael L. Littman. Markov games as a framework for multi-agent Reinforcement Learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 157–163. Morgan Kaufmann, San Francisco (CA), 1994a. ISBN 978-1-55860-335-6. doi: https://doi.org/10.1016/B978-1-55860-335-6.50027-1. URL `https://www.sciencedirect.com/science/article/pii/B9781558603356500271`.

Michael L Littman. Memoryless policies: Theoretical limitations and practical results. In *From Animals to Animats 3: Proceedings of the third international conference on simulation of adaptive behavior*, volume 3, page 238. MIT Press Cambridge, MA, USA, 1994b.

Michael L Littman, Richard S Sutton, and Satinder P Singh. Predictive representations of state. In *Neural Information Processing Systems (NIPS)*, 2002.

Hao Liu and Pieter Abbeel. APS: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747. PMLR, 2021a.

Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, 34:18459–18473, 2021b.

Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. Cooperative exploration for multi-agent deep Reinforcement Learning. In *International Conference on Machine Learning*, pages 6826–6836. PMLR, 2021.

Bibliography

Miao Liu, Kavinayan Sivakumar, Shayegan Omidshafiei, Christopher Amato, and Jonathan P How. Learning for multi-robot cooperation in partially observable stochastic environments with macro-actions. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1853–1860. IEEE, 2017.

Qinghua Liu, Alan Chung, Csaba Szepesvári, and Chi Jin. When is partially observable Reinforcement Learning not scary? In *Conference on Learning Theory*, pages 5175–5220. PMLR, 2022a.

Qinghua Liu, Csaba Szepesvári, and Chi Jin. Sample-efficient Reinforcement Learning of partially observable Markov Games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022b.

John Loch and Satinder Singh. Using eligibility traces to find the best memoryless policy in partially observable Markov Decision Processes. In *ICML*, volume 98, pages 323–331, 1998.

Yao Lu, Karol Hausman, Yevgen Chebotar, Mengyuan Yan, Eric Jang, Alexander Herzog, Ted Xiao, Alex Irpan, Mohi Khansari, Dmitry Kalashnikov, et al. Aw-opt: Learning robotic skills with imitation andreinforcement at scale. In *Conference on Robot Learning*, pages 1078–1088. PMLR, 2022.

Tung M Luu, Thang Vu, Thanh Nguyen, and Chang D Yoo. Visual pretraining via contrastive predictive model for pixel-based Reinforcement Learning. *Sensors*, 22(17):6504, 2022.

Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo. Learning parametric closed-loop policies for Markov Potential Games. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Chinmay Maheshwari, Manxi Wu, Druv Pai, and Shankar Sastry. Independent and decentralized learning in Markov Potential Games. *CoRR*, abs/2205.14590, 2022.

Yutaka Matsuo, Yann LeCun, Maneesh Sahani, Doina Precup, David Silver, Masashi Sugiyama, Eiji Uchibe, and Jun Morimoto. Deep learning, Reinforcement Learning, and world models. *Neural Networks*, 152:267–275, 2022.

R. Andrew McCallum. Overcoming incomplete perception with utile distinction memory. In *International Conference on Machine Learning (ICML)*, 1993.

Francisco S Melo and M Isabel Ribeiro. Q-Learning with linear function approximation. In *International Conference on Computational Learning Theory*, pages 308–322. Springer, 2007.

Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in Reinforcement Learning. In *International Conference on Machine Learning*, pages 7599–7608. PMLR, 2021.

Li Meng, Morten Goodwin, Anis Yazidi, and Paal Engelstad. Unsupervised state representation learning in partially observable Atari Games. In *International Conference on Computer Analysis of Images and Patterns*, pages 212–222. Springer, 2023.

Astrid Merckling. *Unsupervised Pretraining of State Representations in a Rewardless Environment*. PhD thesis, ISIR, Université Pierre et Marie Curie UMR CNRS 7222, 2021.

Alberto Maria Metelli, Matteo Papini, Nico Montali, and Marcello Restelli. Importance sampling techniques for policy optimization. *Journal of Machine Learning Research*, 21(141):1–75, 2020. URL `http://jmlr.org/papers/v21/20-124.html`.

Nicolas Meuleau, Leonid Peshkin, Kee-Eung Kim, and Leslie Pack Kaelbling. Learning finite-state controllers for partially observable environments. In *Uncertainty in Artificial Intelligence (UAI)*, pages 427–436, 1999.

Richard Meyes, Hasan Tercan, Simon Roggendorf, Thomas Thiele, Christian Büscher, Markus Obdenbusch, Christian Brecher, Sabina Jeschke, and Tobias Meisen. Motion planning for industrial robots using Reinforcement Learning. *Procedia CIRP*, 63:107–112, 2017.

Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V Albrecht. A survey of ad hoc teamwork research. In *European conference on multi-agent systems*, pages 275–293. Springer, 2022.

Sobhan Miryoosefi, Kianté Brantley, Hal Daume III, Miro Dudik, and Robert E Schapire. Reinforcement Learning with convex constraints. In *Advances in Neural Information Processing Systems*, 2019.

Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation Reinforcement Learning. In *International Conference on Machine Learning*, pages 6961–6971. PMLR, 2020.

Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep Reinforcement Learning. *Nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep Reinforcement Learning. In *International Conference on Machine Learning*, pages 1928–1937. PmLR, 2016.

Aditya Modi, Jinglin Chen, Akshay Krishnamurthy, Nan Jiang, and Alekh Agarwal. Model-free representation learning and exploration in low-rank MDPs. *Journal of Machine Learning Research*, 25(6):1–76, 2024.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. ISBN 978-0-262-03940-6.

Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden Markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 366–375, 2005.

Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon Markov Decision Process problems. *Journal of the ACM (JACM)*, 47(4):681–720, 2000.

Rémi Munos. Error bounds for approximate value iteration. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 1006. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

Mirco Mutti. *Unsupervised Reinforcement Learning via state entropy maximization*. PhD thesis, alma, Marzo 2023. URL https://amsdottorato.unibo.it/id/eprint/10588/.

Mirco Mutti and Marcello Restelli. An intrinsically-motivated approach for learning highly exploring and fast mixing policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5232–5239, 2020.

Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. Task-agnostic exploration via policy gradient of a nonparametric state entropy estimate. In *AAAI Conference on Artificial Intelligence*, 2021.

Mirco Mutti, Riccardo De Santi, Piersilvio De Bartolomeis, and Marcello Restelli. Challenging common assumptions in convex Reinforcement Learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022a. Curran Associates Inc. ISBN 9781713871088.

Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-Markovianity in maximum state entropy exploration. In *International Conference on Machine Learning*, 2022b.

Mirco Mutti, Stefano Del Col, and Marcello Restelli. Reward-free policy space compression for Reinforcement Learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3187–3203. PMLR, 2022c.

Mirco Mutti, Mattia Mancassola, and Marcello Restelli. Unsupervised Reinforcement Learning in multiple environments. In *AAAI Conference on Artificial Intelligence*, 2022d.

Mirco Mutti, Riccardo De Santi, Piersilvio De Bartolomeis, and Marcello Restelli. Convex Reinforcement Learning in finite trials. *Journal of Machine Learning Research*, 24(250):1–42, 2023. URL http://jmlr.org/papers/v24/22-1514.html.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. DualDICE: Behavior-Agnostic Estimation of Discounted Stationary Distribution Corrections. November 2019. URL http://arxiv.org/abs/1906.04733. arXiv:1906.04733 [cs, stat].

Bibliography

Ranjit Nair, Milind Tambe, Makoto Yokoo, David Pynadath, and Stacy Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, volume 3, pages 705–711, 2003.

Alexander Nedergaard and Matthew Cook. K-Means maximum entropy exploration. *arXiv preprint arXiv:2205.15623*, 2022.

Gerhard Neumann. Variational inference for policy search in changing situations. *Proceedings of the International Conference on Machine Learning (ICML)*, 2011.

Andrew Y Ng and Michael I Jordan. PEGASUS: A policy search method for large mdps and POMDPs. *arXiv preprint arXiv:1301.3878*, 2013.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:108, 2007.

Art B. Owen. *Monte Carlo theory, methods and examples*. `https://artowen.su.domains/mc/`, 2013.

Joni Pajarinen, Hong Linh Thai, Riad Akrour, Jan Peters, and Gerhard Neumann. Compatible natural gradient policy search. *Machine Learning*, 108(8):1443–1466, 2019.

Liam Paninski. Estimation of entropy and mutual information. *Neural Comput.*, 15(6):1191–1253, June 2003. ISSN 0899-7667. doi: 10.1162/089976603321780272. URL `https://doi.org/10.1162/089976603321780272`.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Chanwoo Park, Kaiqing Zhang, and Asuman E. Ozdaglar. Multi-player zero-sum Markov Games with networked separable interactions. *CoRR*, abs/2307.09470, 2023.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pages 2778–2787. PMLR, 2017.

Sarath Pattathil, Kaiqing Zhang, and Asuman E. Ozdaglar. Symmetric (optimistic) natural policy gradient for multi-agent learning with parameter convergence. In Francisco J. R. Ruiz, Jennifer G. Dy, and Jan-Willem van de Meent, editors, *International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5641–5685. PMLR, 2023.

Tim Pearce, Tabish Rashid, Dave Bignell, Raluca Georgescu, Sam Devlin, and Katja Hofmann. Scaling laws for pre-training agents and world models. *arXiv preprint arXiv:2411.04434*, 2024.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. FACMAC: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021.

Julien Perolat, Bruno Scherrer, Bilal Piot, and Olivier Pietquin. Approximate dynamic programming for two-player zero-sum Markov games. In *International Conference on Machine Learning (ICML)*, pages 1321–1329. PMLR, 2015.

Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of Stratego with model-free multiagent Reinforcement Learning. *Science*, 378(6623):990–996, 2022.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating Reinforcement Learning with learned skill priors. In *Conference on robot learning*, pages 188–204. PMLR, 2021.

Jan Peters and Stefan Schaal. Natural Actor-Critic. *Neurocomputing*, 71(7-9):1180–1190, 2008a.

Jan Peters and Stefan Schaal. Reinforcement Learning of motor skills with policy gradients. *Neural Networks*, 2008b.

Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for POMDPs. In *Ijcai*, volume 3, pages 1025–1032, 2003.

Miruna Pislar, David Szepesvari, Georg Ostrovski, Diana Borsa, and Tom Schaul. When should agents explore? *arXiv preprint arXiv:2108.11811*, 2021.

Pascal Poupart, Kee-Eung Kim, and Dongho Kim. Closing the gap: Improved bounds on optimal pomdp solutions. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 21, pages 194–201, 2011.

LA Prashanth and Mohammad Ghavamzadeh. Actor-Critic algorithms for risk-sensitive mdps. In *Advances in Neural Information Processing Systems*, 2013.

Martin L Puterman. *Markov Decision Processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Tiancheng Qin and S. Rasoul Etesami. Scalable and independent learning of nash equilibrium policies in n-player stochastic Games with unknown independent chains. *arXiv preprint arXiv:2312.01587*, 2023. URL `https://arxiv.org/abs/2312.01587`.

Zengyi Qin, Yuxiao Chen, and Chuchu Fan. Density constrained Reinforcement Learning. In *International Conference on Machine Learning*, pages 8682–8692. PMLR, 2021.

Guannan Qu, Adam Wierman, and Na Li. Scalable Reinforcement Learning for multiagent networked systems. *Operations Research*, 70(6):3601–3628, 2022.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. PIRLNAV: Pretraining with imitation andRL finetuning for objectnav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17896–17906, 2023.

Aviv Rosenberg and Yishay Mansour. Online convex optimization in adversarial Markov Decision Processes. In *International Conference on Machine Learning*, pages 5478–5486. PMLR, 2019.

Matthew Rosencrantz, Geoff Gordon, and Sebastian Thrun. Learning low dimensional predictive representations. In *International Conference on Machine Learning (ICML)*, 2004.

Gavin A Rummery and Mahesan Niranjan. *On-line Q-Learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Yagiz Savas, Michael Hibbard, Bo Wu, Takashi Tanaka, and Ufuk Topcu. Entropy maximization for partially observable Markov Decision Processes. *IEEE Transactions on Automatic Control*, 67(12):6948–6955, 2022. doi: 10.1109/TAC.2022.3183564.

Muhammed Sayin, Kaiqing Zhang, David Leslie, Tamer Basar, and Asuman Ozdaglar. Decentralized Q-Learning in zero-sum Markov games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 18320–18334, 2021.

Muhammed O. Sayin. Decentralized learning for stochastic Games: Beyond zero sum and identical interest. *CoRR*, abs/2310.07256, 2023.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Bruno Scherrer. Approximate policy iteration schemes: A comparison. In *International Conference on Machine Learning*, pages 1314–1322. PMLR, 2014.

Bibliography

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pages 222–227, 1991.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE transactions on autonomous mental development*, 2(3):230–247, 2010.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Wolfram Schultz. Neuronal reward and decision signals: from theories to data. *Physiological reviews*, 95(3):853–951, 2015.

Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*, 2021.

Younggyo Seo, Kimin Lee, Stephen L James, and Pieter Abbeel. Reinforcement Learning with action-free pre-training from videos. In *International Conference on Machine Learning*, pages 19561–19579. PMLR, 2022.

Guy Shani, Ronen I Brafman, and Solomon Eyal Shimony. Forward search value iteration for POMDPs. In *IJCAI*, pages 2619–2624, 2007.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *International Conference on Learning Representations (ICLR)*, 2019.

Harshit Sikchi, Andrea Tirinzoni, Ahmed Touati, Yingchen Xu, Anssi Kanervisto, Scott Niekum, Amy Zhang, Alessandro Lazaric, and Matteo Pirotta. Fast adaptation with behavioral foundation models. *arXiv preprint arXiv:2504.07896*, 2025. URL https://arxiv.org/abs/2504.07896.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pages 387–395. Pmlr, 2014.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Satinder Singh, Tommi Jaakkola, and Michael Jordan. Reinforcement Learning with Soft State Aggregation. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994a.

Satinder P Singh, Tommi Jaakkola, and Michael I Jordan. Learning without state-estimation in partially observable markovian decision processes. In *Machine Learning Proceedings 1994*, pages 284–292. Elsevier, 1994b.

Satinder P Singh, Michael L Littman, Nicholas K Jong, David Pardoe, and Peter Stone. Learning predictive state representations. In *International Conference on Machine Learning (ICML)*, 2003.

Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 1938.

Laura Smith, Ilya Kostrikov, and Sergey Levine. A walk in the park: Learning to walk in 20 minutes with model-free Reinforcement Learning. *arXiv preprint arXiv:2208.07860*, 2022. URL https://arxiv.org/abs/2208.07860.

Trey Smith and Reid Simmons. Heuristic search value iteration for POMDPs. *arXiv preprint arXiv:1207.4166*, 2012.

Edward J. Sondik. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Oper. Res.*, 26(2):282–304, 1978. ISSN 0030-364X. doi: 10.1287/opre.26.2.282. URL https://doi.org/10.1287/opre.26.2.282.

Matthijs TJ Spaan. Partially observable Markov Decision Processes. In *Reinforcement Learning: State-of-the-art*, pages 387–414. Springer, 2012.

Matthijs TJ Spaan and Nikos Vlassis. PERSEUS: Randomized point-based value iteration for POMDPs. *Journal of artificial intelligence research*, 24:195–220, 2005.

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and Reinforcement Learning in partially observed systems. *The Journal of Machine Learning Research*, 23(1):483–565, 2022.

L Enrique Sucar. Parallel Markov Decision Processes. *Advances in Probabilistic Graphical Models*, pages 295–309, 2007.

Tobias Sutter, David Sutter, Peyman Mohajerin Esfahani, and John Lygeros. Generalized maximum entropy estimation. *Journal of Machine Learning Research*, 20:138:1–138:29, 2017.

Richard S Sutton. The reward hypothesis. 2004. URL `http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/rewardhypothesis.html`. Accessed: 2025-02-18.

Richard S Sutton. Reinforcement Learning: An introduction. *A Bradford Book*, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for Reinforcement Learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. *Advances in Neural Information Processing Systems*, 20, 2007.

Umar Syed, Michael Bowling, and Robert E Schapire. Apprenticeship learning using linear programming. In *Proceedings of the 25th international conference on Machine learning*, pages 1032–1039, 2008.

Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Springer Nature, 2022.

Daniel Szer, François Charpillet, and Shlomo Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. *arXiv preprint arXiv:1207.1359*, 2012.

Aviv Tamar and Shie Mannor. Variance adjusted Actor-Critic algorithms. *arXiv preprint arXiv:1310.3697*, 2013.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, 2015.

Jean Tarbouriech and Alessandro Lazaric. Active exploration in Markov Decision Processes. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Jean Tarbouriech, Shubhanshu Shekhar, Matteo Pirotta, Mohammad Ghavamzadeh, and Alessandro Lazaric. Active model estimation in Markov Decision Processes. In *Conference on Uncertainty in Artificial Intelligence*, pages 1019–1028. PMLR, 2020.

Jean Tarbouriech, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. A provably efficient sample collection strategy for Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34:7611–7624, 2021.

Davide Tenedini, Riccardo Zamboni, Mirco Mutti, and Marcello Restelli. From parameters to behavior: Unsupervised compression of the policy space, 2025. URL `https://arxiv.org/abs/2509.22566`.

Kaleab Abebe Tessera, Leonard Hinckeldey, Riccardo Zamboni, David Abel, and Amos Storkey. Remembering the Markov Property in Cooperative MARL, 2025. URL `https://arxiv.org/abs/2507.18333`.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *International Conference on Learning Representations*, 2019.

Daniil Tiapkin, Denis Belomestny, Daniele Calandriello, Eric Moulines, Remi Munos, Alexey Naumov, Pierre Perrault, Yunhao Tang, Michal Valko, and Pierre Menard. Fast rates for maximum entropy exploration. In *International Conference on Machine Learning*, pages 34161–34221. PMLR, 2023.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. *arXiv preprint arXiv:2504.11054*, 2025. URL `https://arxiv.org/abs/2504.11054`.

Bibliography

Dhruva Tirumala, Alexandre Galashov, Hyeonwoo Noh, Leonard Hasenclever, Razvan Pascanu, Jonathan Schwarz, Guillaume Desjardins, Wojciech Marian Czarnecki, Arun Ahuja, Yee Whye Teh, et al. Behavior priors for efficient Reinforcement Learning. *Journal of Machine Learning Research*, 23(221):1–68, 2022.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

Ahmed Touati, Jeremy Rapin, and Yann Ollivier. Does zero-shot Reinforcement Learning exist?, 2023. URL `https://arxiv.org/abs/2209.14935`.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, et al. Jump-start Reinforcement Learning. In *International Conference on Machine Learning*, pages 34556–34583. PMLR, 2023.

Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer Series in Statistics. Springer New York, New York, NY, 1996. ISBN 978-1-4757-2547-6 978-1-4757-2545-2. doi: 10.1007/978-1-4757-2545-2.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with double Q-Learning. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 30, 2016.

Benjamin Van Roy. Performance Loss Bounds for Approximate Value Iteration with State Aggregation. *Mathematics of Operations Research*, 31(2):234–244, May 2006. ISSN 0364-765X, 1526-5471. doi: 10.1287/moor.1060.0188.

V. Vapnik. Principles of Risk Minimization for Learning Theory. In *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in Starcraft ii using multi-agent Reinforcement Learning. *Nature*, 575(7782):350–354, 2019.

Nikos Vlassis, Michael L Littman, and David Barber. On the computational complexity of stochastic controller optimization in POMDPs. *ACM Transactions on Computation Theory (TOCT)*, 4(4):1–8, 2012.

Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, Thanasis Lianeas, Panayotis Mertikopoulos, and Georgios Piliouras. No-regret learning and mixed Nash equilibria: They do not mix. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2007. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000001.

Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. On reward-free Reinforcement Learning with linear function approximation. *Advances in Neural Information Processing Systems*, 33:17816–17826, 2020.

Shaojun Wang, Russell Greiner, and Shaomin Wang. Consistency and Generalization Bounds for Maximum Entropy Density Estimation. *Entropy*, 15(12):5439–5463, December 2013. ISSN 1099-4300. doi: 10.3390/e15125439.

Tao Wang, Michael Bowling, and Dale Schuurmans. Dual representations for dynamic programming and Reinforcement Learning. In *2007 IEEE International symposium on approximate dynamic programming and Reinforcement Learning*, pages 44–51. IEEE, 2007.

Yuanhao Wang, Qinghua Liu, Yu Bai, and Chi Jin. Breaking the curse of multiagency: Provably efficient decentralized multi-agent RL with function approximation. *CoRR*, abs/2302.06606, 2023.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep Reinforcement Learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR, 2016.

Christopher JCH Watkins and Peter Dayan. Q-Learning. *Machine learning*, 8:279–292, 1992.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Last-iterate convergence of decentralized optimistic gradient descent/ascent in infinite-horizon competitive Markov Games. In *Conference on Learning Theory (COLT)*, volume 134 of *Proceedings of Machine Learning Research*, pages 4259–4299. PMLR, 2021.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J. Weinberger. Inequalities for the L1 deviation of the empirical distribution. 2003. URL `https://api.semanticscholar.org/CorpusID:12164823`.

Steven D Whitehead and Long-Ji Lin. Reinforcement Learning of non-Markov Decision Processes. *Artificial Intelligence*, 73(1-2):271–306, 1995.

Daan Wierstra, Alexander Foerster, Jan Peters, and Juergen Schmidhuber. Solving deep memory POMDPs with recurrent policy gradients. In *International Conference on Artificial Neural Networks (ICANN)*, 2007.

John K Williams and Satinder P Singh. Experimental results on learning stochastic memoryless policies for partially observable Markov Decision Processes. In *Neural Information Processing Systems (NIPS)*, pages 1073–1080, 1999.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist Reinforcement Learning. *Machine learning*, 8:229–256, 1992.

Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion gran turismo drivers with deep Reinforcement Learning. *Nature*, 602(7896):223–228, 2022.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34:27395–27407, 2021.

Tengyang Xie, Dylan J Foster, Yu Bai, Nan Jiang, and Sham M Kakade. The role of coverage in online Reinforcement Learning. *arXiv preprint arXiv:2210.04157*, 2022.

Pan Xu and Quanquan Gu. A finite-time analysis of Q-Learning with neural network function approximation. In *International Conference on Machine Learning*, pages 10555–10565. PMLR, 2020.

Qisong Yang and Matthijs TJ Spaan. Cem: Constrained entropy maximization for task-agnostic safe exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10798–10806, 2023.

Yuepeng Yang and Cong Ma. $O(t^{-1})$ convergence of optimistic-follow-the-regularized-leader in two-player zero-sum Markov Games. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement Learning with Prototypical Representations. In *International Conference on Machine Learning*, 2021.

Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline Reinforcement Learning. *arXiv preprint arXiv:2201.13425*, 2022.

Haotian Ye, Xiaoyu Chen, Liwei Wang, and Simon Shaolei Du. On the power of pre-training for generalization in RL: Provable benefits and hardness. In *International Conference on Machine Learning*, pages 39770–39800. PMLR, 2023.

Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for Reinforcement Learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Bin Yu. Rates of Convergence for Empirical Processes of Stationary Mixing Sequences. *The Annals of Probability*, 22(1):94–116, 1994. ISSN 0091-1798. Publisher: Institute of Mathematical Statistics.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent Games. *arXiv preprint arXiv:2103.01955*, 2022. URL `https://arxiv.org/abs/2103.01955`.

Peihong Yu, Manav Mishra, Syed Zaidi, and Pratap Tokekar. Tactic: Task-agnostic contrastive pre-training for inter-agent communication. *arXiv preprint arXiv:2501.02174*, 2025.

Bibliography

Tiancheng Yu, Yi Tian, Jingzhao Zhang, and Suvrit Sra. Provably efficient algorithms for multi-objective competitive Reinforcement Learning. In *International Conference on Machine Learning*, pages 12167–12176. PMLR, 2021.

Haoqi Yuan, Zhancun Mu, Feiyang Xie, and Zongqing Lu. Pre-training goal-based models for sample-efficient Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhecheng Yuan, Zhengrong Xue, Bo Yuan, Xueqian Wang, Yi Wu, Yang Gao, and Huazhe Xu. Pre-trained image encoder for generalizable visual Reinforcement Learning. *Advances in Neural Information Processing Systems*, 35:13022–13037, 2022.

Tom Zahavy, Alon Cohen, Haim Kaplan, and Yishay Mansour. Apprenticeship learning via Frank-Wolfe. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6720–6728, 2020.

Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. Reward is enough for convex MDPs. *Advances in Neural Information Processing Systems*, 34:25746–25759, 2021.

Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. Discovering policies with DOMiNO: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

Riccardo Zamboni, Alberto Maria Metelli, and Marcello Restelli. Distributional policy evaluation: a maximum entropy approach to representation learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 13127–13137. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/2a98af4fea6a24b73af7b588ca95f755-Paper-Conference.pdf`.

Riccardo Zamboni, Duilio Cirino, Marcello Restelli, and Mirco Mutti. How to explore with belief: state entropy maximization in POMDPs. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Riccardo Zamboni, Duilio Cirino, Marcello Restelli, and Mirco Mutti. The limits of pure exploration in POMDPs: When the observation entropy is enough. *RLJ*, 2:676–692, 2024b. URL `https://rlj.cs.umass.edu/2024/papers/Paper95.html`.

Riccardo Zamboni, Enrico Brunetti, and Marcello Restelli. Scalable multi-agent offline Reinforcement Learning and the role of information, 2025a. URL `https://arxiv.org/abs/2502.11260`.

Riccardo Zamboni, Mirco Mutti, and Marcello Restelli. Towards principled unsupervised multi-agent Reinforcement Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 38, 2025b.

Andrea Zanette, Alessandro Lazaric, Mykel J Kochenderfer, and Emma Brunskill. Provably efficient reward-agnostic navigation with linear value iteration. *Advances in Neural Information Processing Systems*, 33:11756–11766, 2020.

Sihan Zeng, Thinh Doan, and Justin Romberg. Regularized gradient descent ascent for two-player zero-sum Markov games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 34546–34558, 2022.

Wenhao Zhan, Baihe Huang, Audrey Huang, Nan Jiang, and Jason Lee. Offline Reinforcement Learning with realizability and single-policy concentrability. In *Conference on Learning Theory*, pages 2730–2775. PMLR, 2022.

Amy Zhang, Zachary C. Lipton, Luis Pineda, Kamyar Azizzadenesheli, Anima Anandkumar, Laurent Itti, Joelle Pineau, and Tommaso Furlanello. Learning causal state representations of partially observable environments. 2019. arXiv:1906.10437.

Chuheng Zhang, Yuanying Cai, Longbo Huang, and Jian Li. Exploration by maximizing Rényi entropy for reward-free RL framework. In *AAAI Conference on Artificial Intelligence*, 2021a.

Hao Zhang. Partially observable Markov Decision Processes: A geometric technique and analysis. *Operations Research*, 58(1):214–228, 2010.

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.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent Reinforcement Learning with networked agents. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 5867–5876. PMLR, 2018.

Kaiqing Zhang, Sham Kakade, Tamer Basar, and Lin Yang. Model-based multi-agent RL in zero-sum markov games with near-optimal sample complexity. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1166–1178, 2020b.

Runyu Zhang, Qinghua Liu, Huan Wang, Caiming Xiong, Na Li, and Yu Bai. Policy optimization for Markov Games: Unified framework and faster convergence. In *NeurIPS*, 2022a.

Runyu Zhang, Jincheng Mei, Bo Dai, Dale Schuurmans, and Na Li. On the global convergence rates of decentralized softmax gradient play in Markov Potential Games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022b.

Shangtong Zhang, Bo Liu, and Shimon Whiteson. Mean-variance policy iteration for risk-averse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 2021b.

Weihong Zhang. *Algorithms for partially observable Markov Decision Processes*. Hong Kong University of Science and Technology (Hong Kong), 2001.

Xuezhou Zhang, Yuzhe Ma, and Adish Singla. Task-agnostic exploration in Reinforcement Learning. *Advances in Neural Information Processing Systems*, 33:11734–11743, 2020c.

Yizhou Zhang, Guannan Qu, Pan Xu, Yiheng Lin, Zaiwei Chen, and Adam Wierman. Global convergence of localized policy iteration in networked multi-agent Reinforcement Learning. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(1):1–51, 2023.

Zihan Zhang, Simon S Du, and Xiangyang Ji. Nearly minimax optimal reward-free Reinforcement Learning. *arXiv preprint arXiv:2010.05901*, 2020d.

Zhaoyi Zhou, Zaiwei Chen, Yiheng Lin, and Adam Wierman. Convergence rates for localized Actor-Critic in networked Markov potential games. In *Uncertainty in Artificial Intelligence*, pages 2563–2573. PMLR, 2023.

Ev Zisselman, Itai Lavie, Daniel Soudry, and Aviv Tamar. Explore to generalize in zero-shotRL. *Advances in Neural Information Processing Systems*, 36:63174–63196, 2023.

Appendix

APPENDIX *A*

Maximum Entropy for Representation Learning

In this chapter, we provide an additional example of how Maximum Entropy formulations turn out to be effective tools for RL, introducing a new algorithm for representation learning that combines a Maximum Entropy principle with distributional Reinforcement Learning [dRL, Bellemare et al., 2023]. The content of this chapter is based on the paper *"Distributional Policy Evaluation: a Maximum Entropy approach to Representation Learning"* co-authored with Alberto Maria Metelli, and Marcello Restelli, and published at NeurIPS 2023.[1]

A.1 Preliminaries

In distributional Reinforcement Learning [Bellemare et al., 2023], an agent aims to estimate the entire distribution of the returns achievable by acting according to a specific policy. This is in contrast to and more complex than classic RL [Sutton, 2018, Szepesvári, 2022], where the objective is to predict the expected return only.

In recent years, several algorithms for dRL have been proposed, both in evaluation and control settings. The push towards distributional approaches was particularly driven by additional flavors they can bring into the discourse, such as risk-averse considerations, robust control, and many regularization techniques [Chow et al., 2017, Brown et al., 2020a, Keramati et al., 2020]. Most of them varied in how the distribution of the returns is modeled. The choice of the model was shown to have a cascading effect on how such a distribution can be learned, how efficiently and with what guarantees, and how it can be used for the control problem.

Due to these successes, one might wonder whether the potential of looking into the entire distribution of returns somehow transpires into the representation learning of the state-action spaces, that is to find a good feature representation of the decision-making space so as to make the overall learning problem easier, tenderly by reducing the dimensionality of such spaces.

Now, the RL literature proved that reducing the state space size while preserving the important features of the original state space is beneficial, namely with state-aggregation feature functions [Singh et al., 1994a, Van Roy, 2006, Dong et al., 2020]. This is particularly true when high dimensionality can make learning slower and more unstable, as in classic RL in general, or when the learning process is almost unfeasible in small-samples regimes, as for dRL, where learning the entire distribution of returns requires a large number of samples.

Thus, motivated by these considerations, while D-Max-Ent Policy Evaluation allows for the use of any type of structural constraint, this chapter focuses on state-aggregation feature functions, and we answer following question:

[1] A complete reference can be found in the bibliography [Zamboni et al., 2023]

How are representation learning and policy evaluation intertwined? Do distributional methods offer a new way to highlight and exploit this connection?

In order to answer this question, we first need to introduce two of the building blocks of the following results.

Distributions of Returns

Given an MDP $\mathcal{M}$ with discount factor γ, the *Discounted Return* is the sum of rewards received from the initial state onwards, discounted according to their time of occurrence:

$$\mathcal{G}^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s. \tag{A.1}$$

The *Value Function* of a given policy π is the expectation of this quantity under the policy itself:

$$V^{\pi}(s) = \mathbb{E}[\mathcal{G}(s)] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s\right]. \tag{A.2}$$

The *Return Distribution Function* η^{π} of a given policy π is a collection of distributions, one for each state $s \in \mathcal{S}$, where each element is the distribution of the random variable $\mathcal{G}^{\pi}(s)$:

$$\eta^{\pi}(s) = \mathcal{D}^{\pi}_{(s)}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s_0 = s\right], \tag{A.3}$$

where $\mathcal{D}^{\pi}_{(s)}$ extracts the probability distribution of a random variable under the joint distribution of the trajectory.

The *Distributional Policy Evaluation Problem* then consists of estimating the return distribution function of Eq. (A.3) for a fixed policy π.

Maximum Entropy Estimation

Maximum Entropy (Max-Ent) methods [Dudík and Schapire, 2006, Wainwright and Jordan, 2007, Sutter et al., 2017] are density estimation methods that select the distribution that maximizes the uncertainty, i.e., the one with maximum entropy.[2] Additionally, they assume that the learner has access to a feature mapping $\mathcal{F}$ from $\mathcal{X}$ to $\mathbb{R}^M$. In the most general case, we may have $M = +\infty$. We will denote by Φ the class of real-valued functions containing the component feature functions $f_j \in \mathcal{F}$ with $j \in [M]$.

A distribution p is *consistent* with the true underlying distribution p_0 if

$$\mathbb{E}_{x\sim p}[f_j(x)] = \mu_j, \quad \forall j \in [M], \tag{A.4}$$

where

$$\mu_j := \mathbb{E}_{x\sim p_0}[f_j(x)] \tag{A.5}$$

In this case, we say that p satisfies (in expectation) the structural constraints imposed by the features in $\mathcal{F}$. In practice, p_0 is not available and Max-Ent methods enforce empirical consistency over N independent and i.i.d. observations $\mathcal{D} = \{x_1, \dots, x_N\} \sim p_0$ with support in $\mathcal{X}$ by replacing the definition in Eq. (A.5) with

$$\hat{\mu}_j(\mathcal{D}) := \frac{1}{N} \sum_{i\in[1:N]} f_j(x_i), \quad \forall j \in [M]. \tag{A.6}$$

The distribution p is said to be consistent with the data $\mathcal{D}$ if it matches the empirical expectations. The empirical Max-Ent problem consists then of the following optimization problem

$$\begin{aligned} \max_{p\in\Delta(\mathcal{X})} \quad & \mathcal{H}(p) \\ \text{s.t.} \quad & \mathbb{E}_{x\sim p}[f_j(x)] = \hat{\mu}_j, \quad \forall j \in [M], \end{aligned} \tag{A.7}$$

with the optimization problem in expectation differing just in the constraints (i.e., replacing constraint from Eq. (A.6) with the ones from Eq. (A.5)). It is well known that the optimal solution to the empirical Max-Ent problem in

[2] With little abuse of notation, we will use the same symbol for the probability distribution and its p.d.f., which we assume to exist w.r.t. a reference measure.

Eq. (A.7) is a distribution $p_\lambda \in \Delta(\mathcal{X})$ belonging to the class of exponential distributions parametrized by the parameters λ, namely:

$$p_\lambda(x) = \Phi_\lambda \exp\left(\sum_{j\in[M]} \lambda_j f_j(x)\right), \tag{A.8}$$

where $\Phi_\lambda := \int_{\mathcal{X}} \exp\left(\sum_{j\in[M]} \lambda_j f_j(x')\right) dx'$ is a normalization constant, which ensures that $p \in \Delta(\mathcal{X})$, and its log-transformation takes the name of log-partition function $A(\lambda) := \log \int_{\mathcal{X}} \exp(\sum_{j\in[M]} \lambda_j f_j(x))dx$. The log-partition function defines the set of well-behaved distributions $\Omega = \{\lambda \in \mathbb{R}^M : A(\lambda) < +\infty\}$.

At optimality, the parameters are defined as $\hat{\lambda}$ and correspond to the optimal Lagrangian multipliers of the dual of the empirical Max-Ent problem in Eq. (A.7). Now on, we will use $\hat{p}$ to identify $p_{\hat{\lambda}}$ for simplicity.

A.2 A Maximum Entropy Approach to Distributional Policy Evaluation

The proposed approach turns distributional PE into a pure density estimation problem in a Max-Ent framework, called *Distributional Max-Ent Policy Evaluation*, as described in Algorithm 1. For this translation, the algorithm uses the distribution of returns η as p, N-trajectory samples $\mathcal{H}_N = \{\mathcal{H}\}_{n=0}^N$ as data, and a fixed set of features functions $\mathcal{F}$ belonging to a function class Φ. Note that to do this, we need to slightly change the notation concerning the dRL framework: η will not be a $|\mathcal{S}|$-vector of distributions with support over $\mathbb{R}$, but rather a joint distribution over the whole support $\mathcal{X} = \mathcal{S} \times \mathbb{R}$.

Algorithm A.2: Distributional Max-Ent Progressive Evaluation (**D-Max-Ent PE**)

```
Require: $(\mathcal{H}_N, \mathcal{F})$        ▷ $N$ trajectories, set of features
    $\hat{\eta} = \arg\max_{\eta} \mathcal{H}(\eta)$
    s.t. $\mathbb{E}_{X\sim\eta}[f_j(X)] = \hat{\mu}_j(\mathcal{H}_N) \quad \forall j \in [M]$
        $\eta \in \Delta(\mathcal{X})$
    return $\hat{\eta}$
```

Turning PE into a Max-Ent problem has many upsides. First of all, the Max-Ent principle allows to deal with any kind of support $\mathcal{X}$, unifying continuous and discrete cases under the same framework; secondly, it does not require specifying a family of probability distributions to choose from; moreover, it implicitly manages the uncertainty by seeking a distribution as agnostic as possible, i.e., as close to the uniform distribution as possible.

Finally, Max-Ent allows to include of structural constraints over the return distribution under many different flavors, both as in the standard value-function approximation methods [Van Roy, 2006] and as in more recent works based on statistical functionals acting over the return portion $\mathbb{R}$ of the support [Bellemare et al., 2023]. One of the possible limitations might be the requirement to have access to a batch of i.i.d. samples, but this is not necessarily restrictive: the result can be generalized for a single β-mixing sample path by exploiting blocking techniques [Yu, 1994, Nachum et al., 2019].

Generalization Error Bound

As previously said, the inner properties of Max-Ent allow for translating the results from density estimation methods to the distributional PE setting, and in particular, generalization-error bounds defined as KL-divergences.[3] Unfortunately, the generalization error bounds of traditional Max-Ent theory contain a conservative term that compares the solutions of the expectation and empirical Max-Ent problems, $\bar{\eta}, \hat{\eta}$ respectively, by taking the maximum between the 1-norm of the respective multipliers, namely $\max_{\lambda\in\{\bar{\lambda},\hat{\lambda}\}} ||\lambda||_1$. This quantity is bounded yet unknown, making the result unpractical.

In the following, we extend the previous results with a more practical bound containing $||\hat{\lambda}||_1$ instead of the maximum, requiring some additional assumptions about the expressiveness of the feature functions. This result is of independent interest and allows us to directly use the bound from an algorithmic perspective.

Theorem A.2.1 (Generalization Error Bound of D-Max-Ent PE)**.** *Assume that the set of features $\mathcal{F}$ belong to the function class Φ, which it is such that $\sup_{x\in\mathcal{X}, f\in\mathcal{F}} ||f(x)||_\infty = F < +\infty$ and that the minimum singular value $\sigma_{\min}$ of the empirical covariance matrix of the features $\hat{\mathbb{C}\text{ov}}(\mathcal{F})$ is strictly positive, namely $\sigma_{\min}(\hat{\mathbb{C}\text{ov}}(\mathcal{F})) > 0$. Then, given a sample batch $\{x_1, \dots, x_N\} \in \mathcal{X}^N$ of N i.i.d. points drawn from the true distribution η^π, for any $\delta \in (0,1)$, it holds with probability at least $1 - \delta$ that the solution to the sampled Max-Ent problem $\hat{\eta}$ satisfies the*

[3] The KL-divergence between two distributions p, q is defined as $KL(p||q) = \mathbb{E}_{x\sim p}[\log(p(x)/q(x))]$

following:

$$KL(\eta^\pi||\hat{\eta}) \lesssim -\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}) + B(\hat{\lambda}, \mathcal{F}, N, \delta) \tag{A.9}$$

$$\tilde{\mathcal{L}}(\hat{\eta}) = -\frac{1}{N}\sum_{i\in[0:N]} \log \hat{\eta}(x_i) \tag{A.10}$$

$$B(\hat{\lambda}, \mathcal{F}, N, \delta) = 10\|\hat{\lambda}\|_1 \left(\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right), \tag{A.11}$$

where $\lesssim$ stands for the fact that the bound comprises additional terms that decrease at a higher rate in sample complexity and were therefore neglected. $\mathcal{H}(\eta^\pi)$ and $\tilde{\mathcal{L}}(\hat{\eta})$, the empirical log-likelihood of the solution, form a bias term. The remaining term $B(\hat{\lambda}, \mathcal{F}, N, \delta)$ is a variance term depending on the multipliers characterizing the solution $\hat{\lambda}$, the number of samples, the confidence level δ, and the feature class complexity as the empirical Rademacher complexity of the class $\mathcal{R}_N(\Phi)$ [Mohri et al., 2018].

Proof Sketch. Here we report the main steps of the proof of Th. A.2.1. The interested reader can find the complete proof in Appendix B.3. First, define the set containing the solutions to the expected and sampled Max-Ent problems with $\mathcal{S} := \{\bar{\eta}, \hat{\eta}\}$, the related set for the multipliers $\Omega_\mathcal{S} := \{\bar{\lambda}, \hat{\lambda}\}$, and a quantity that will be central now on $h(x_1, \cdots, x_N) := \max_{\eta\in\mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N}\sum_{i\in[N]} \log \eta(x_i)|$. Then, the building blocks of the error term $KL(\eta^\pi||\hat{\eta})$, namely $KL(\bar{\eta}||\hat{\eta})$ and $KL(\eta^\pi||\bar{\eta})$ are bounded by:

$$KL(\bar{\eta}||\hat{\eta}) \leqslant 2h(\cdot)$$
$$KL(\eta^\pi||\bar{\eta}) \leqslant -\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}) + 3h(\cdot).$$

It is possible to show that:

$$h(\cdot) \leqslant 2\sup_{\lambda\in\Omega_\mathcal{S}} ||\lambda||_1 \left(\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right)$$
$$\sup_{\lambda\in\Omega_\mathcal{S}} ||\lambda||_1 \leqslant ||\hat{\lambda}||_1 + \sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\text{ov}}(\mathcal{F}))}h(\cdot)}.$$

The first inequality is obtained with standard methods as in van der Vaart and Wellner [1996], Dudley [1999], Koltchinskii and Panchenko [2002], Wang et al. [2013]. The second one is obtained by exploiting the intrinsic properties of the Max-Ent solution and by noting that it is possible to link $h(\cdot)$ with the Bregman divergence of the log-partition function $D_A(\bar{\lambda}, \hat{\lambda})$.

One can see that the use of the second inequality introduces an additional assumption about the expressiveness of the feature functions, requiring the minimum singular value of the sampled covariance matrix $\sigma_{\min}(\hat{\mathbb{C}\text{ov}}(\mathcal{F}))$ to be strictly positive. As a final step, setting $x = \sqrt{h(x_1, \cdots, x_N)}$ and combining the two previous inequalities yields a quadratic inequality:

$$\begin{cases} x^2 - bx - c \leqslant 0 \\ b = 2\sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\text{ov}}(\mathcal{F}))}}\left[\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right] \\ c = 2||\hat{\lambda}||_1\left[\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right] \end{cases},$$

which is well-defined and solves for

$$h(x_1, \cdots, x_N) \lesssim \|\hat{\lambda}\|_1 \left(\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right),$$

by neglecting higher-order terms. The statement of the theorem is then just a matter of combining all these results. □

A.3 Distributional Representation Learning with State Aggregation

This section addresses the research question, namely how to use the bound in Th. A.2.1 from an algorithmic perspective to automatically refine the features used to represent the state space in a principled way while performing D-Max-Ent PE. In particular, the focus is on a specific instance of feature functions for return distributions, namely state aggregation. More specifically, the state aggregation feature functions $\mathcal{F} = \{f_j\}_{j\in[M]}$ split the state space into M disjoint subsets, one for each function, i.e., $\mathcal{S} = \cup_{j\in[M]} S_j$ and $S_j \cap S_{j'} = \varnothing,\ j, j' \in [M],\ j \neq j'$, and gives back the associated return $g \in \mathbb{R}$, namely:

$$\begin{aligned} &f_j : \mathcal{S} \times \mathbb{R} \to \mathbb{R} \\ &f_j(s, g) = g\mathbf{1}_{[s\in S_j]}. \end{aligned} \tag{A.12}$$

These features are bounded by the maximum return G_{max}, while the empirical Rademacher complexity over N samples of returns $\{(s_i, g_i)\}_{i\in[N]}$ can be directly computed as in Clayton [2014]:

$$\mathcal{R}_N(\Phi) = G_{\text{max}} \sum_{j\in[M]} \sqrt{\hat{P}(S_j)}, \tag{A.13}$$

where $\hat{P}(S_j) = N_j/N$ and $N_j = |\{(g_i, s_i) : s_i \in S_j, i \in [N]\}|$. The decomposition of the Rademacher term into single terms leads to rewriting $B(\hat{\lambda}, \mathcal{F}, N, \delta)$ as in the following lemma.

Lemma A.3.1. *For Distributional Max-Ent Evaluation with a state-aggregation feature class, the variance term $B(\hat{\lambda}, \mathcal{F}, N, \delta)$ is given by;*

$$B(\hat{\lambda}, \mathcal{F}, N, \delta) = 10\|\hat{\lambda}\|_1 G_{max} \left(\sum_{j\in[M]} \sqrt{\hat{P}(S_j)} + \sqrt{\frac{\log 1/\delta}{2N}} \right). \tag{A.14}$$

Representation Refinement: Progressive Factorization

State aggregation features are of interest due to the possibility of progressively refining the representation by increasing the factorization level, that is, by splitting a subset S_j into further disjoint subsets. This refinement is called *progressive factorization* and is defined as follows.

Definition A.3.1 (Progressive Factorization)**.** *For two sets of state aggregation feature functions, $\mathcal{F}, \mathcal{F}_j$, we say that $\mathcal{F}_j$ is a progressive factorization of $\mathcal{F}$, i.e., $\mathcal{F} \subset \mathcal{F}_j$, if $\mathcal{F} = \{f_1, \ldots, f_{j-1}, f_{j+1}, \ldots, f_M\} \cup \{f_j\}, \mathcal{F}_j = \{f_1, \ldots, f_{j-1}, f_{j+1}, \ldots, f_M\} \cup \{f_j^k\}_{k\in[K]}$ and the additional functions $\{f_j^k\}_{k\in[K]}$ are such that the corresponding subsets satisfy*

$$S_j = \bigcup_{k\in[K]} S_j^k, \quad S_j^k \cap S_j^{k'} = \varnothing,\ k, k' \in [K],\ k \neq k',$$

where only non-degenerate class factorizations will be considered, meaning that the new subsets S_j^k are non-empty.

It is relevant for our interests that, in the case of progressive factorizations $\mathcal{F} \subset \mathcal{F}'$, the respective Max-Ent solutions enjoy the following monotonicity property

Lemma A.3.2 (Monotonicity)**.** *The multipliers of the Max-Ent solutions $\hat{\lambda}, \hat{\lambda}'$ using $\mathcal{F} \subset \mathcal{F}'$ are such that*

$$\|\hat{\lambda}\|_1 \leqslant \|\hat{\lambda}'\|_1. \tag{A.15}$$

This result ensures a monotonically increasing of all terms contained in the variance term of Eq. (A.11) since the complexity term is monotonically increasing by definition. On the other hand, the bias represented by Eq. (A.10) is guaranteed to decrease monotonically at finer levels of factorizations.

D-Max-Ent Progressive Factorization Algorithm

Algorithm B.1: Distributional Max-Ent Progressive Factorization

```
Require: $(\mathcal{H}_N, \mathcal{F}_0, \delta, \beta, K)$ ▷ $N$-trajectory samples, initial feature set, confidence level, boosting factor,
    factorization factor
    Done ← False, $i^* \leftarrow 0$
    while not Done do
        $\mathcal{F} \leftarrow \mathcal{F}_{i^*}, M \leftarrow |\mathcal{F}|$
        $\hat{\eta} \leftarrow$ D-Max-Ent PE$(\mathcal{H}_N, \mathcal{F})$
        $\mathcal{J}(\hat{\eta}) \leftarrow \beta\mathcal{L}(\hat{\eta}) + B(\hat{\lambda}, \mathcal{F}, N, \delta)$
        $\{\mathcal{F}_j\}_{j\in[M]} \leftarrow$ Progressive Factor$(\mathcal{F}, K)$
        for $j \in [M]$ do
            $\hat{\eta}_j \leftarrow$ D-Max-Ent PE$(\mathcal{H}_N, \mathcal{F}_j)$
            $\mathcal{J}(\hat{\eta}_j) \leftarrow \beta\mathcal{L}(\hat{\eta}_j) + B(\hat{\lambda}_j, \mathcal{F}_j, N, \delta)$
            if $\mathcal{J}(\hat{\eta}_j) < \mathcal{J}(\hat{\eta})$ then
                $i^* \leftarrow j$
            end if
        end for
        if $\mathcal{F}_{i^*} == \mathcal{F}$ then
            Done ← True
        end if
    end while
    return $\hat{\eta}_{i^*}$
```

In summary, D-Max-Ent PE shows a generalization error bound whose quantities are either known or estimated and change monotonically between progressive factorizations. On these results, we build an algorithm called *D-Max-Ent Progressive Factorization*, shown in Algorithm 2, which iteratively constructs a sequence of feature sets $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \dots$ with progressive factorization while performing PE.

The behavior of the algorithm is similar to what is done in Structural Risk-Minimization [SRM, Vapnik, 1991], and it involves optimizing for a trade-off: the bias term (i.e., empirical risk) decreases by taking into account more complex features classes, while the variance term (i.e., the confidence interval) increases. The whole algorithm is then based on the progressive search for the new set of feature functions which reduces a proxy of the generalization error bound of D-Max-Ent PE:

$$\mathcal{J}(\hat{\eta}) = \beta\mathcal{L}(\hat{\eta}) + B(\hat{\lambda}, \mathcal{F}, N, \delta), \tag{A.16}$$

and the procedure will continue until there are no further improvements in the trade-off. Due to the nature of the proxy function, the role of $\beta > 0$ is to regulate the tendency to factorize. Higher values of β will increase the magnitude of the decreasing term, causing a boost in the tendency to factorize. On the other hand, lower values will further decrease the importance of this term, resulting in a lower tendency to factorization.

Finally, the *Progressive Factor* function takes as input the list of feature functions and a factor K and returns a list of progressively factored set of feature functions. More specifically, each element in $\{\mathcal{F}_j\}_{j\in[M]}$ corresponds to a progressive factorization of the feature f_j, factoring the related subset S_j into K disjoint subsets as in Definition A.3.1. The new K subsets $\{\mathcal{S}_k^j\}_{k\in[K]}$ are constructed in the worst-case scenario: the complexity term in Eq. (A.13) is maximized with partitions of a set leading to a uniform distribution of samples in each new partitioned subset, and since it is not possible to know in advance which samples will be contained in which new subset, one way is then to proceed with a uniform factorization. We decided to maintain the most agnostic approach over the set of possible features, but prior knowledge could be used to narrow down the partitions to consider.

A.4 Numerical Validation

This section reports the results of some illustrative numerical simulations that make use of Algorithm 2.

Simulations Objectives. The objective of the simulations is to illustrate two essential features of the proposed method that were only suggested by the theoretical results. First of all, to analyze the outcome of performing policy evaluations with aggregated states at different sample regimes, by comparing the output of the proposed algorithm with some relevant baseline distributions. Secondly, the aim is to study the role of the boosting parameter β and the sampling regime N, being the main hyper-parameters of Algorithm 2, in the tendency to factor the representation at utterly different sample regimes.

MDP Instance Design. The effectiveness of the proposed approach is expected to be particularly evident in MDPs admitting a factored representation of the return distribution, namely the ones in which many states are nearly equivalent under the evaluation of a policy. This factorizability property is not uncommon in general since it is present in any environment with symmetries and Block-MDPs [Du et al., 2019] as well. The MDP instance is then designed to be a Block-MDP indeed since it allows for better evaluate the simulation objectives: one would expect that operating on MDPs admitting a factored representation would allow for lower values of β to be effective enough, while a higher level of boosting would force over-factorizations that are unnecessary, leading to no further improvement or even degradation of the results. The simulations are run on a rectangular GridWorld, with a height of 4 and length of 8, with traps on the whole second line and goals all over the top. The policy is selected as a uniform distribution over the set of actions $\mathcal{A} = (\text{up, left, right})$.

Performance Indexes. The proposed MDP instance presents many upsides in terms of the interpretability of the output as well. First of all, it allows us to directly compute the true underlying return distribution with Monte-Carlo estimation. Secondly, it permits to compare of the output distribution of the algorithm with the result of performing plain Distributional Max-Ent Policy Evaluation (Algorithm 1) with two baseline representations: an **oracle factorization** that aggregates together states known to be equivalent under the policy, and in particular all the upper and lower states respectively; a **full factorization** that employs $|\mathcal{S}|$-singletons of states as representations, i.e., the most fine-grained representation possible. The comparison is made via two relevant quantities, the KL divergence with respect to the true distribution (the *bounded quantity*), and the total bound $\mathrm{B}_{\text{tot}} = \tilde{\mathcal{L}}(\hat{\eta}) + B(\hat{\lambda}, \mathcal{F}, N, \delta)$ (the *bounding quantity*). Finally, the value of the partition splitting K is set to 2, to reduce the exponential search space of all possible uniform partitions, the discount factor γ is set to 0.98 and the confidence δ to 0.1, the results are averaged over 10 rounds with the respective standard deviation.

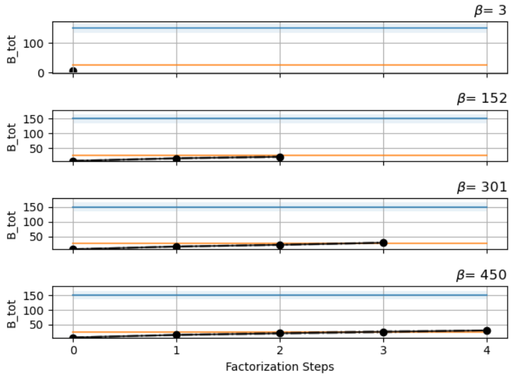
Results Discussion. The results of the simulations are reported from Fig. A.1 to Fig. A.4, with the quantity related to the oracle parametrization being in **orange**, while the ones related to the full parametrization being in **blue**. It is possible to notice that these two distributions have almost the same KL divergences with respect to the true return distribution (Fig. A.2, A.4), yet they highly differ in the bound B_{tot} (Fig. A.1, A.3) mostly due to the variance term, which is way higher in the case of full factorization.

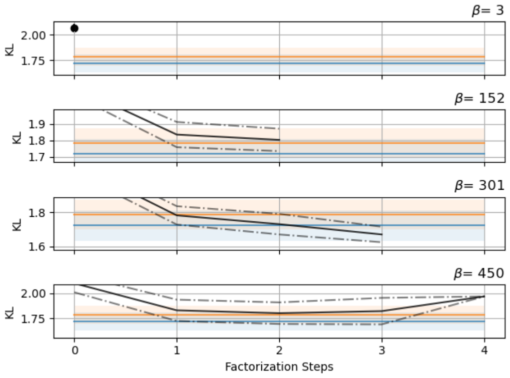
This suggests that the bound is indeed able to distinguish between the two. The plotting of the outputs of Algorithm 2 stops at the optimal number of factorization steps found for different values of β, namely at $\mathcal{F}_{i^\star}$.

The plots should be read as follows: while the bound term B_{tot} is expected to increase at each factorization step, the KL divergences with respect to the true return distribution should decrease as much as possible. In all cases, it is evident that the value of β pushes towards a higher number of factorization steps, going from performing no factorization at all using low values ($\beta = 3$), to performing up to 4 factorization steps even in this simple scenario with higher values ($\beta = 450$), both at low and high sample regimes ($N \in \{50, 1000\}$).

Furthermore, at higher sample regimes, it is possible to see how the higher quality of the estimation counteracts the action of β, and increasing it generally induces still fewer factorizations compared to the low sample regimes with same values of β, as in Fig. A.3, A.4.

Finally, it is apparent that minimizing for Eq. (A.16) successfully decreases the KL divergence. Nonetheless, its values stop decreasing significantly after the first factorization, which splits the state space over the two rows and further factorizations might lead to performance degradation as well.

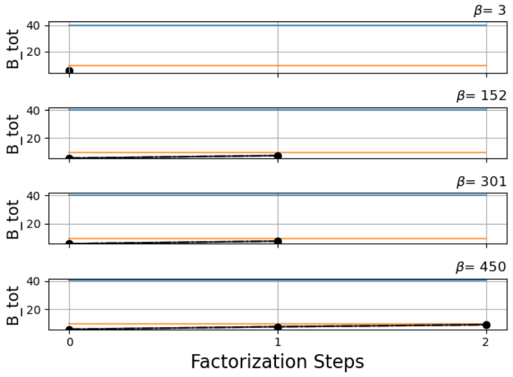




t

Figure A.1: *Bound Trend for different* β *(*$N = 50$*)*

Figure A.2: *KL Trend for different* β *(*$N = 50$*)*



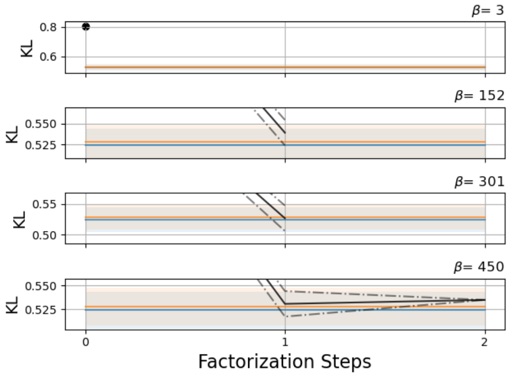


Figure A.3: *Bound Trend for different* β *(*$N = 1000$*)*

Figure A.4: *KL Trend for different* β *(*$N = 1000$*)*

Concluding Remarks

In this section, we presented in a dRL framework a new policy evaluation approach based on Maximum Entropy density estimation, called Distributional Max-Ent Policy Evaluation, which benefits from the learning guarantees of Max-Ent and the generality of the setting, being able to enforce even complex feature families.

We extended previous results and derived a practical formulation of the generalization error bound, which contains only estimated and known quantities of the problem. We then instantiated a particular class of features, namely state aggregation, and we proposed an algorithm called Distributional Max-Ent Progressive Factorization to adaptively find a feature representation that optimizes for a proxy of the generalization error bound in a Structural Risk Minimization fashion.

In this way, we showed that performing PE can indeed drive the learning of a reduced-dimension representation in the distributional setting. We then provided illustrative simulations showing the empirical behaviors of these approaches, while clarifying the links between some hyperparameters and the sample regime.

Much of our analysis and theoretical guarantees straightforwardly extend to other feature classes, and an open question is to investigate other instances of features and settings that can benefit from the proposed framework.

APPENDIX $\mathcal{B}$

Missing Proofs

B.1 Proofs of Chapter 4

Belief MDPs Interestingly, the MBE objective has a clean and neat equivalent formulation in belief-state POMDPs that can be turned into a dual problem as for MDPs, yet the resulting problem is still intractable. More specifically, having defined belief states, we can encode the POMDP $\mathcal{M}$ into a corresponding *belief* MDP $\mathcal{M}_{\mathcal{B}} := (\mathcal{B}, \mathcal{A}, \widetilde{\mathbb{P}}, \mathbb{B}, b_0, T)$ where

- $\mathcal{B}$ is a finite set of states such that each $b \in \mathcal{B}$ corresponds to a belief state, and $\mathcal{B}$ is obtained by running Algorithm B.1 in $\mathcal{M}$;
- $\mathcal{A}$ is the set of actions in $\mathcal{M}$;
- $\widetilde{\mathbb{P}} : \mathcal{B} \times \mathcal{A} \to \Delta_{\mathcal{B}}$ is the transition model of the belief MDP defined in a few lines;
- $b_0 \in \mathcal{B}$ is the initial state;
- T is the horizon length.

To fully characterize $\mathcal{M}_{\mathcal{B}}$, we can extract the transition model $\widetilde{\mathbb{P}}$ from $\mathcal{M}$ as

$$\begin{aligned}\widetilde{\mathbb{P}}(b'|b,a) &= \sum_{\{o \in \mathcal{O} | b' = \mathbb{T}^{ao}(b)\}} P(o|b,a) = \sum_{\{o \in \mathcal{O} | b' = \mathbb{T}^{ao}(b)\}} \sum_{s \in \mathcal{S}} P(o|s) P(s|b,a) \\ &= \sum_{\{o \in \mathcal{O} | b' = \mathbb{T}^{ao}(b)\}} \sum_{s \in \mathcal{S}} \mathbb{O}(o|s) \sum_{s' \in \mathcal{S}} b(s') \mathbb{P}(s|s',a).\end{aligned}$$

Let us denote as $d^\pi \in \Delta_{\mathcal{S}}$ the expected finite-horizon state distribution induced by a policy $\pi \in \Pi_{\mathcal{I}}$ on the true (unobserved) states. Then, we can define the objective function of our problem as

$$\max_{\pi \in \Pi_{\mathcal{I}}} \mathcal{F}(d^\pi) = \min_{\pi \in \Pi_{\mathcal{I}}} \mathop{\mathbb{E}}_{s \sim d^\pi} [\log d^\pi(s)] \tag{B.1}$$

Following standard techniques for MDPs [Puterman, 2014], we can obtain the optimal planning policy for (B.1) by solving the dual convex program

$$\begin{aligned} &\underset{\substack{\boldsymbol{d}\in\Delta_{\mathcal{S}} \\ \{\boldsymbol{\omega_t}\in\Delta_{\mathcal{B}\times\mathcal{A}}\}_{t\in[1:T]}}}{\text{maximize}} \quad \mathcal{F}(d) \\ &\text{subject to} \quad \sum_{a'\in\mathcal{A}} \omega_{t+1}(b',a') = \sum_{b\in\mathcal{B},a\in\mathcal{A}} \omega_t(b,a)\widetilde{\mathbb{P}}(b'|b,a) \qquad \forall b'\in\mathcal{B},\ \forall t\in 1\dots T \\ &\qquad\qquad d(s) = \frac{1}{T}\sum_{t\in[T]}\sum_{b\in\mathcal{B},a\in\mathcal{A}} \omega_t(b,a)b(s) \qquad \forall (s,a)\in\mathcal{S}\times\mathcal{A} \end{aligned}$$

and then obtaining the resulting (non-stationary) policy from the solution $\boldsymbol{\omega}^*$ as $\pi_t(a|b) = \omega^*_t(b,a)/\sum_{a'\in\mathcal{A}}\omega^*_t(b,a')$, $\forall (b,a)\in\mathcal{B}\times\mathcal{A}$. As one may notice, while this problem has a neat and concise formulation, the dimensionality of the optimization problem does not scale with the dimension of $\mathcal{M}$.

Belief Set Computation The belief states set reachable in a T step interaction with a POMDP can be computed via the following Algorithm B.1:

Algorithm B.1: Belief set

```
Input: belief $b$, set $\mathcal{B}$, step $t$, horizon $T$
if $t < T$ then
    for $(o,a) \in \mathcal{O}\times\mathcal{A}$ do
        $b' = T^{ao}(b)$
        if $b' \notin \mathcal{B}$ then
            $\mathcal{B} = BeliefSset(b', \mathcal{B}\cup\{b'\}, t+1, T)$
        end if
    end for
end if
return $\mathcal{B}$
```

Proofs of Policy Gradients Computation

Proposition 4.2.1 ((General) Policy Gradient for single-trial cPOMDPs). *Let* $\pi_\theta \in \Pi_{\mathcal{I}}$ *a policy parametrized by* $\theta\in\Theta\subseteq\mathbb{R}^{IA}$, *and let the* policy scores $\nabla_\theta \log\pi_\theta(\mathbf{ia}) = \sum_{t\in[T]}\nabla_\theta\log\pi_\theta(\mathbf{a}[t]|\mathbf{i}[t])$. *We can compute the* policy gradient *of* π_θ *as*

$$\nabla_\theta \mathcal{J}_{1,\mathcal{O}}(\pi_\theta) = \mathop{\mathbb{E}}_{\mathbf{ia}\sim p^{\pi_\theta}_{\mathcal{IA},1}}\Big[\nabla_\theta\log\pi_\theta(\mathbf{ia})\mathcal{F}(d_{\mathcal{I}}(\cdot|\mathbf{i}))\Big], \tag{4.32}$$

where $\mathcal{I}\in\{\mathcal{S},\mathcal{O}\}$.

Theorem 4.2.3. *For a policy* $\pi_\theta\in\Pi_{\mathcal{I}}$ *parametrized by* $\theta\in\Theta\subseteq\mathbb{R}^{SA}$, *we have*

$$\nabla_\theta\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_\theta) = \mathop{\mathbb{E}}_{\boldsymbol{b}\sim p^\pi_{\mathcal{B}}}\ \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim p(\cdot|\boldsymbol{b})}\Big[\nabla_\theta\log\pi_\theta(\tilde{\mathbf{s}})\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))\Big], \tag{4.34}$$

where $\nabla_\theta\log\pi_\theta(\tilde{\mathbf{s}})$ *are defined as in 4.2.1. Additionally, let* $\mathcal{T}_{\mathcal{B}}(\pi_1,\pi_2) = \{\boldsymbol{b}\in\mathcal{T}_{\mathcal{B}} : p^{\pi_1}(\boldsymbol{b}) > 0 \vee p^{\pi_2}(\boldsymbol{b}) > 0\}$, $\boldsymbol{b}^\star = \arg\max_{\boldsymbol{b}\in\mathcal{T}_{\mathcal{B}}(\pi_1,\pi_2)}\mathbb{E}_{\tilde{\mathbf{s}}\sim p(\cdot|\boldsymbol{b})}\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$, *and* $\bar{\mathcal{F}}(\boldsymbol{b}^\star) = \mathbb{E}_{\tilde{\mathbf{s}}\sim p(\cdot|\boldsymbol{b})}\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$, *we have*

$$|\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_1) - \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_2)| \leqslant T\bar{\mathcal{F}}(\boldsymbol{b}^\star)d^{TV}(\pi_1,\pi_2). \tag{4.35}$$

Proof. Let us denote $\mathbf{h} = \mathbf{s}\oplus\mathbf{o}\oplus\boldsymbol{b}\oplus\tilde{\mathbf{s}}$ and for a generic $i\in\{\mathcal{S},\mathcal{O},\mathcal{B},\tilde{\mathcal{S}}\}$ we denote $\mathbf{i}$ as the trajectory on the information set. This is done to be able to use any kind of policy class considered in the main sections as well. For a generic single trajectory objective defined with $\mathcal{J}\in\{\mathcal{J}_{1,\mathcal{S}},\mathcal{J}_{1,\mathcal{O}},\tilde{\mathcal{J}}_{1,\mathcal{S}}\}$ it is possible to write:

$$\begin{aligned} \nabla_\theta\mathcal{J}(\pi) &= \nabla_\theta\mathop{\mathbb{E}}_{\mathbf{h}\sim p^\pi}[\mathcal{F}(d(\cdot|\mathbf{h}))] \\ &= \nabla_\theta\sum_{\mathbf{h}}p^\pi(\mathbf{h})\mathcal{F}(d(\cdot|\mathbf{h})) \\ &= \sum_{\mathbf{h}}\Big(\nabla_\theta p^\pi(\mathbf{h})\Big)\mathcal{F}(d(\cdot|\mathbf{h})) \end{aligned}$$

Thanks to the usual log-trick

$$= \sum_{\mathbf{h}} p^{\pi}(\mathbf{h})\Big(\nabla_\theta \log p^{\pi}(\mathbf{h})\Big)\mathcal{F}(d(\cdot|\mathbf{h}))$$
$$= \mathop{\mathbb{E}}_{\mathbf{h}\sim p^{\pi}} \big[\nabla_\theta \log p^{\pi}(\mathbf{h})\mathcal{F}(d(\cdot|\mathbf{h}))\big]$$

The computation of the gradient is then reconducted to the calculation of the log-policy term $\nabla_\theta \log p^{\pi}(\mathbf{h})$ for the generic class $\pi \in \Pi_{\mathcal{I}}$. It follows that

$$\begin{aligned}\nabla_\theta \log p^{\pi}(\mathbf{h}) &= \nabla_\theta \log \Big(\mu(s_0) \prod_{t\in[T]} \mathcal{O}(o_t|s_t)\pi(a_t|i_t)\mathbb{P}(s_{t+1}|s_t,a_t)\mathcal{T}^{o_t a_t}(b_{t+1}|b_t)\Big)\\ &= \nabla_\theta \Big(\log(\mu(s_0)) + \sum_{t\in[T]} \log(\mathcal{O}(o_t|s_t)) + \log(\pi(a_t|i_t)) + \log(\mathbb{P}(s_{t+1}|s_t,a_t)) + \log(\mathcal{T}^{o_t a_t}(b_{t+1}|b_t))\Big)\end{aligned}$$

Where the only terms depending on θ are indeed the $\mathcal{I}$-specific log-policy terms, leading to

$$\nabla_\theta \log p^{\pi}(\mathbf{h}) = \sum_{t\in[T]} \nabla_\theta \log \pi_\theta(a_t|i_t)$$

which leads to the standard REINFORCE-like formulation of policy gradients. □

Proofs of Lipschitz Constants Computation

Theorem 4.2.2 (Local Lipschitz Constants). *Let $\pi_1, \pi_2 \in \Pi_{\mathcal{I}}$, let $\mathcal{T}_{\mathcal{I}}(\pi_1,\pi_2) = \{\mathbf{i} \in \mathcal{T}_{\mathcal{I}} : p^{\pi_1}(\mathbf{i}) > 0 \vee p^{\pi_2}(\mathbf{i}) > 0\}$ be the set of realizable trajectories over $\mathcal{I} \in \{\mathcal{S}, \mathcal{O}\}$, and let $\mathbf{i}^\star = \arg\max_{\mathbf{i}\in\mathcal{T}_{\mathcal{I}}(\pi_1,\pi_2)} \mathcal{F}(d_{\mathcal{I}}(\cdot|\mathbf{i}))$. It holds*

$$|\mathcal{J}_{1,\mathcal{I}}(\pi_1) - \mathcal{J}_{1,\mathcal{I}}(\pi_2)| \leqslant T\mathcal{F}(d_{\mathcal{I}}(\cdot|\mathbf{i}))d^{TV}(\pi_1,\pi_2).$$

Proof. Let us define the set of reachable trajectories in T steps by following a generic policy π_i as $T_i = \{\mathbf{h} \in T_i : p^{\pi_i}(\mathbf{h}) > 0\}$, it follows that for both MSE and MOE objective, by defining $\mathbf{h}$ as $\mathbf{s}$ or $\mathbf{o}$ respectively, we can see that

$$\begin{aligned}|\mathcal{J}(\pi_1) - \mathcal{J}(\pi_2)| &= \Big|\mathop{\mathbb{E}}_{\mathbf{h}\sim p^{\pi_1}}[\mathcal{F}(d(\cdot|\mathbf{h}))] - \mathop{\mathbb{E}}_{\mathbf{h}\sim p^{\pi_2}}[\mathcal{F}(d(\cdot|\mathbf{h}))]\Big|\\ &\leqslant \sum_{\mathbf{h}\in T_1 \cup T_2} \mathcal{F}(d(\cdot|\mathbf{h}))\Big|p^{\pi_1}(\mathbf{h}) - p^{\pi_2}(\mathbf{h})\Big|\end{aligned}$$

By defining $\mathbf{h}^\star \in \arg\max_{\mathbf{h}\in T_1\cup T_2} \mathcal{F}[d(\cdot|\mathbf{h})]$

$$\leqslant \mathcal{F}[d(\cdot|\mathbf{h}^\star)] \sum_{\mathbf{h}\in T_1\cup T_2} \Big|p^{\pi_1}(\mathbf{h}) - p^{\pi_2}(\mathbf{h})\Big|$$

We notice that $p^{\pi_i} = \prod_t^{\pi} p_t^{\pi_i}$ and that the total variation between two product distributions can be upper-bounder by the summation over the per-step total variations, namely $d^{\text{TV}}(\prod_t^{\pi} p_t^{\pi_i}, \prod_t^{\pi} p_t^{\pi_j}) \leqslant \sum_{t\in[T]} d^{\text{TV}}(p_t^{\pi_i}, p_t^{\pi_j})$, leading to

$$\begin{aligned}&= \mathcal{F}[d(\cdot|\mathbf{h}^\star)]d^{\text{TV}}(p^{\pi_1}, p^{\pi_2})\\ &\leqslant \mathcal{F}[d(\cdot|\mathbf{h}^\star)] \sum_{t\in[T]} d^{\text{TV}}(p_t^{\pi_1}, p_t^{\pi_2})\end{aligned}$$

The only difference between the two distributions (for a fixed step) consists of the policies

$$= T\mathcal{F}[d(\cdot|\mathbf{h}^\star)]d^{\text{TV}}(\pi^1,\pi^2) = \mathcal{L}(\pi_1,\pi_2)d^{\text{TV}}(\pi^1,\pi^2)$$

It follows a (bound on a) Lipschitz constant dependent on the two policies to be compared that is directly proportional to the best single trajectory (in terms of entropy) reachable by the policies themselves. Any policy able to generate a maximum entropic trajectory will have the highest possible Lipschitz constant. The constant then gets steeper as the quality of the policies improves.

□

Theorem 4.2.3. *For a policy $\pi_\theta \in \Pi_{\mathcal{I}}$ parametrized by $\theta \in \Theta \subseteq \mathbb{R}^{SA}$, we have*

$$\nabla_\theta \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_\theta) = \mathop{\mathbb{E}}_{\boldsymbol{b}\sim p^\pi_{\mathcal{B}}} \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim p(\cdot|\boldsymbol{b})}\Big[\nabla_\theta \log \pi_\theta(\tilde{\mathbf{s}})\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))\Big], \tag{4.34}$$

where $\nabla_\theta \log \pi_\theta(\tilde{\mathbf{s}})$ are defined as in 4.2.1. Additionally, let $\mathcal{T}_{\mathcal{B}}(\pi_1, \pi_2) = \{\boldsymbol{b} \in \mathcal{T}_{\mathcal{B}} : p^{\pi_1}(\boldsymbol{b}) > 0 \vee p^{\pi_2}(\boldsymbol{b}) > 0\}$, $\boldsymbol{b}^\star = \arg\max_{\boldsymbol{b}\in\mathcal{T}_{\mathcal{B}}(\pi_1,\pi_2)} \mathbb{E}_{\tilde{\mathbf{s}}\sim p(\cdot|\boldsymbol{b})} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$, and $\bar{\mathcal{F}}(\boldsymbol{b}^\star) = \mathbb{E}_{\tilde{\mathbf{s}}\sim p(\cdot|\boldsymbol{b})} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$, we have

$$|\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_1) - \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi_2)| \leqslant T\bar{\mathcal{F}}(\boldsymbol{b}^\star) d^{TV}(\pi_1, \pi_2). \tag{4.35}$$

Proof. Similarly to the previous steps,

$$\begin{aligned}
|\tilde{\mathcal{J}}(\pi_1) - \tilde{\mathcal{J}}(\pi_2)| &= \Big| \mathop{\mathbb{E}}_{\boldsymbol{b}\sim p^{\pi_1}} \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}(\cdot)} [\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))] - \mathop{\mathbb{E}}_{\boldsymbol{b}\sim p^{\pi_2}} \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}(\cdot)} [\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))] \Big| \\
&\leqslant \sum_{\boldsymbol{b}\in T_1\cup T_2} \sum_{\tilde{\mathbf{s}}} \boldsymbol{b}(\tilde{\mathbf{s}})\mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \Big| p^{\pi_1}(\boldsymbol{b}) - p^{\pi_2}(\boldsymbol{b}) \Big| \\
&= \sum_{\boldsymbol{b}\in T_1\cup T_2} \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \Big| p^{\pi_1}(\boldsymbol{b}) - p^{\pi_2}(\boldsymbol{b}) \Big|
\end{aligned}$$

Again let us define $\boldsymbol{b}^\star \in \arg\max_{\boldsymbol{b}\in T_1\cup T_2} \mathbb{E}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))$

$$\begin{aligned}
&\leqslant \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}^\star} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) d^{\text{TV}}(p^{\pi_1}, p^{\pi_2}) \\
&\leqslant \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}^\star} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \sum_{t\in[T]} d^{\text{TV}}(p^{\pi_1}_t, p^{\pi_2}_t) \\
&= T \mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim \boldsymbol{b}^\star} \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) d^{\text{TV}}(\pi^1, \pi^2) = \tilde{\mathcal{L}}(\pi_1, \pi_2) d^{\text{TV}}(\pi^1, \pi^2)
\end{aligned}$$

Again, the (local) Lipschitz constant $\tilde{\mathcal{L}}(\pi_1, \pi_2)$ is dependent on the maximum (expected) entropy that can be induced by one of the policies. One may notice that $\mathcal{L}(\pi_1, \pi_2)$ will be usually higher than $\tilde{\mathcal{L}}(\pi_1, \pi_2)$.

□

Proofs of Proxy Gaps

Theorem 4.2.4 (Proxy Gaps). *For a fixed policy $\pi \in \Pi_{\mathcal{I}}$, the MSE objective $\mathcal{J}_{1,\mathcal{S}}(\pi)$ is bounded by the MOE objective according to*

$$\begin{aligned}
\mathcal{J}_{1,\mathcal{S}}(\pi) &\leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim \bar{p}_{\mathcal{S}}} \Big[\frac{1}{\mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})} \mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) \Big] \\
\mathcal{J}_{1,\mathcal{S}}(\pi) &\geqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim \bar{p}_{\mathcal{S}}} \Big[\frac{1}{1 - \mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})} \mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) \Big] - \mathop{\mathbb{E}}_{\mathbf{s}\sim \bar{p}_{\mathcal{S}}} \Big[\frac{\mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})}{1 - \mathbb{P}(\mathcal{T}_{\mathcal{O}}|\mathbf{s})} \Big] \log O
\end{aligned}$$

Analogously, $\mathcal{J}_{1,\mathcal{S}}(\pi)$ is bounded by the MBE objective according to

$$\begin{aligned}
\mathcal{J}_{1,\mathcal{S}}(\pi) &\leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim \bar{p}} \Big[\frac{1}{\bar{p}_{\mathcal{S}}(\mathbf{s})} \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s}) \Big] \\
\mathcal{J}_{1,\mathcal{S}}(\pi) &\geqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim \bar{p}} \Big[\frac{1}{1 - \bar{p}_{\mathcal{S}}(\mathbf{s})} \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s}) \Big] - \mathop{\mathbb{E}}_{\mathbf{s}\sim \bar{p}} \Big[\frac{\bar{p}_{\mathcal{S}}(\mathbf{s})}{1 - \bar{p}_{\mathcal{S}}(\mathbf{s})} \Big] \log S
\end{aligned}$$

Proof. **MOE**: Let us define the set of observation-trajectories that have an entropy higher than the entropy of a fixed trajectory over true states, namely $\mathbf{h}_{\mathcal{O}}(\mathbf{s}) = \{\mathbf{o} \in \mathbf{h}_{\mathcal{O}} : \mathcal{F}(d(\cdot|\mathbf{o})) \geqslant \mathcal{F}(d(\cdot|\mathbf{s}))\}$. It follows that by employing the conditional trajectory probability $p^\pi(\mathbf{o}|\mathbf{s})$ one can define the probability $\mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s}) = \sum_{\mathbf{o}\in\mathbf{h}_{\mathcal{O}}(\mathbf{s})} p^\pi(\mathbf{o}|\mathbf{s})$. It follows that

$$\begin{aligned}
\mathcal{J}_{1,\mathcal{S}} - \mathcal{J}_{1,\mathcal{O}} &= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^\pi(\cdot)} \Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) \Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^\pi(\cdot)} \Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathop{\mathbb{E}}_{\mathbf{o}\sim p^\pi(\cdot|\mathbf{s})} \mathcal{F}(d(\cdot|\mathbf{o})) \Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^\pi(\cdot)} \Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \sum_{\mathbf{o}} p^\pi(\mathbf{o}|\mathbf{s}) \mathcal{F}(d(\cdot|\mathbf{o})) \Big]
\end{aligned}$$

By definition $\mathcal{F}(d(\mathbf{o} \in \mathbf{h}_{\mathcal{O}}(\mathbf{s}))) \geqslant \mathcal{F}(d(\cdot|\mathbf{s}))$, and by positivity of the entropy function $\mathcal{F}(d(\mathbf{o} \notin \mathbf{h}_{\mathcal{O}}(\mathbf{s}))) \geqslant 0$

$$\begin{aligned}
&\leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})\mathcal{F}(d(\cdot|\mathbf{s}))\Big] \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[(1 - \mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s}))\mathcal{F}(d(\cdot|\mathbf{s}))\Big]
\end{aligned}$$

It follows that

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\frac{1}{\mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})}\mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s})\Big]$$

In the same way, focusing on the terms inside the outer expectation for simplicity, one obtains:

$$\begin{aligned}
\mathcal{J}_{1,\mathcal{S}} - \mathcal{J}_{1,\mathcal{O}} &= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s})\Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathop{\mathbb{E}}_{\mathbf{o}\sim p^{\pi}(\cdot|\mathbf{s})}\mathcal{F}(d(\cdot|\mathbf{o}))\Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \sum_{\mathbf{o}} p^{\pi}(\mathbf{o}|\mathbf{s})\mathcal{F}(d(\cdot|\mathbf{o}))\Big]
\end{aligned}$$

Again, one notices that $\mathcal{F}(d(\cdot|\mathbf{o} \in \mathbf{h}_{\mathcal{O}}(\mathbf{s})))) \leqslant \mathcal{F}(d(\cdot|\mathbf{s}))$ and $\mathcal{F}(d(\cdot|\mathbf{o} \in \mathbf{h}_{\mathcal{O}}(\mathbf{s})))) \leqslant \log(O)$, from which the inner expectation turns out to be bounded by the use of the complementary probability $\mathbb{P}(\mathbf{h}_{\mathcal{O}}^{C}|\mathbf{s}) = \sum_{\mathbf{o}\notin\mathbf{h}_{\mathcal{O}}(\mathbf{s})} p^{\pi}(\mathbf{o}|\mathbf{s})$

$$\begin{aligned}
&\geqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[(1 - \mathbb{P}(\mathbf{h}_{\mathcal{O}}^{C}|\mathbf{s}))\mathcal{F}(d(\cdot|\mathbf{s})) - \mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})\log(O)\Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})\mathcal{F}(d(\cdot|\mathbf{s})) - \mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})\log(O)\Big]
\end{aligned}$$

Leading to

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \geqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\frac{\mathcal{J}_{1,\mathcal{O}}(\pi|\mathbf{s}) - \mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})\log(O)}{1 - \mathbb{P}(\mathbf{h}_{\mathcal{O}}|\mathbf{s})}\Big]$$

MBE: Let us define the similar set for hallucinated trajectories $\mathbf{h}(\mathbf{s}) = \{\tilde{\mathbf{s}} \in \mathbf{h}_{\tilde{\mathcal{S}}} : \mathcal{F}(d(\cdot|\tilde{\mathbf{s}})) \geqslant \mathcal{F}(d(\cdot|\mathbf{s}))\}, \mathbb{P}(\mathbf{h}|\boldsymbol{b}) = \sum_{\mathbf{s}\in\mathbf{h}(\mathbf{s})} \boldsymbol{b}(\mathbf{s})$.

$$\begin{aligned}
\mathcal{J}_{1,\mathcal{S}}(\pi) - \tilde{\mathcal{J}}(\pi) &= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s})\Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathop{\mathbb{E}}_{\mathbf{oa},\boldsymbol{b}\sim p^{\pi}(\cdot|\mathbf{s})}\mathop{\mathbb{E}}_{\tilde{\mathbf{s}}\sim\boldsymbol{b}}\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))\Big] \\
&= \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathop{\mathbb{E}}_{\mathbf{oa},\boldsymbol{b}\sim p(\cdot|\mathbf{s})}\sum_{\tilde{\mathbf{s}}}\boldsymbol{b}(\tilde{\mathbf{s}})\mathcal{F}(d(\cdot|\tilde{\mathbf{s}}))\Big]
\end{aligned}$$

Again $\mathcal{F}(d(\cdot|\tilde{\mathbf{s}} \in \mathbf{h}_{\mathcal{S}}(\mathbf{s}))) \geqslant \mathcal{F}(d(\cdot|\mathbf{s}))$ and $\mathcal{F}(d(\cdot|\tilde{\mathbf{s}} \notin \mathbf{h}_{\mathcal{S}}(\mathbf{s}))) \geqslant 0$

$$\begin{aligned}
&\leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\mathcal{F}(d(\cdot|\mathbf{s})) - \mathop{\mathbb{E}}_{\mathbf{oa},\boldsymbol{b}\sim p(\cdot|\mathbf{s})}[\mathbb{P}(\mathbf{h}|\boldsymbol{b})]\mathcal{F}(d(\cdot|\mathbf{s}))\Big] \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[(1 - \mathop{\mathbb{E}}_{\mathbf{oa},\boldsymbol{b}\sim p(\cdot|\mathbf{s})}\mathbb{P}(\mathbf{h}|\boldsymbol{b}))\mathcal{F}(d(\cdot|\mathbf{s}))\Big]
\end{aligned}$$

We call $\bar{p}_{\mathcal{S}}(\mathbf{s}) = \mathbb{E}_{\boldsymbol{b}\sim p^{\pi}(\cdot|\mathbf{s})}\mathbb{P}(\mathbf{h}|\boldsymbol{b})$, it follows that

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \leqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\frac{1}{\bar{p}(\mathbf{s})}\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s})\Big]$$

In the same way as before, by simply changing the definitions accordingly, one obtains that:

$$\mathcal{J}_{1,\mathcal{S}}(\pi) \geqslant \mathop{\mathbb{E}}_{\mathbf{s}\sim p^{\pi}(\cdot)}\Big[\frac{\tilde{\mathcal{J}}_{1,\mathcal{S}}(\pi|\mathbf{s}) - \bar{p}(\mathbf{s})\log S}{1 - \bar{p}(\mathbf{s})}\Big]$$

□

B.2 Proofs of Chapter 5

Theorem 5.1.1. *Let d^{π} be the (categorical) distribution induced by π over the finite set $\mathcal{S}$ with $|\mathcal{S}| = S$, and let d_K be the empirical distribution obtained from K independent samples drawn from d^{π}. Then, for any $\epsilon > 0$, the following bound holds:*

$$\mathbb{P}\left(\mathcal{H}(d_\pi) - \mathcal{H}(d_K) > \epsilon\right) \leqslant 2S \exp\left(-K \cdot \frac{\epsilon^2 \mathrm{Var}(d_\pi)}{2S^3\mathcal{H}^2(d_\pi)}\right),$$

where $\mathcal{H}(d_K)$ and $\mathcal{H}(d_\pi)$ denote the entropy of the empirical and true distributions, respectively, and $\mathrm{Var}(d_\pi) = \sum_{s\in[\mathcal{S}]} d_\pi(s)(1 - d_\pi(s))$ is the variance of a random variable associated with the categorical distribution d_π. Furthermore, to ensure this concentration with confidence $1-\delta$, the number of samples n must satisfy the following lower bound:

$$K \geqslant \frac{2S^3\mathcal{H}^2(d_\pi)}{\epsilon^2\mathrm{Var}(d_\pi)} \cdot \ln\frac{2S}{\delta}.$$

Proof. The proof consists of three main steps. In order to keep the derivation agnostic from the state or trajectory based setting, we will now introduce a different yet equivalent notation: let p be a categorical distribution over a finite set $\mathcal{X}$ with $|\mathcal{X}| = K$, and let $\hat{p}$ be the empirical distribution obtained from n.

Decomposing the Problem via Union Bound. First, we expand the entropy terms to highlight the contribution of the single components:

$$\mathbb{P}(\mathcal{H}(p) - \mathcal{H}(\hat{p}) > \epsilon) \leqslant \mathbb{P}\left(\sum_{i=1}^{K} p_i \log\left(\frac{1}{p_i}\right) - \hat{p}_i \log\left(\frac{1}{\hat{p}_i}\right) > \epsilon\right) = \mathbb{P}\left(\sum_{i=1}^{K} h(p_i) - h(\hat{p}_i) > \epsilon\right),$$

where $p_i = \mathbb{P}(X = x_i)$ and $h(x) = x\log\left(\frac{1}{x}\right)$.

Applying the union bound to the previous result, we get:

$$\mathbb{P}(\mathcal{H}(p) - \mathcal{H}(\hat{p}) > \epsilon) \leqslant \sum_{i=1}^{K} \mathbb{P}\left(h(p_i) - h(\hat{p}_i) > \frac{\epsilon}{K}\right). \tag{B.2}$$

Bounding the Entropy of the Components using a Linear Approximation. Now, we focus on finding an upper bound to $h(p_i) - h(\hat{p}_i)$. We introduce a lower bound to $h(\hat{p}_i)$ obtained by a combination of functions that are linear in the deviation $|p_i - \hat{p}_i|$:

$$h(\hat{p}_i) \leqslant h(p_i) - \frac{h(p_i)|p_i - \hat{p}_i|}{\max(p_i, 1-p_i)} \leqslant h(p_i) - \frac{h(p_i)|p_i - \hat{p}_i|}{p_i(1-p_i)}.$$

As a consequence

$$\mathbb{P}\left(h(p_i) - h(\hat{p}_i) > \epsilon\right) \leqslant \mathbb{P}\left(\frac{h(p_i)|p_i - \hat{p}_i|}{p_i(1-p_i)} > \epsilon\right) \leqslant \mathbb{P}\left(|p_i - \hat{p}_i| > \frac{p_i(1-p_i)}{h(p_i)}\epsilon\right). \tag{B.3}$$

Thanks to this last inequality, we can now focus on the concentration inequality of the Bernoulli distributions associated with the parameters p_i.

Applying a Concentration Inequality for Bernoulli Distributions. Finally, we use a concentration inequality on the estimation of a Bernoulli-distributed parameter to express this probability bound in terms of the variance of p_i ($\mathrm{Var}(p_i) = p_i(1-p_i)$).

Leveraging Chernoff bound for Bernoulli distributions, we get:

$$\mathbb{P}(|p_i - \hat{p}_i| > \epsilon) \leqslant e^{-nD_{D_{\mathrm{KL}}}(p_i+\epsilon||p_i)} + e^{-nD_{D_{\mathrm{KL}}}(p_i-\epsilon||p_i)} \leqslant 2e^{-\frac{n\epsilon^2}{2p_i(1-p_i)}} = 2e^{-\frac{n\epsilon^2}{2\mathrm{Var}(p_i)}}. \tag{B.4}$$

We now complete the proof by combining the results in Eqs (B.2), (B.3), and (B.4):

$$\mathbb{P}(\mathcal{H}(p)-\mathcal{H}(\hat{p}) > \epsilon) \leqslant \sum_{i=1}^{K} \mathbb{P}\left(h(p_i) - h(\hat{p}_i) > \frac{\epsilon}{K}\right) \leqslant \sum_{i=1}^{K} \mathbb{P}\left(|p_i - \hat{p}_i| > \frac{p_i(1-p_i)}{Kh(p_i)}\epsilon\right) \leqslant 2\sum_{i=1}^{K} e^{-\frac{n\epsilon^2 p_i(1-p_i)}{2K^2h^2(p_i)}}. \tag{B.5}$$

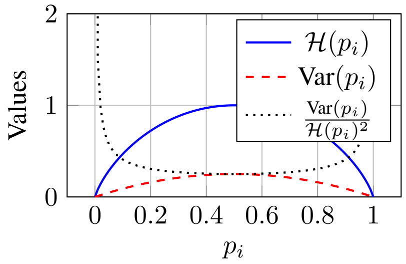


Figure B.1: *The plot shows that $\mathcal{H}(p_i)$ and $\mathrm{Var}(p_i)$ are concave symmetric function with their maximum located at $p_i = 0.5$, while $\frac{Var(p_i)}{\mathcal{H}(p_i)^2}$ is a convex symmetric function with its minimum located at $p_i =$* 0.5.

In order to remove the summation over the K components of the distribution, we need to find a lower bound to the term $\frac{p_i(1-p_i)}{\mathcal{H}^2(p_i)}$ that is independent of the specific component parameter p_i. Here, we show the chain of passages that achieve this goal:

$$\min_i \frac{p_i(1-p_i)}{h^2(p_i)} \geqslant \min_i \frac{p_i(1-p_i)}{\mathcal{H}^2(p_i)} = \frac{\max_i p_i(1-p_i)}{\max_i \mathcal{H}^2(p_i)} \geqslant \frac{\sum_i p_i(1-p_i)}{K \max_i \mathcal{H}^2(p_i)} \geqslant \frac{\mathrm{Var}(p)}{K\mathcal{H}^2(p)}.$$

The motivations for each step are:

1. $\mathcal{H}(p_i) \geqslant h(p_i)$.
2. The value of p_i that minimizes $\frac{p_i(1-p_i)}{\mathcal{H}^2(p_i)}$ is the one with the highest entropy (see FigureB.1). Since the higher the entropy $\mathcal{H}(p_i)$ the higher is also the variance $p_i(1-p_i)$, we can restate the minimization problem as the ratio of two maximization problems.
3. The term at the numerator is the maximum variance, which can be lower bounded by the average variance.
4. The maximum entropy among the Bernoulli distributions associated with all the components is upper bounded by the entropy of the categorical distribution p.

Leveraging this result in Eq. (B.5) concludes the proof.

□

Lemma 5.2.1 (Entropy Mismatch)**.** *For every cMG $\mathcal{M}^{\mathcal{H}}$ equipped with an entropy functional, for a fixed (joint) policy $\pi = (\pi^i)_{i\in\mathcal{N}}$ the infinite-trials objectives are ordered according to:*

$$\frac{\mathcal{H}(d^\pi)}{|\mathcal{N}|} \leqslant \frac{1}{|\mathcal{N}|}\sum_{i\in[|\mathcal{N}|]} \mathcal{H}(d_i^\pi) \leqslant \mathcal{H}(\tilde{d}^\pi)$$

$$\mathcal{H}(\tilde{d}^\pi) \leqslant \sup_{i\in[|\mathcal{N}|]} \mathcal{H}(d_i^\pi) + \log(|\mathcal{N}|) \leqslant \mathcal{H}(d^\pi) + \log(|\mathcal{N}|)$$

Proof. The bounds follow directly from simple yet fundamental relationships between entropies of Joint, marginal and mixture distributions which can be found in Paninski [2003], Kolchinsky and Tracey [2017], in particular:

$$\frac{1}{|\mathcal{N}|}\mathcal{H}(d^\pi) \leqslant \frac{1}{|\mathcal{N}|}\sum_{i\in[\mathcal{N}]} \mathcal{H}(d_i^\pi) \overset{(a)}{\leqslant} \mathcal{H}(\tilde{d}^\pi) \overset{(b)}{\leqslant} \frac{1}{|\mathcal{N}|}\sum_{i\in[\mathcal{N}]} \mathcal{H}(d_i^\pi) + \log(|\mathcal{N}|) \overset{(c)}{\leqslant} \sup_{i\in[\mathcal{N}]} \mathcal{H}(d_i^\pi) + \log(|\mathcal{N}|) \leqslant \mathcal{H}(d^\pi) + \log(|\mathcal{N}|)$$

where step (a) and (b) use the fact that $\tilde{d}^\pi(s) := \frac{1}{|\mathcal{N}|}\sum_{i\in[\mathcal{N}]} d_i^\pi(s)$ is a uniform mixture over the agents, whose distribution over the weights has entropy $\log(|\mathcal{N}|)$, so as we can apply the bounds from Kolchinsky and Tracey [2017]. Step (c) uses the fact that $\mathcal{H}(d^\pi) = \sum_{i\in[\mathcal{N}]} \mathcal{H}(d_i^\pi|d_{<i}^\pi)$, then taking the supremum as first i it follows that $\sup_{i\in[\mathcal{N}]} \mathcal{H}(d_i^\pi) = \mathcal{H}(d^\pi) - \sum_{j\in[\mathcal{N}]>i} \mathcal{H}(d_j^\pi|d_{<j}^\pi, d_i^\pi) \leqslant \mathcal{H}(d^\pi)$ due to non-negativity of entropy. □

Theorem 5.2.2 (Objectives Mismatch in cMGs)**.** *For every cMG $\mathcal{M}^{\mathcal{F}}$ equipped with a L-Lipschitz function $\mathcal{F}$ (see Ass. 2.3.1), let $K \in \mathbb{N}^+$ be a number of evaluation episodes/trials, and let $\delta \in (0,1]$ be a confidence level, then for any (joint) policy $\pi = (\pi^i \in \Pi^i)_{i\in[|\mathcal{N}|]}$, it holds that*

$$|\zeta_K(\pi) - \zeta_\infty(\pi)| \leqslant LT\sqrt{\frac{2|\mathcal{S}|\log(2T/\delta)}{K}},$$

$$|\zeta_K^i(\pi) - \zeta_\infty^i(\pi)| \leqslant LT\sqrt{\frac{2|\tilde{\mathcal{S}}|\log(2T/\delta)}{K}},$$

$$|\tilde{\zeta}_K(\pi) - \tilde{\zeta}_\infty(\pi)| \leqslant LT\sqrt{\frac{2|\tilde{\mathcal{S}}|\log(2T/\delta)}{|\mathcal{N}|K}}.$$

Proof. For the general proof structure, we adapt the steps of Mutti et al. [2022a] for Convex MDPs to the different objectives possible in cMGs. Let us start by considering Joint objectives, then:

$$\begin{aligned}\left|\zeta_K(\pi) - \zeta_\infty(\pi)\right| &= \left|\mathop{\mathbb{E}}_{d_K\sim p_K^\pi}[\mathcal{F}(d_K)] - \mathcal{F}(d^\pi)\right| \leqslant \mathop{\mathbb{E}}_{d_K\sim p_K^\pi}[|\mathcal{F}(d_K) - \mathcal{F}(d^\pi)|]\\ &\overset{(a)}{\leqslant} \mathop{\mathbb{E}}_{d_K\sim p_K^\pi}\left[L\left\|d_K - d^\pi\right\|_1\right] \leqslant L\mathop{\mathbb{E}}_{d_K\sim p_K^\pi}\left[\left\|d_K - d^\pi\right\|_1\right]\\ &\overset{(b)}{\leqslant} L\mathop{\mathbb{E}}_{d_K\sim p_K^\pi}\left[\max_{t\in[T]}\left\|d_{K,t} - d_t^\pi\right\|_1\right],\end{aligned}$$

where in step (a) we use the assumption of $\mathcal{F}$ being Lipschitz to write and in step (b) we apply a maximization over $t \in [T]$ since $d_K = \frac{1}{T}\sum_{t\in[T]} d_{K,t}$ and $d^\pi = \frac{1}{T}\sum_{t\in[T]} d_t^\pi$. We then apply bounds in high probability

$$\begin{aligned}Pr\Big(\max_{t\in[T]}\left\|d_{K,t} - d_t^\pi\right\|_1 \geqslant \epsilon\Big) &\leqslant Pr\Big(\bigcup_t \left\|d_{K,t} - d_t^\pi\right\|_1 \geqslant \epsilon\Big)\\ &\overset{(c)}{\leqslant} \sum_t Pr\Big(\left\|d_{K,t} - d_t^\pi\right\|_1 \geqslant \epsilon\Big)\\ &\leqslant T\, Pr\Big(\left\|d_{K,t} - d_t^\pi\right\|_1 \geqslant \epsilon\Big),\end{aligned}$$

with $\epsilon > 0$ and in step (c) we applied a union bound. We then consider standard concentration inequalities for empirical distributions [Weissman et al., 2003] so to obtain the final bound

$$Pr\left(\left\|d_{K,t} - d_t^\pi\right\|_1 \geqslant \sqrt{\frac{2|\mathcal{S}|\log(2/\delta')}{K}}\right) \leqslant \delta'. \tag{B.6}$$

By setting $\delta' = \delta/T$, and then plugging the empirical concentration inequality, we have that with probability at least $1-\delta$

$$\left|\zeta_K(\pi) - \zeta_\infty(\pi)\right| \leqslant LT\sqrt{\frac{2|\mathcal{S}|\log(2T/\delta)}{K}},$$

which concludes the proof for Joint objectives.

The proof for disjoint objectives follows the same rationale by bounding each per-agent term separately and after noticing that due to Assumption 5.1.1, the resulting bounds get simplified in the overall averaging. As for mixture objectives, the only core difference is after step (b), where $\tilde{d}_K$ takes the place of d_K and $\tilde{d^\pi}$ of d^π. The remaining steps follow the same logic, out of noticing that the empirical distribution with respect to $\tilde{d^\pi}$ is taken with respect $|\mathcal{N}|K$ samples in total. Both the two bounds then take into account that the support of the empirical distributions have size $|\tilde{\mathcal{S}}|$ and not $|\mathcal{S}|$. □

Policy Gradient in cMGs with Infinite-Trials.

In this Section, we analyze policy search for the infinite-trials Joint problem ζ_∞ of Eq. (5.1), via projected gradient ascent over parametrized policies, providing in Th. B.2.3 the formal counterpart of Fact 5.2.1. As a side note, all of the following results hold for the (infinite-trials) mixture objective $\tilde{\zeta}_\infty$ of Eq. (5.6).

We will consider the class of parametrized policies with parameters $\theta_i \in \Theta_i \subset \mathbb{R}^d$, with the Joint policy then defined as $\pi_\theta, \theta \in \Theta = \times_{i\in[\mathcal{N}]}\Theta_i$. Additionally, we will focus on the computational complexity only, by assuming access to the exact gradient. The study of statistical complexity surpasses the scope of the current work. We define the **(independent) Policy Gradient Ascent** (PGA) update as:

$$\theta_i^{k+1} = \mathop{\arg\max}_{\theta_i\in\Theta_i} \zeta_\infty(\pi_{\theta^k}) + \left\langle \nabla_{\theta_i}\zeta_\infty(\pi_{\theta^k}), \theta_i - \theta_i^k\right\rangle - \frac{1}{2\eta}\|\theta_i - \theta_i^k\|^2 = \Pi_{\Theta_i}\{\theta_i^k + \eta\nabla_{\theta_i}\zeta_\infty(\pi_{\theta^k})\} \tag{B.7}$$

where $\Pi_{\Theta_i}\{\cdot\}$ denotes Euclidean projection onto Θ_i, and equivalence holds by the convexity of Θ_i. The classes of policies that allow for this condition to be true will be discussed shortly.

In general the overall proof is built of three main steps, shared with the theory of Potential Markov Games [Leonardos et al., 2022]: (i) prove the existence of well behaved stationary points; (ii) prove that performing independent policy gradient is equivalent to perform Joint policy gradient; (iii) prove that the (joint) PGA update converges to the stationary points via single-agent like analysis.

In order to derive the subsequent convergence proof, we will make the following assumptions:

Assumption B.2.1. *Define the quantity $\lambda(\theta) := d^{\pi_\theta}$, then:*
(i). *$\lambda(\cdot)$ forms a bijection between Θ and $\lambda(\Theta)$, where Θ and $\lambda(\Theta)$ are closed and convex.*
(ii). *The Jacobian matrix $\nabla_\theta \lambda(\theta)$ is Lipschitz continuous in Θ.*
(iii). *Denote $g(\cdot) := \lambda^{-1}(\cdot)$ as the inverse mapping of $\lambda(\cdot)$. Then there exists $\ell_\theta > 0$ s.t. $\|g(\lambda) - g(\lambda')| \leqslant \ell_\theta \|\lambda - \lambda'\|$ for some norm $\|\cdot\|$ and for all $\lambda, \lambda' \in \lambda(\Theta)$.*

Assumption B.2.2. *There exists $L > 0$ such that the gradient $\nabla_\theta \zeta_\infty(\pi_\theta)$ is L-Lipschitz.*

Assumption B.2.3. *The agents have access to a gradient oracle $\mathcal{O}(\cdot)$ that returns $\nabla_{\theta_i} \zeta_\infty(\pi_\theta)$ for any deployed Joint policy π_θ.*

On the Validity of Assumption B.2.1. This set of assumptions enforces the objective $\zeta_\infty(\pi_\theta)$ to be well-behaved with respect to θ even if non-convex in general, and will allow for a rather strong result. Yet, the assumptions are known to be true for directly parametrized policies over the whole support of the distribution d^π [Zhang et al., 2020a], and as a result they implicitly require agents to employ policies conditioned over the full state-space $\mathcal{S}$. Fortunately enough, they also guarantee Θ to be convex.

Lemma B.2.1 (**(i)** Global optimality of stationary policies [Zhang et al., 2020a])**.** *Suppose Assumption B.2.1 holds, and $\mathcal{F}$ is a concave, and continuous function defined in an open neighborhood containing $\lambda(\Theta)$. Let θ^* be a first-order stationary point of problem* (5.1)*, i.e.,*

$$\exists u^* \in \hat{\partial}(\mathcal{F} \circ \lambda)(\theta^*), \quad s.t. \quad \langle u^*, \theta - \theta^* \rangle \leqslant 0 \qquad for \qquad \forall \theta \in \Theta. \tag{B.8}$$

Then θ^* is a globally optimal solution of problem (5.1)*.*

This result characterizes the optimality of stationary points for Eq. (5.1). Furthermore, we know from Leonardos et al. [2022] that stationary points of the objective are Nash Equilibria.

Lemma B.2.2 (**(ii)** Projection Operator [Leonardos et al., 2022])**.** *Let $\theta := (\theta_1, ..., \theta_\mathcal{N})$ be the parameter profile for all agents and use the update of Eq.* (B.7) *over a non-disjoint infinite-trials objective. Then, it holds that*

$$\Pi_\Theta\{\theta^k + \eta \nabla_\theta \zeta_\infty(\pi_{\theta^k})\} = \Big(\Pi_{\Theta_i}\{\theta_i^k + \eta \nabla_{\theta_i} \zeta_\infty(\pi_{\theta^k})\}\Big)_{i \in [\mathcal{N}]}$$

This result will only be used for the sake of the convergence analysis, since it allows to analyze independent updates as Joint updates over a single objective. The following Theorem is the formal counterpart of Fact 5.2.1 and it is a direct adaptation to the multi-agent case of the single-agent proof by Zhang et al. [2020a], by exploiting the previous result.

Theorem B.2.3 (**(iii)** Convergence rate of independent PGA to stationary points (Formal Fact 5.2.1))**.** *Let Assumptions B.2.1 and B.2.2 hold. Denote $D_\lambda := \max_{\lambda, \lambda' \in \lambda(\Theta)} \|\lambda - \lambda'\|$ as defined in Assumption B.2.1(iii). Then the independent policy gradient update* (B.7) *with $\eta = 1/L$ satisfies for all k with respect to a stationary (joint) policy $\pi_{\theta*}$ the following*

$$\zeta_\infty(\pi_{\theta*}) - \zeta_\infty(\pi_{\theta^k}) \leqslant \frac{4L\ell_\theta^2 D_\lambda^2}{k+1}.$$

Proof. First, the Lipschitz continuity in Assumption B.2.2 indicates that

$$\Big|\zeta_\infty(\lambda(\theta)) - \zeta_\infty(\lambda(\theta^k)) - \langle \nabla_\theta \zeta_\infty(\lambda(\theta^k)), \theta - \theta^k \rangle\Big| \leqslant \frac{L}{2}\|\theta - \theta^k\|^2.$$

Consequently, for any $\theta \in \Theta$ we have the ascent property:

$$\zeta_\infty(\lambda(\theta)) \geqslant \zeta_\infty(\lambda(\theta^k)) + \langle \nabla_\theta \zeta_\infty(\lambda(\theta^k)), \theta - \theta^k \rangle - \frac{L}{2}\|\theta - \theta^k\|^2 \geqslant \zeta_\infty(\lambda(\theta)) - L\|\theta - \theta^k\|^2. \tag{B.9}$$

The optimality condition in the policy update rule (B.7) coupled with the result of Lemma B.2.2 allows us to follow the same rationale as Zhang et al. [2020a]. We will report their proof structure after this step for completeness.

$$\begin{aligned}
\zeta_\infty(\lambda(\theta^{k+1})) &\geqslant \zeta_\infty(\lambda(\theta^k)) + \langle \nabla_\theta \zeta_\infty(\lambda(\theta^k)), \theta^{k+1} - \theta^k \rangle - \frac{L}{2}\|\theta^{k+1} - \theta^k\|^2 \\
&= \max_{\theta\in\Theta} \zeta_\infty(\lambda(\theta^k)) + \langle \nabla_\theta \zeta_\infty(\lambda(\theta^k)), \theta - \theta^k \rangle - \frac{L}{2}\|\theta - \theta^k\|^2 \\
&\overset{(a)}{\geqslant} \max_{\theta\in\Theta} \zeta_\infty(\lambda(\theta)) - L\|\theta - \theta^k\|^2 \\
&\overset{(b)}{\geqslant} \max_{\alpha\in[0,1]} \left\{ \zeta_\infty(\lambda(\theta_\alpha)) - L\|\theta_\alpha - \theta^k\|^2 : \theta_\alpha = g(\alpha\lambda(\theta^*) + (1-\alpha)\lambda(\theta^k)) \right\}. \qquad \text{(B.10)}
\end{aligned}$$

where step (a) follows from (B.9) and step (b) uses the convexity of $\lambda(\Theta)$. Then, by the concavity of ζ_∞ and the fact that the composition $\lambda \circ g = id$ due to Assumption B.2.1(i), we have that:

$$\zeta_\infty(\lambda(\theta_\alpha)) = \zeta_\infty(\alpha\lambda(\theta^*) + (1-\alpha)\lambda(\theta^k)) \geqslant \alpha\zeta_\infty(\lambda(\theta^*)) + (1-\alpha)\zeta_\infty(\lambda(\theta^k)).$$

Moreover, due to Assumption B.2.1(iii) we have that:

$$\begin{aligned}
\|\theta_\alpha - \theta^k\|^2 &= \|g(\alpha\lambda(\theta^*) + (1-\alpha)\lambda(\theta^k)) - g(\lambda(\theta^k))\|^2 \qquad \text{(B.11)} \\
&\leqslant \alpha^2 \ell_\theta^2 \|\lambda(\theta^*) - \lambda(\theta^k)\|^2 \\
&\leqslant \alpha^2 \ell_\theta^2 D_\lambda^2.
\end{aligned}$$

From which we get

$$\begin{aligned}
&\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^{k+1})) \\
&\quad \leqslant \min_{\alpha\in[0,1]} \left\{ \zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta_\alpha)) + L\|\theta_\alpha - \theta^k\|^2 : \theta_\alpha = g(\alpha\lambda(\theta^*) + (1-\alpha)\lambda(\theta^k)) \right\} \\
&\quad \leqslant \min_{\alpha\in[0,1]} (1-\alpha)\big(\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^k))\big) + \alpha^2 L \ell_\theta^2 D_\lambda^2 . \qquad \text{(B.12)}
\end{aligned}$$

We define $\Lambda(\pi_\theta) := \lambda(\theta)$, then $\alpha_k = \frac{\zeta_\infty(\Lambda(\pi^*)) - \zeta_\infty(\Lambda(\pi^k))}{2L\ell_\theta^2 D_\lambda^2} \geqslant 0$, which is the minimizer of the RHS of (B.12) as long as it satisfies $\alpha_k \leqslant 1$. Now, we claim the following: If $\alpha_k \geqslant 1$ then $\alpha_{k+1} < 1$. Further, if $\alpha_k < 1$ then $\alpha_{k+1} \leqslant \alpha_k$. The two claims together mean that $(\alpha_k)_k$ is decreasing and all α_k are in $[0, 1)$ except perhaps α_0.

To prove the first of the two claims, assume $\alpha_k \geqslant 1$. This implies that $\zeta_\infty(\Lambda(\pi^*)) - \zeta_\infty(\Lambda(\pi^k)) \geqslant 2L\ell_\theta^2 D_\lambda^2$. Hence, choosing $\alpha = 1$ in (B.12), we get

$$\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^k)) \leqslant L\ell_\theta^2 D_\lambda^2$$

which implies that $\alpha_{k+1} \leqslant 1/2 < 1$. To prove the second claim, we plug α_k into (B.12) to get

$$\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^{k+1})) \leqslant \left(1 - \frac{\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^k))}{4L\ell_\theta^2 D_\lambda^2}\right)(\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^k))),$$

which shows that $\alpha_{k+1} \leqslant \alpha_k$ as required.

Now, by our preceding discussion, for $k = 1, 2, \ldots$ the previous recursion holds. Using the definition of α_k, we rewrite this in the equivalent form

$$\frac{\alpha_{k+1}}{2} \leqslant \left(1 - \frac{\alpha_k}{2}\right) \cdot \frac{\alpha_k}{2}.$$

By rearranging the preceding expressions and algebraic manipulations, we obtain

$$\frac{2}{\alpha_{k+1}} \geqslant \frac{1}{\left(1 - \frac{\alpha_k}{2}\right) \cdot \frac{\alpha_k}{2}} = \frac{2}{\alpha_k} + \frac{1}{1 - \frac{\alpha_k}{2}} \geqslant \frac{2}{\alpha_k} + 1.$$

For simplicity assume that $\alpha_0 < 1$ also holds. Then, $\frac{2}{\alpha_k} \geqslant \frac{2}{\alpha_0} + k$, and consequenlty

$$\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^k)) \leqslant \frac{\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^0))}{1 + \frac{\zeta_\infty(\lambda(\theta^*)) - \zeta_\infty(\lambda(\theta^0))}{4L\ell_\theta^2 D_\lambda^2} \cdot k} \leqslant \frac{4L\ell_\theta^2 D_\lambda^2}{k}.$$

A similar analysis holds when $\alpha_0 > 1$. Combining these two gives that $\zeta_\infty(\lambda(\pi^*)) - \zeta_\infty(\lambda(\pi^k)) \leqslant \frac{4L\ell_\theta^2 D_\lambda^2}{k+1}$ no matter the value of α_0, which proves the result. □

Parallel MaxEnt in cPMDPs with Infinite-Trials.

In this section, we provide in Th. B.2.4 the formal counterpart of Fact 5.2.2, which states that by performing MaxEnt-like updates in a parallel fashion in cPMDPs over *infinite-trials* objectives as reported in Algorithm B.2, it is possible to obtain *faster convergence rates* with respect to the non-parallel formulation, through a convenient scaling factor of $1/N$ with N being the number of parallel instantiations.

In order to allow for a simpler derivation, we will assume access to two kinds of oracles. First, some **approximate planning oracles** (one per each agent) that given a reward function (on states) $r : S \to \mathbb{R}$ and a sub-optimality gap ϵ_1, returns a policy $\pi =$ APPROXPLAN(r, ϵ_1) with the guarantee that $\mathcal{H}(\pi) \geqslant \max_{\bar{\pi}} \mathcal{H}(\bar{\pi}) - \epsilon_1$.

In addition, some **state distribution estimate oracles** (one per each agent) that estimate the state distribution $\hat{d} =$ DENSITYEST(π, ϵ_0) of any given (non-stationary) policy π, guaranteeing that $\|d^\pi - \hat{d}\|_\infty \leqslant \epsilon_1$.

Finally, we will assume that the entropy functional $\mathcal{H}$ is β-smooth, B-bounded, and that it satisfies the following inequality for all X, Y:

$$\|\nabla\mathcal{H}(X) - \nabla\mathcal{H}(Y)\| \leqslant \beta\|X - Y\|_\infty$$
$$-\beta\mathbb{I} \leqslant \nabla^2\mathcal{H}(X) \leqslant \beta\mathbb{I}; \ \|\nabla\mathcal{H}(X)\|_\infty \leqslant B$$

Under these assumptions, it follows that Algorithm B.2 enjoys the following:

Theorem B.2.4 (Convergence of Parallel MaxEnt). *For any $\varepsilon > 0$, set $\varepsilon_1 = 0.1\varepsilon$, $\varepsilon_0 = 0.1\beta^{-1}\varepsilon$, and $\eta = 0.1|\mathcal{S}|^{-1}\beta^{-1}N\varepsilon$, where Algorithm B.2 is run for T iterations over N agents in parallel where:*

$$T \geqslant 10\beta|\mathcal{S}|N^{-1}\varepsilon^{-1}\log 10B\varepsilon^{-1}\,,$$

we have that:

$$H(\pi_{mix,T}) \geqslant \max_\pi H(d_\pi) - \varepsilon\,.$$

Proof. Let π^* be a maximum-entropy policy, ie. $\pi^* \in \operatorname{argmax}_\pi H(d_\pi)$.

$$\begin{aligned}
H(d_{\pi_{\text{mix},t+1}}) &= H((1-\eta)d_{\pi_{\text{mix},t}} + \eta d_{\pi_{t+1}}) \\
&\geqslant H(d_{\pi_{\text{mix},t}}) + \eta\langle d_{\pi_{t+1}} - d_{\pi_{\text{mix},t}}, \nabla H(d_{\pi_{\text{mix},t}})\rangle - \eta^2\beta\|d_{\pi_{t+1}} - d_{\pi_{\text{mix},t}}\|_2^2 \\
&\geqslant H(d_{\pi_{\text{mix},t}}) + \frac{\eta}{N}\sum_i\langle d_{\pi^i_{t+1}} - d_{\pi_{\text{mix},t}}, \nabla H(d_{\pi_{\text{mix},t}})\rangle - \frac{\eta^2\beta}{N^2}\sum_i\|d_{\pi^i_{t+1}} - d_{\pi_{\text{mix},t}}\|_2^2
\end{aligned}$$

The second inequality follows from the smoothness of H, the third applies the definition of distributions induced by mixture policies. Now, to incorporate the error due to the two oracles, observe that for each agent it holds

$$\begin{aligned}
\langle d_{\pi^i_{t+1}}, \nabla H(d_{\pi_{\text{mix},t}})\rangle &\geqslant \langle d_{\pi^i_{t+1}}, \nabla H(\hat{d}^i_{\pi_{\text{mix},t}})\rangle - \beta\|d_{\pi_{\text{mix},t}} - \hat{d}^i_{\pi_{\text{mix},t}}\|_\infty \\
&\geqslant \langle d_{\pi^*}, \nabla H(\hat{d}^i_{\pi_{\text{mix},t}})\rangle - \beta\varepsilon_0 - \varepsilon_1 \\
&\geqslant \langle d_{\pi^*}, \nabla H(d_{\pi_{\text{mix},t}})\rangle - 2\beta\varepsilon_0 - \varepsilon_1
\end{aligned}$$

The first and last inequalities invoke the assumptions on the entropy functional. Note that the second inequality above follows from the defining character of the planning oracle. Using the above fact and continuing on

$$\begin{aligned}
H(d_{\pi_{\text{mix},t+1}}) &\geqslant H(d_{\pi_{\text{mix},t}}) + \frac{\eta}{N}\sum_i\langle d_{\pi^i_{t+1}} - d_{\pi_{\text{mix},t}}, \nabla H(d_{\pi_{\text{mix},t}})\rangle - \frac{\eta^2\beta}{N^2}\sum_i\|d_{\pi^i_{t+1}} - d_{\pi_{\text{mix},t}}\|_2^2 \\
&\geqslant H(d_{\pi_{\text{mix},t}}) + \eta\langle d_{\pi^\star} - d_{\pi_{\text{mix},t}}, \nabla H(d_{\pi_{\text{mix},t}})\rangle - 2\beta\eta\varepsilon_0 - \eta\varepsilon_1 - \frac{\eta^2\beta}{N}|\mathcal{S}| \\
&\geqslant (1-\eta)H(d_{\pi_{\text{mix},t}}) + \eta H(d_{\pi^*}) - 2\eta\beta\varepsilon_0 - \eta\varepsilon_1 - \frac{\eta^2\beta|\mathcal{S}|}{N}
\end{aligned}$$

The last step here utilizes the concavity of H. It follows that

$$H(d_{\pi^*}) - H(d_{\pi_{\text{mix},t+1}}) \leqslant (1-\eta)(H(d_{\pi^*}) - H(d_{\pi_{\text{mix},t}})) + 2\eta\beta\varepsilon_0 + \eta\varepsilon_1 + \frac{\eta^2\beta|\mathcal{S}|}{N}.$$

Telescoping the inequality, this simplifies to

$$
\begin{aligned}
H(d_{\pi *}) - H(d_{\pi_{\text{mix},T}}) &\leqslant (1-\eta)^T (H(d_{\pi *}) - H(d_{\pi_{\text{mix},0}})) + 2\beta\varepsilon_0 + \varepsilon_1 + \eta\beta \\
&\leqslant Be^{-T\eta} + 2\beta\varepsilon_0 + \varepsilon_1 + \frac{\eta^2 \beta |\mathcal{S}|}{N}.
\end{aligned}
$$

$$
\begin{aligned}
&H(d_{\pi *}) - H(d_{\pi_{\text{mix},T}}) \\
&\leqslant (1-\eta)^T (H(d_{\pi *}) - H(d_{\pi_{\text{mix},0}})) + 2\beta\varepsilon_0 + \varepsilon_1 + \frac{\eta\beta|\mathcal{S}|}{N} \\
&\leqslant Be^{-T\eta} + 2\beta\varepsilon_0 + \varepsilon_1 + \frac{\eta\beta|\mathcal{S}|}{N}.
\end{aligned}
$$

Setting $\varepsilon_1 = 0.1\varepsilon$, $\varepsilon_0 = 0.1\beta^{-1}\varepsilon$, $\eta = 0.1N|\mathcal{S}|^{-1}\beta^{-1}\varepsilon$, $T = \eta^{-1}\log 10B\varepsilon^{-1}$ leads to the final result. □

Algorithm B.2: Parallel MaxEnt

```
Input: Step size $\eta$, number of iterations $T$, number of agents $N$, planning oracle tolerance $\varepsilon_1 > 0$, distribution estimation oracle tolerance $\varepsilon_0 > 0$.
Set $\{C_0^i = \{\pi_0^i\}\}_{i \in N}$ where $\pi_0^i$ is an arbitrary policy, $\alpha_0^i = 1$.
for $t = 0, \ldots, T-1$ do
    Each agent call the state distribution oracle on $\pi_{\text{mix},t} = \frac{1}{N}\sum_i (\alpha_t^i, C_t^i)$:
        $\hat{d}^i_{\pi_{\text{mix},t}} = \textsc{DensityEst}\left(\pi_{\text{mix},t}, \varepsilon_0\right)$
    Define the reward function $r_t^i$ for each agent $i$ as
        $r_t^i(s) = \nabla H(\hat{d}^i_{\pi_{\text{mix},t}}) := \left.\frac{d\mathcal{H}(X)}{dX}\right|_{X = \hat{d}^i_{\pi_{\text{mix},t}}}$.
    Each agent computes the (approximately) optimal policy on $r_t$:
        $\pi_{t+1}^i = \textsc{ApproxPlan}\left(r_t^i, \varepsilon_1\right)$.
    Each agent updates
        $C_{t+1}^i = (\pi_0^i, \ldots, \pi_t^i, \pi_{t+1}^i),$    (B.13)
        $\alpha_{t+1}^i = ((1-\eta)\alpha_t^i, \eta).$    (B.14)
end for
$\pi_{\text{mix},T} = \frac{1}{N}\sum_i (\alpha_T^i, C_T^i)$.
```

The Use of Markovian and Non-Markovian Policies in cMGs with Finite-Trials.

The following result describes how in cMGs, as for convex MDPs, Non-Markovian policies are the right policy class to employ to guarantee well-behaved results.

Lemma B.2.5 (Sufficiency of Disjoint Non-Markvoian Policies)**.** *For every Convex Markov Game $\mathcal{M}$ there exist a Joint policy $\pi^\star = (\pi^{\star,i})_{i\in\mathcal{N}}$, with $\pi^{\star,i} \in \Delta_{\mathcal{S}^T}^{\mathcal{A}^i}$ being a deterministic Non-Markovian policy, that is a Nash Equilibrium for non-Disjoint single-trial objectives, for $K = 1$.*

Proof. The proof builds over a straight reduction. We build from the original MG $\mathcal{M}$ a temporally extended Markov Game $\tilde{\mathcal{M}} = (\mathcal{N}, \tilde{\mathcal{S}}, \mathcal{A}, \mathbb{P}, r, \mu, T)$. A state $\tilde{s}$ is defined for each history that can be induced, i.e., $\tilde{s} \in \tilde{\mathcal{S}} \iff \mathbf{s} \in \mathcal{S}^T$. We keep the other objects equivalent, while we define the extended transition model by only looking at the history's last entry to compute the probability conditioned on the next history. We introduce a common reward function across all the agents $r : \tilde{\mathcal{S}} \to \mathbb{R}$ such that $r(\tilde{s}) = \mathcal{H}(d(\tilde{s}))$ for Joint objectives and $r(\tilde{s}) = (1/N)\sum_{i\in[\mathcal{N}]}\mathcal{H}(d_i(\tilde{s}_i))$ for mixture objectives, for any history of horizon T and 0 otherwise. We now know that there is a deterministic Markovian policy such that $\tilde{\pi}^\star = (\tilde{\pi}^i)_{i\in\mathcal{N}}, \tilde{\pi}^i \in \Delta_{\tilde{\mathcal{S}}}^{\mathcal{A}_i}$ that is a Nash Equilibrium for $\tilde{\mathcal{M}}$ [Leonardos et al., 2022, Theorem 3.1]. Since $\tilde{s}$ corresponds to the set of histories of the original game, $\tilde{\pi}^\star$ maps to a non-Markovian policy in it. Finally, it is straightforward to notice that the NE of $\tilde{\pi}^\star$ for $\tilde{\mathcal{M}}$ implies the NE of $\tilde{\pi}^\star$ for the original CMG $\mathcal{M}$. □

The previous result implicitly asks for policies conditioned over the Joint state space, as happened for infinite-trials objectives as well. Interestingly, finite-trials objectives allow for a further characterization of how an optimal Markovian policy would behave when conditioned on the per-agent states only:

Lemma B.2.6 (Behavior of Optimal Markovian Decentralized Policies)**.** *Let $\pi_{NM} = (\pi^i_{NM} \in \Delta^{\mathcal{A}^i}_{\mathcal{S}^T})_{i \in [\mathcal{N}]}$ an optimal deterministic non-Markovian centralized policy and $\bar{\pi}_M = (\bar{\pi}^i_M \in \Delta^{\mathcal{A}^i}_{\mathcal{S}})_{i \in [\mathcal{N}]}$ the optimal Markovian centralized policy, namely $\bar{\pi}_M = \arg\max_{\pi = (\pi^i \in \Delta^{\mathcal{A}^i}_{\mathcal{S}})_{i \in [\mathcal{N}]}} \zeta_1(\pi)$. For a fixed sequence $\mathbf{s}_t \in \mathcal{S}^t$ ending in state $s = (s_i, s_{-i})$, the variance of the event of the optimal Markovian decentralized policy $\pi_M = (\pi^i_M \in \Delta^{\mathcal{A}^i}_{\mathcal{S}_i})_{i \in [\mathcal{N}]}$ taking $a^* = \pi_{NM}(\cdot|\mathbf{s}_t) = \bar{\pi}_M(\cdot|s,t)$ in s_i at step t is given by*

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\mathcal{B}(\pi_M(a^*|s_i,t))\right] &= \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}\oplus s \sim p_t^{\pi_{NM}}}\left[\mathbb{E}\left[\mathcal{B}(\pi_{NM}(a^*|\mathbf{s}\oplus s))\right]\right] \\ &+ \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}\oplus(\cdot,s_{-i})\sim p_t^{\bar{\pi}_M}}\left[\mathbb{E}\left[\mathcal{B}(\bar{\pi}_M(a^*|s_i,s_{-i},t))\right]\right].\end{aligned}$$

where $\mathbf{s} \oplus s \in \mathcal{S}^t$ is a generic t-lenght sequence with s as last state, that is $\mathbf{s} \oplus s := (\mathbf{s}_{t-1} \in \mathcal{S}^{t-1}) \oplus s$, and $\mathcal{B}(x)$ is a Bernoulli with parameter x.

Unsurprisingly, this Lemma shows that whenever the optimal Non-Markovian strategy for requires to adapt its decision in a Joint state s according to the history that led to it, an optimal Markovian policy for the same objective must necessarily be a stochastic policy, additionally, whenever the optimal Markovian policy conditioned over per-agent states only will need to be stochastic whenever the optimal Markovian strategy conditioned on the full states randomizes its decision based on the Joint state s.

Proof. Let us consider the random variable $A_i \sim \mathcal{P}_i$ denoting the event "the agent i takes action $a^*_i \in \mathcal{A}_i$". Through the law of total variance [Bertsekas and Tsitsiklis, 2002], we can write the variance of A given $s \in \mathcal{S}$ and $t \geqslant 0$ as

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[A|s,t\right] &= \mathbb{E}\left[A^2|s,t\right] - \mathbb{E}\left[A|s,t\right]^2 \\ &= \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{E}\left[A^2|s,t,\mathbf{s}\right]\Big] - \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{E}\left[A|s,t,\mathbf{s}\right]\Big]^2 \\ &= \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{V}\mathrm{ar}\left[A|s,t,\mathbf{s}\right] + \mathbb{E}\left[A|s,t,\mathbf{s}\right]^2\Big] - \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathop{\mathbb{E}}_{\pi}\left[A|s,t,\mathbf{s}\right]\Big]^2 \\ &= \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{V}\mathrm{ar}\left[A|s,t,\mathbf{s}\right]\Big] + \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{E}\left[A|s,t,\mathbf{s}\right]^2\Big] - \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{E}\left[A|s,t,\mathbf{s}\right]\Big]^2 \\ &= \mathop{\mathbb{E}}_{\mathbf{s}}\Big[\mathbb{V}\mathrm{ar}\left[A|s,t,\mathbf{s}\right]\Big] + \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}}\Big[\mathbb{E}\left[A|s,t,\mathbf{s}\right]\Big]. \end{aligned} \tag{B.15}$$

Now let the conditioning event $\mathbf{s}$ be distributed as $\mathbf{s} \sim p^{\pi_{\mathrm{NM}}}_{t-1}$, so that the condition $s, t, \mathbf{s}$ becomes $\mathbf{s} \oplus s$ where $\mathbf{s} \oplus s = (s_0, a_0, s_1, \dots, s_t = s) \in \mathcal{S}^t$, and let the variable A be distributed according to $\mathcal{P}$ that maximizes the objective given the conditioning. Hence, we have that A follows a Bernoulli distributions $\mathcal{B}(\bar{\pi}_{\mathrm{M}}(a^*|s,t))$, and the variable A on the right hand side of (B.16) is distributed as a Bernoulli $\mathcal{B}(\pi_{\mathrm{NM}}(a^*|\mathbf{s} \oplus s))$. Thus, we obtain

$$\mathbb{V}\mathrm{ar}\left[\mathcal{B}(\bar{\pi}_{\mathrm{M}}(a^*|s,t))\right] = \mathop{\mathbb{E}}_{\mathbf{s}\oplus s\sim p_t^{\pi_{\mathrm{NM}}}}\left[\mathbb{V}\mathrm{ar}\left[\mathcal{B}(\pi_{\mathrm{NM}}(a^*|\mathbf{s}\oplus s))\right]\right] + \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}\oplus s\sim p_t^{\pi_{\mathrm{NM}}}}\left[\mathbb{E}\left[\mathcal{B}(\pi_{\mathrm{NM}}(a^*|\mathbf{s}\oplus s))\right]\right]. \tag{B.16}$$

We know from Lemma B.2.5 that the policy π_{NM} is deterministic, so that $\mathbb{V}\mathrm{ar}\left[\mathcal{B}(\pi_{\mathrm{NM}}(a^*|\mathbf{s}\oplus s))\right] = 0$ for every $\mathbf{s} \oplus s$. We then repeat the same steps in order to compare the two different Markovian policies:

$$\mathbb{V}\mathrm{ar}\left[A|s_i,t\right] = \mathop{\mathbb{E}}_{s_{-i}}\Big[\mathbb{V}\mathrm{ar}\left[A|s_i,s_{-i},t\right]\Big] + \mathop{\mathbb{V}\mathrm{ar}}_{s_{-i}}\Big[\mathbb{E}\left[A|s_i,s_{-i},t\right]\Big].$$

Repeating the same considerations as before we get that we can use (B.16) to get:

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[\mathcal{B}(\pi_{\mathrm{M}}(a^*|s_i,t))\right] &= \mathop{\mathbb{E}}_{\mathbf{s}\oplus(\cdot,s_{-i})\sim p_t^{\bar{\pi}_{\mathrm{M}}}}\left[\mathbb{V}\mathrm{ar}\left[\mathcal{B}(\bar{\pi}_{\mathrm{M}}(a^*|s_i,s_{-i},t))\right]\right] + \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}\oplus(\cdot,s_{-i})\sim p_t^{\bar{\pi}_{\mathrm{M}}}}\left[\mathbb{E}\left[\mathcal{B}(\bar{\pi}_{\mathrm{M}}(a^*|s_i,s_{-i},t))\right]\right] \\ &= \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}\oplus s\sim p_t^{\pi_{\mathrm{NM}}}}\left[\mathbb{E}\left[\mathcal{B}(\pi_{\mathrm{NM}}(a^*|\mathbf{s}\oplus s))\right]\right] + \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{s}\oplus(\cdot,s_{-i})\sim p_t^{\bar{\pi}_{\mathrm{M}}}}\left[\mathbb{E}\left[\mathcal{B}(\bar{\pi}_{\mathrm{M}}(a^*|s_i,s_{-i},t))\right]\right].\end{aligned}$$

□

B.3 Proofs of Appendix A

Main Proof and Lemmas

In this section, we proceed to provide a proof of Theorem A.2.1, together with some useful lemmas instrumental for proving it. Again, we define the set containing the solutions to the expected and sampled Max-Ent problems with $\mathcal{S} := \{\bar{\eta}, \hat{\eta}\}$, the related set for the multipliers $\Omega_{\mathcal{S}} := \{\bar{\lambda}, \hat{\lambda}\}$, which is a restriction of $\Omega = \{\lambda \in \mathbb{R}^M : A(\lambda) < +\infty\}$, and a quantity that will be central now on $h(x_1, \cdots, x_N) := \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_i^N \log \eta(x_i)|$.

Contribution Highlights The whole structure of the proof is built upon several intermediate results, of which some use standard techniques, and others are novel to this work. Here we report some comments to better clarify our contributions:

- Lemma B.3.1 bounds the generalization-error with $h(\cdot)$, and it is based on the straighforward combination of Lemma B.3.2 and Lemma B.3.3.
- Lemma B.3.2 introduces a slight modification to Wang et al. [2013] that is the use of the $\max_{\Omega_{\mathcal{S}}}$ over a finite set rather than $\sup_{\Omega}$ over the entire set of distributions. This will allow us to combine the result with the one of Lemma B.3.3 and to deal with a simpler term, namely $h(x_1, \cdots, x_N)$ defined over the max instead of the sup.
- Lemma B.3.3 is a novel contribution, which was needed to obtain a practical form for the generalization error, compared to the intermediate result of Wang et al. [2013]. In this lemma as well $\max_{\Omega_{\mathcal{S}}}$ is employed, rather than $\sup_{\Omega}$.
- Lemma B.3.4 uses standard techniques as can be found in van der Vaart and Wellner [1996], Dudley [1999], Koltchinskii and Panchenko [2002], but the analysis is again restricted to $\max_{\Omega_{\mathcal{S}}}$ thanks to the previous results.
- Lemma B.3.5, Lemma B.3.6 are novel results. They are needed to derive a practical generalization-error bound. Lemma B.3.5 upper-bounds $\|\bar{\lambda}\|$ with $\|\hat{\lambda}\|$ by requiring additional constraints about the expressiveness of the feature functions. Lemma B.3.6 uses this result to substitute $\max_{\lambda \in \Omega_{\mathcal{S}}} \|\lambda\|$ with $\|\hat{\lambda}\|$.

As previously said, one of the main positives of this derivation is the ability to operate over $\max_{\eta \in \mathcal{S}}$ rather than $\sup_{\lambda \in \Omega}$. We will highlight the passages where this quantity is introduced with a $(\star)$, and provide further comments.

Initial step

First of all, we proceed in bounding the generalization error by bounding two sub-terms building it, that the following Lemma B.3.1 will consist of a combination of two following lemmas, Lemma B.3.2 and Lemma B.3.3.

Lemma B.3.1. *The generalization error between the true distribution and the Max-Ent solution of the sampled problem η^π, $\hat{\eta}$ (expressed as KL-divergence between the two distributions), given N i.i.d. samples, can be bounded with the following quantity:*

$$D_{\mathrm{KL}}(\eta^\pi || \hat{\eta}) \leqslant -\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}) + 5 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|$$

Proof. As said, the result directly follows by considering that for the problem under consideration $D_{\mathrm{KL}}(\eta^\pi || \hat{\eta}) = D_{\mathrm{KL}}(\bar{\eta} || \hat{\eta}) + D_{\mathrm{KL}}(\eta^\pi || \bar{\eta})$, since the two solutions correspond to the exact and sampled estimation problems. To bound the term on the right it is sufficient to bound the two terms on the left. We know that according to Lemma B.3.2,

$$D_{\mathrm{KL}}(\bar{\eta} || \hat{\eta}) \leqslant 2 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|$$

And according to Lemma B.3.3

$$D_{\mathrm{KL}}(\eta^\pi || \bar{\eta}) \leqslant -\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}) + 3 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|$$

And the result directly follows. □

Lemma B.3.2. *For the solutions of the exact and sampled Max-Ent problems, $\bar{\eta}$ and $\hat{\eta}$ respectively, it holds that*

$$D_{\mathrm{KL}}(\bar{\eta}||\hat{\eta}) \leqslant 2 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|$$

Proof.

$$\begin{aligned} D_{\mathrm{KL}}(\bar{\eta}||\hat{\eta}) &= D_{\mathrm{KL}}(\eta^\pi||\hat{\eta}) - D_{\mathrm{KL}}(\eta^\pi||\bar{\eta}) \\ &= (\underset{\eta^\pi}{\mathbb{E}}[\log \bar{\eta}] - \underset{\bar{\eta}}{\mathbb{E}}[\log \bar{\eta}]) + (\underset{\bar{\eta}}{\mathbb{E}}[\log \hat{\eta}] - \underset{\eta^\pi}{\mathbb{E}}[\log \hat{\eta}]) + (\underset{\bar{\eta}}{\mathbb{E}}[\log \bar{\eta}] - \underset{\bar{\eta}}{\mathbb{E}}[\log \hat{\eta}]) \\ &\leqslant 2 \max_{\eta \in \mathcal{S} = \{\bar{\eta}, \hat{\eta}\}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| + \frac{1}{N} \sum_{j \in [N]} \log \frac{\bar{\eta}(x_j)}{\hat{\eta}(x_j)} \quad (\star) \\ &\leqslant 2 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \end{aligned}$$

where the term $\frac{1}{N} \sum_{j \in [N]} \log \frac{\bar{\eta}(x_j)}{\hat{\eta}(x_j)}$ is negative and then is removed from the bounding scheme.

($\star$) Here, Wang et al. [2013] bounded conservatively the first two terms $(\mathbb{E}_{\eta^\pi}[\log \bar{\eta}] - \mathbb{E}_{\bar{\eta}}[\log \bar{\eta}]) + (\mathbb{E}_{\bar{\eta}}[\log \hat{\eta}] - \mathbb{E}_{\eta^\pi}[\log \hat{\eta}])$ with the $\sup_{\lambda \in \Omega}$, yet we notice that the only two quantities of interest between which we are asked to maximize over are in the $\max_{\eta \in \mathcal{S} = \{\bar{\eta}, \hat{\eta}\}}$. □

Lemma B.3.3. *For the solutions of the Max-Ent problem in expectation $\bar{\eta}$ it is possible to bound the KL-divergence with respect to the true distribution η^π with the following quantity*

$$D_{\mathrm{KL}}(\eta^\pi||\bar{\eta}) \leqslant -\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}) + 3 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|$$

Proof.

$$\begin{aligned} |\mathcal{L}_{\eta^\pi}(\bar{\eta}) - \tilde{\mathcal{L}}(\hat{\eta})| &= |\mathbb{E}_{\eta^\pi}[\log \bar{\eta}] - \frac{1}{N} \sum_{j \in [N]} \log \hat{\eta}(x_j)| \\ &\leqslant |\mathbb{E}_{\eta^\pi}[\log \bar{\eta}] - \mathbb{E}_{\eta^\pi}[\log \hat{\eta}]| + |\mathbb{E}_{\eta^\pi}[\log \hat{\eta}] - \frac{1}{N} \sum_{j \in [N]} \log \hat{\eta}(x_j)| \\ &\leqslant |D_{\mathrm{KL}}(\eta^\pi||\bar{\eta}) - D_{\mathrm{KL}}(\eta^\pi||\hat{\eta})| + \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \quad (\star) \\ &\leqslant 2 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| + \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \\ &\leqslant 3 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \end{aligned}$$

($\star$) Again, due to the conservative bound in Lemma B.3.2, Wang et al. [2013] maintained the same quantity in this bound for later simplifications. We apply a tighter bound of $\max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|$ to $|\mathbb{E}_{\eta^\pi}[\log \hat{\eta}] - \frac{1}{N} \sum_{j \in [N]} \log \hat{\eta}(x_j)|$.

It follows that it is possible to write

$$\begin{aligned} |\mathcal{L}_{\eta^\pi}(\bar{\eta}) - \tilde{\mathcal{L}}(\hat{\eta})| &= |D_{\mathrm{KL}}(\eta^\pi||\bar{\eta}) + \mathcal{H}(\eta^\pi) - \tilde{\mathcal{L}}(\hat{\eta})| \\ |D_{\mathrm{KL}}(\eta^\pi||\bar{\eta}) - (-\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}))| &\leqslant 3 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \\ ||D_{\mathrm{KL}}(\eta^\pi||\bar{\eta})| - |(-\mathcal{H}(\eta^\pi) + \tilde{\mathcal{L}}(\hat{\eta}))|| &\leqslant 3 \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \end{aligned}$$

which proves the result. □

Intermediate Step

As suggested by the previous considerations, everything boils down to being able to bound the term $h(x_1, \cdots, x_N) := \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N}\sum_{j \in [N]} \log \eta(x_j)|$. To do this, we used standard techniques to derive the following intermediate step, where we can bound the quantity of interest which depends on the supremum between distributions $\max_{\eta \in \mathcal{S}} |\cdot|$ with a quantity depending on the supremum between their respective parameters $\lambda \in \Omega_{\mathcal{S}}$, namely $\sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1$.

Lemma B.3.4. *The supremum difference between the expected log-likelihood and the sampled one, taken over the expected and sampled solutions in $\mathcal{S} = \{\bar{\lambda}, \hat{\lambda}\}$, is defined as $h(x_1, \cdots, x_N) := \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N}\sum_{j \in [N]} \log \eta(x_j)|$ and it can be bounded by*

$$\max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N}\sum_{j \in [N]} \log \eta(x_j)| \leqslant 2 \sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1 \mathcal{R}_N(\Phi) + 2 \sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1 F \sqrt{\frac{\log 1/\delta}{2N}}$$

with $F = \sup_{f \in \mathcal{F}} ||f||_\infty$.

Proof. We define

$$\begin{aligned} h(x_1, \ldots, x_N) &= \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N}\sum_{j \in [N]} \log \eta(x_j)| \\ &= \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} |\mathbb{E}_{\eta^\pi}\langle \lambda, f(x)\rangle - \frac{1}{N}\sum_{j \in [N]} \langle \lambda, f(x)\rangle| \end{aligned}$$

Then by exploiting the definition of the function, we study the differences induced by changing one sample from x_k to x'_k

$$\begin{aligned} &|h(x_1, \ldots, x_M) - h(x_1, \ldots, x'_k, \ldots, x_M)| = \\ &= |\sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} |\mathbb{E}_{\eta^\pi}\langle \lambda, f(x)\rangle - \frac{1}{N}\sum_{j \in [N]} \langle \lambda, f(x)\rangle| \\ &- \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} |\mathbb{E}_{\eta^\pi}\langle \lambda, f(x)\rangle - \frac{1}{N}\sum_{j \in [N] \neq k} \langle \lambda, f(x)\rangle + \langle \lambda, f(x'_k)\rangle|| \\ &\leqslant \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} \frac{1}{N} |\langle \lambda, f(x_k) - f(x'_k)\rangle| \\ &\leqslant \frac{2}{N} \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} ||\lambda||_1 ||f||_\infty = \frac{C}{N} \quad (C = 2 \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} ||\lambda||_1 ||f||_\infty) \end{aligned}$$

Now, by Mc Diarmid's inequality, by studying the function concerning its sampled expectation $\mathbb{E}_{\tilde{\mathcal{X}}} h(\cdot)$ over the samples set $\tilde{\mathcal{X}} = \{x_1, \ldots, x_N\}$:

$$\begin{aligned} P(h(x_1, \ldots, x_N) - \mathbb{E}_{\tilde{\mathcal{X}}} h(x_1, \ldots, x'_k, \ldots, x_N) \geqslant \epsilon) \leqslant \exp(\frac{-2N\epsilon^2}{C^2}) \\ P\Big(h(x_1, \ldots, x_N) - \mathbb{E}_{\tilde{\mathcal{X}}} h(x_1, \ldots, x'_k, \ldots, x_N) \geqslant C\sqrt{\frac{\log 1/\delta}{2N}}\Big) \leqslant \delta \end{aligned}$$

It then follows that

$$\max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N}\sum_{j \in [N]} \log \eta(x_j)| \leqslant \mathbb{E}_{\tilde{\mathcal{X}}} \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} |\mathbb{E}_{\eta^\pi}\langle \lambda, f(x)\rangle - \frac{1}{N}\sum_{j \in [N]} \langle \lambda, f(x)\rangle| + C\sqrt{\frac{\log 1/\delta}{2N}}$$

We now use symmetrization techniques by considering the Rademacher sequence $\{\omega_j\}$ and by using the standard result that given a class of measurable functions $\mathcal{G}$ if

$$Z(\tilde{\mathcal{X}}) = \sup_{g \in \mathcal{G}} |\mathbb{E}\, g(x) - \frac{1}{N}\sum_{j \in [N]} g(x_j)| \quad \text{and } R(\tilde{\mathcal{X}}, \omega) = \sup_{g \in \mathcal{G}} |\frac{1}{N}\sum_{j \in [N]} \omega_j g(x_j)|$$

Then:

$$\mathbb{E}_{\tilde{\mathcal{X}}} Z(\tilde{\mathcal{X}}) \leqslant 2 \mathbb{E}_{\tilde{\mathcal{X}}, \omega} R(\tilde{\mathcal{X}})$$

From this, it follows that the whole expression reduces to

$$\max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| \leqslant 2 \mathop{\mathbb{E}}_{\tilde{\mathcal{X}}, \omega} \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} |\frac{1}{N} \sum_{j \in [N]} \omega_j \langle \lambda, f(x_j) \rangle| + C \sqrt{\frac{\log 1/\delta}{2N}}$$

We extract the supremum over $\lambda \in \Omega_{\mathcal{S}}$ to obtain the (absolute) Rademacher averages of the functions in $\mathcal{F}$

$$\begin{aligned} \mathop{\mathbb{E}}_{\omega} \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} |\frac{1}{N} \sum_{j \in [N]} \omega_j \langle \lambda, f(x_j) \rangle| &\leqslant \sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1 \mathop{\mathbb{E}}_{\omega} \sup_{f \in \mathcal{F}} |\frac{1}{N} \sum_{j \in [N]} \omega_j f(x_j)| \\ &\leqslant \sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1 \mathcal{R}_N(\Phi) \end{aligned}$$

It follows that the final formulation for the term we are studying is the following

$$\begin{aligned} \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)| &\leqslant 2 \sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1 \mathcal{R}_N(\Phi) + C \sqrt{\frac{\log 1/\delta}{2N}} \\ C &= 2 \sup_{\lambda \in \Omega_{\mathcal{S}}, f \in \mathcal{F}} ||\lambda||_1 ||f||_\infty \end{aligned}$$

□

Final Step

The bound offered by Lemma B.3.4 would be unpractical since it relates a quantity central to our analysis to something which is not known in advance. Due to this, we make a further effort with the following Lemma, by substituting the term $\sup_{\lambda \in \Omega_{\mathcal{S}}} ||\lambda||_1$ with $||\hat{\lambda}||_1$. To do this, an additional assumption over the feature functions will be needed though. First of all, we bound the two terms in $\Omega_{\mathcal{S}}$ with

Lemma B.3.5. *The solutions of the expected and sampled Max-Ent problem are related to the bound:*

$$||\bar{\lambda}||_1 \leqslant ||\hat{\lambda}||_1 + \sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\text{ov}}(\mathcal{F}))} \max_{\eta \in \mathcal{S}} |\mathbb{E}_{\eta^\pi}[\log \eta] - \frac{1}{N} \sum_{j \in [N]} \log \eta(x_j)|}$$

Proof. We take into account the following relationships which are valid for the solutions of the MaxEnt problem under structural constraints, i.e. $\mathbb{E}_{\bar{\eta}}[f] = \mathbb{E}_{\eta^\pi}[f]$ and $\mathbb{E}_{\hat{\eta}}[f] = \mathbb{E}_{\tilde{\eta}}[f]$

$$\begin{aligned} \mathcal{H}(\eta) &= \log \sum_y \exp(\langle \lambda, f(y) \rangle) - \langle \lambda, \mathop{\mathbb{E}}_{\eta}[f] \rangle \\ &= A(\lambda) - \langle \lambda, \mathop{\mathbb{E}}_{\eta}[f] \rangle = A(\lambda) - \langle \lambda, \nabla A(\lambda) \rangle \end{aligned}$$

From which it follows that it is possible to recover the Bregman divergence under the log-partition function $D_A(\lambda_1, \lambda_2)$

$$\begin{aligned} \mathcal{H}(\bar{\eta}) - \mathcal{H}(\hat{\eta}) &= A(\bar{\lambda}) - A(\hat{\lambda}) - \langle \bar{\lambda}, \nabla A(\bar{\lambda}) \rangle + \langle \hat{\lambda}, \nabla A(\hat{\eta}) \rangle \\ &= A(\bar{\lambda}) - A(\hat{\lambda}) - \langle \bar{\lambda}, \nabla A(\bar{\lambda}) \rangle + \langle \hat{\lambda}, \nabla A(\hat{\lambda}) \rangle + \langle \bar{\lambda}, \nabla A(\hat{\lambda}) \rangle - \langle \bar{\lambda}, \nabla A(\hat{\eta}) \rangle \\ &= A(\bar{\lambda}) - A(\hat{\lambda}) - \langle \bar{\lambda} - \hat{\lambda}, \nabla A(\hat{\lambda}) \rangle + \langle \hat{\lambda}, \nabla A(\hat{\lambda}) - \nabla A(\bar{\lambda}) \rangle \\ &= D_A(\bar{\lambda}, \hat{\lambda}) + \langle \bar{\lambda}, \nabla A(\hat{\lambda}) - \nabla A(\bar{\lambda}) \rangle \end{aligned}$$

Now using the Taylor expansion of the divergence and the fact that $\nabla^2 A(\hat{\lambda}) = \hat{\mathbb{C}\text{ov}}(\mathcal{F})$

$$\begin{aligned} \mathcal{H}(\bar{\eta}) - \mathcal{H}(\hat{\eta}) + \langle \bar{\lambda}, \nabla A(\bar{\lambda}) - \nabla A(\hat{\lambda}) \rangle &= D_A(\bar{\lambda}, \hat{\lambda}) \\ &\geqslant \frac{1}{2} (\bar{\lambda} - \hat{\lambda})^\intercal \nabla^2 A(\hat{\lambda}) (\bar{\lambda} - \hat{\lambda}) = \frac{1}{2} ||\bar{\lambda} - \hat{\lambda}||^2_{\nabla^2 A(\hat{\lambda})} \\ &\geqslant \sigma_{\min}(\nabla^2 A(\hat{\lambda})) ||\bar{\lambda} - \hat{\lambda}||_2^2 \\ &\geqslant \frac{\sigma_{\min}(\nabla^2 A(\hat{\lambda}))}{M} ||\bar{\lambda} - \hat{\lambda}||_1^2 \\ &\geqslant \frac{\sigma_{\min}(\hat{\mathbb{C}\text{ov}}(\mathcal{F}))}{M} ||\bar{\lambda} - \hat{\lambda}||_1^2 \end{aligned}$$

where M corresponds to the number of the features. Finally, by exploiting the zero duality gap and the results of Lemma B.3.8

$$\begin{aligned}
||\bar\lambda - \hat\lambda||_1^2 &\leqslant \frac{M}{\sigma_{\min}(\mathbb{C}\mathrm{ov}_{\hat\lambda}(f))}(\mathcal{H}(\bar\eta) - \mathcal{H}(\hat\eta) + \langle\bar\lambda, \nabla A(\bar\lambda) - \nabla A(\hat\lambda)\rangle) \\
&= \frac{M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}(\mathcal{L}_0(\bar\lambda) - \tilde{\mathcal{L}}(\hat\lambda) + \langle\bar\lambda, \nabla A(\bar\lambda) - \nabla A(\hat\lambda)\rangle) \\
&\leqslant \frac{M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}(|\mathcal{L}_0(\bar\lambda) - \tilde{\mathcal{L}}(\hat\lambda)| + |\langle\bar\lambda, \nabla A(\bar\lambda) - \nabla A(\hat\lambda)\rangle|) \\
&\leqslant \frac{2M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}|\mathcal{L}_0(\bar\lambda) - \tilde{\mathcal{L}}(\hat\lambda)| \\
&\leqslant \frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)|
\end{aligned}$$

It is then possible to write

$$\begin{aligned}
||\bar\lambda - \hat\lambda||_1 &\leqslant \sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)|} \\
|||\bar\lambda||_1 - ||\hat\lambda||_1| \leqslant ||\bar\lambda - \hat\lambda||_1 &\leqslant \sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)|}
\end{aligned}$$

which concludes the proof. □

Now, it is possible to combine all the previous results in

Lemma B.3.6. *Assume that the minimum singular value of the sampled covariance matrix is strictly positive, that is $\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F})) > 0$, then the supremum term of Lemma B.3.4 can be bounded with*

$$\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)| \lesssim 2||\hat\lambda||_1\mathcal{R}_N(\Phi) + 2||\hat\lambda||_1 F\sqrt{\frac{\log 1/\delta}{2N}}$$

Proof. Taking all together the terms obtained so far from Lemmas [B.3.4, B.3.5], setting $C = 2\sup_{\lambda\in\{\bar\lambda,\hat\lambda\},f\in\mathcal{F}}||\lambda||_1||f||_\infty$ we have

$$\begin{aligned}
\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)| &\leqslant 2\sup_{\lambda\in\{\bar\lambda,\hat\lambda\}}||\lambda||_1\mathcal{R}_N(\Phi) + C\sqrt{\frac{\log 1/\delta}{2N}} \\
\sup_{\lambda\in\{\bar\lambda,\hat\lambda\}}||\lambda||_1 &\leqslant ||\hat\lambda||_1 + \sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)|}
\end{aligned}$$

It follows the quadratic form in $x = \sqrt{\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)|}$

$$\begin{aligned}
&x^2 - bx - c \leqslant 0 \\
&b = 2\sqrt{\frac{6M}{\sigma_{\min}(\hat{\mathbb{C}\mathrm{ov}}(\mathcal{F}))}}\left[\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right] \geqslant 0 \\
&c = 2||\hat\lambda||_1\left[\mathcal{R}_N(\Phi) + F\sqrt{\frac{\log 1/\delta}{2N}}\right] \geqslant 0
\end{aligned}$$

The discriminant is well defined $\Delta = b^2 + 4c \geqslant 0$ and the solution is given by

$$\begin{aligned}
\max_{\eta\in\mathcal{S}}|\mathbb{E}_{\eta^\pi}[\log\eta] - \frac{1}{N}\sum_{j\in[N]}\log\eta(x_j)| &\leqslant \Big(\frac{b + \sqrt{b^2 + 4c}}{2}\Big)^2 \\
&\leqslant \frac{b^2}{2} + c + b\sqrt{b^2 + 4c} \\
&\lesssim 2||\hat\lambda||_1\mathcal{R}_N(\Phi) + 2||\hat\lambda||_1 F\sqrt{\frac{\log 1/\delta}{2N}}
\end{aligned}$$

The final step was done because all additional terms out of c itself are of higher order. □

Further instrumental Lemmas

In this section, we present some additional standard lemmas which summarize some important properties of the Max-Ent solutions and distributions in the exponential family that was used in the employed section.

Lemma B.3.7. *For any distribution η in the exponential family, it holds that for the log-likelihood with respect to a distribution η^π it holds that*

$$\mathcal{L}_{\eta^\pi}(\lambda) = A(\lambda) - \langle \lambda, \mathop{\mathbb{E}}_{\eta^\pi}[f] \rangle$$

Proof.

$$\mathcal{L}_{\eta^\pi}(\lambda) = - \mathop{\mathbb{E}}_{\eta^\pi}[\log \eta] = - \mathop{\mathbb{E}}_{\eta^\pi}[\langle \lambda, f \rangle - \log \Phi_\lambda] = -\langle \lambda, \mathop{\mathbb{E}}_{\eta^\pi}[f] \rangle + A(\lambda)$$

□

Lemma B.3.8. *For any distribution η in the exponential family, it holds that*

$$|\mathcal{L}_{\eta^\pi}(\lambda) - \tilde{\mathcal{L}}(\lambda)| = |\langle \lambda, \mathop{\mathbb{E}}_{\eta^\pi}[f] - \tilde{\mathbb{E}}[f] \rangle|$$

where $\mathcal{L}_{\eta^\pi}(\lambda)$ is the negative log-likelihood of η with respect to η^π.

Proof.

$$\begin{aligned} |\mathcal{L}_{\eta^\pi}(\lambda) - \tilde{\mathcal{L}}(\lambda)| &= | - \langle \lambda, \mathop{\mathbb{E}}_{\eta^\pi}[f] \rangle + A(\lambda) + \langle \lambda, \mathop{\mathbb{E}}_{\tilde{\eta}}[f] \rangle - A(\lambda)| \\ &= |\langle \lambda, - \mathop{\mathbb{E}}_{\eta^\pi}[f] + \mathop{\mathbb{E}}_{\tilde{\eta}}[f] \rangle| \\ &= |\langle \lambda, \mathop{\mathbb{E}}_{\eta^\pi}[f] - \mathop{\mathbb{E}}_{\tilde{\eta}}[f] \rangle| \end{aligned}$$

□

We will now derive some properties between the sampled log-likelihood and the log-likelihood with respect to the true distribution η^π, called $\mathcal{L}_0$ for simplicity

Lemma B.3.9. *For the solutions of the exact and sampled Max-Ent problems, $\bar{\eta}$ and $\hat{\eta}$ respectively, it holds that*

$$\begin{aligned} |\mathcal{L}_0(\bar{\lambda}) - \tilde{\mathcal{L}}(\hat{\lambda})| &\leqslant |\langle \hat{\lambda}, \mathop{\mathbb{E}}_{\eta^\pi}[f] - \mathop{\mathbb{E}}_{\tilde{\eta}}[f] \rangle| \\ &\leqslant |\langle \hat{\lambda}, \nabla A(\bar{\lambda}) - \nabla A(\hat{\lambda}) \rangle| \end{aligned}$$

$$\begin{aligned} |\mathcal{L}_0(\bar{\lambda}) - \tilde{\mathcal{L}}(\hat{\eta})| &\geqslant |\langle \bar{\lambda}, \mathop{\mathbb{E}}_{\eta^\pi}[f] - \mathop{\mathbb{E}}_{\tilde{\eta}}[f] \rangle| \\ &\geqslant |\langle \bar{\lambda}, \nabla A(\bar{\lambda}) - \nabla A(\hat{\lambda}) \rangle| \end{aligned}$$

Proof. The proof follows directly from the fact that $\bar{\lambda}$ is optimal with respect to $\hat{\eta}$ in the exact problem $\mathcal{L}_0(\bar{\lambda}) \leqslant \mathcal{L}_0(\hat{\lambda})$ and viceversa $\tilde{\mathcal{L}}(\bar{\lambda}) \geqslant \tilde{\mathcal{L}}(\hat{\eta})$. □

Monotonicity Lemma

In this section, we provide the proof of Lemma A.3.2.

Proof. Taking into account two features with increased factorization $\mathcal{F} \subset \mathcal{F}'$ we consider the particular set of factorized features $\bar{\alpha}, \{\bar{\alpha}_k\}$, since the rest of the features are the same. It follows that

$$\begin{aligned}
|\hat{\mu}_{\bar{\alpha}}| &= \sum_k |\hat{\mu}_{\bar{\alpha}_k}| \\
\frac{|\hat{\mu}_{\bar{\alpha}}|}{|S_{\bar{\alpha}}|} &= \sum_k \frac{|\hat{\mu}_{\bar{\alpha}_k}|}{|S_{\bar{\alpha}}|} \quad (\forall \bar{\alpha}_k : |S_{\bar{\alpha}_k}| < |S_{\bar{\alpha}}|) \\
\frac{|\hat{\mu}_{\bar{\alpha}}|}{|S_{\bar{\alpha}}|} &\leqslant \sum_k \frac{|\hat{\mu}_{\bar{\alpha}_k}|}{|S_{\bar{\alpha}_k}|}
\end{aligned}$$

Now, due to the relationship of Lemma B.3.10 we know that $\hat{\lambda}_\alpha = f(\frac{|\hat{\mu}_{\bar{\alpha}}|}{|S_{\bar{\alpha}}|})$ with $f(\cdot)$ being an unknown but subadditive for positive values of λ. Moreover, the functions are the same for all the terms, so that

$$\begin{aligned}
\frac{|\hat{\mu}_{\bar{\alpha}}|}{|S_{\bar{\alpha}}|} &\leqslant \sum_k \frac{|\hat{\mu}_{\bar{\alpha}_k}|}{|S_{\bar{\alpha}_k}|} \\
f(\frac{|\hat{\mu}_{\bar{\alpha}}|}{|S_{\bar{\alpha}}|}) &\leqslant f(\sum_k \frac{|\hat{\mu}_{\bar{\alpha}_k}|}{|S_{\bar{\alpha}_k}|}) \leqslant \sum_k f(\frac{|\hat{\mu}_{\bar{\alpha}_k}|}{|S_{\bar{\alpha}_k}|}) \\
|\lambda_{\bar{\alpha}}| &\leqslant \sum_k |\lambda_{\bar{\alpha}_k}|
\end{aligned}$$

Since the rest of the terms are the same, this concludes the proof. □

Lemma B.3.10. *There exists a monotonic and anti-symmetric function $f(\cdot)$ such that it is possible to univocally define $\hat{\lambda}_\alpha = f(\hat{\mu}_\alpha, |S_\alpha|, G_{max})$*

Proof. We start by considering the Lagrangian formulation of the Max-Ent problem,

$$\mathcal{L}(\eta, \lambda) = \mathcal{H}(\eta) + \sum_{\alpha \in \mathcal{I}_{\mathcal{F}}} \lambda_\alpha (\mathbb{E}_\eta[f_\alpha] - \hat{\mu}_\alpha) + \mu(\mathbb{E}[\eta] - 1) \tag{B.17}$$

By taking the gradient of the Lagrangian with respect to the distribution it follows that each x-term of the support gives

$$(\nabla_\eta \mathcal{L})(x) = -1 - \log \eta(x) + \lambda_\alpha f_\alpha + \mu$$

From which it follows that with $\lambda_0 = \mu - 1$ the equation for the α-constraint is

$$\eta_\alpha(x) = e^{\lambda_0} e^{\lambda_\alpha f_\alpha(x)}$$

We now compute insert this value inside the constraint equation under the feature class $f_\alpha = g 1_{s \in \mathcal{S}_\alpha}$

$$\begin{aligned}
\int_{\mathcal{R}} \int_{\mathcal{X}} g\eta(x)_\alpha &= \hat{\mu}_\alpha \\
|S_\alpha| \int_{G_{\min}}^{G_{\max}} g e^{\lambda_0} e^{\lambda_\alpha r} dg &= \hat{\mu}_\alpha
\end{aligned}$$

which leads to the implicit formulation for λ_α by solving the integral by setting $G = G_{\max}$

$$e^{\lambda_0} \frac{2\lambda_\alpha \cosh(G\lambda_\alpha) - 2\sinh(G\lambda_\alpha))}{\lambda_\alpha^2} = \frac{\hat{\mu}_\alpha}{|S_\alpha|}$$

Now, it can be proven by considering the normalization constraint that $e^{\lambda_0} = 1/Z(\lambda)$ with $Z(\lambda)$ a constant depending on λ_α, in particular:

$$\begin{aligned}
Z(\lambda) &= \int_{\mathcal{X}} e^{\sum_\alpha \lambda_\alpha f_\alpha} dx \\
&= \sum_\alpha |S_\alpha| \int_{\mathcal{R}} e^{\lambda_\alpha f_\alpha} dr \\
&= \sum_\alpha |S_\alpha| \frac{\sinh \lambda_\alpha G}{\lambda_\alpha} \\
&= |S_\alpha| \frac{\sinh \lambda_\alpha G}{\lambda_\alpha} + C
\end{aligned}$$

The whole equation then becomes

$$\frac{2\lambda_\alpha \cosh(G\lambda_\alpha) - 2\sinh(G\lambda_\alpha)}{\lambda_\alpha^2} = \hat{\mu}_\alpha(\frac{\sinh \lambda_\alpha G}{\lambda_\alpha} + C)$$

This equation provides an implicit definition for λ_α and it can be shown to be convex for positive values of λ. The function for lambda is the inverse of this whole term, which is then concave and has a zero in the origin, thus it is sub-additive. □

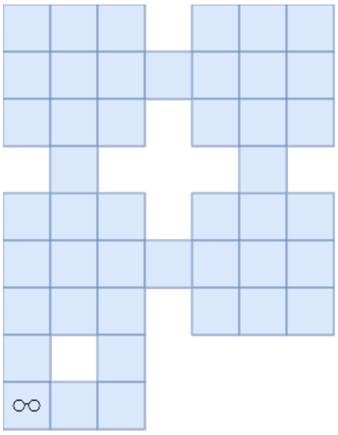
APPENDIX *C*

Experimental Details

C.1 Experimental Details of Section 4.1

In the following Section, we report additional details on the experiments of Section 4.2.4. Specifically, we describe the employed domains and their properties, we comment on the choice of hyper-parameters, and on the effect of the regularization on the results of *PG for Reg-MOE*.

Environments



Most of the reported experiments refer to the gridworld reported on the left, which is composed of a set of rooms connected by narrow corridors. The grid is composed of 44 cells, which define both the set of states ($|\mathcal{S}| = 44$) and observations ($|\mathcal{O}| = 44$). The set of actions $\mathcal{A}$ include an action to move to the adjacent cell in every direction ($|\mathcal{A}| = 4$). To every action is associated a probability of failure $\bar{p} = 0.1$ that leads the agent to an adjacent cell (at random) different from the one intended by the taken action. The episode horizon is $T = 55$ and the initial state distribution μ was set to be a deterministic over the top-left cell. The glasses icon in the bottom left cell of the grid represents a state that "flips" the behavior of the observations. This is only relevant in the experiment in Figure 5.3 and is better explained below. All the experiments of Section 4.2.4 were performed with a regularization factor $\beta = 0.8$ (for *PG for Reg-MOE*) and a learning rate of $\alpha = 0.9$. Finally, the batch size was $N = 6$ and the number of independent runs was set to 16.

Observations. The observations were set to be Gaussian distributions $\mathcal{G}(0, \sigma^2)$ over the Manhattan distance centered in the true state and without caring about any obstacles, with 0 mean and different values of variance σ^2. The resulting observation matrices are reported in Figures. Finally, the effect of "wearing" the glasses (i.e., reaching the bottom-left cell of the grid) is to make the observation function fully deterministic. Note that the information on whether the agent wears the glasses is encoded in the state themselves, doubling the size of the set of states to $|\mathcal{S}| = 88$.

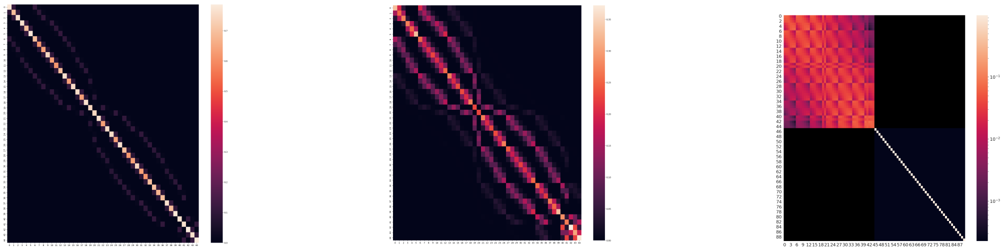


Figure C.1: *Exp. of Fig. 4.4*

Figure C.2: *Exp. of Fig. 4.5*

Figure C.3: *Exp. of Fig. 4.6*

Heatmaps of $\mathbb{O}$ in the experiments of Section 4.2.4.Figure C.3 has logarithmic scale.

Hyper-Parameters

In this section, we briefly discuss the choice behind the selection of specific hyper-parameters employed in the experiments.

Learning Rate. As for the learning rate α, a value of $\alpha = 0.9$ was selected across the experiments. As one can see from the Figures, the best performance were reached with a learning rate between $\alpha = 1$ and $\alpha = 0.7$, so $\alpha = 0.9$ can be seen as a robust choice across the boards.

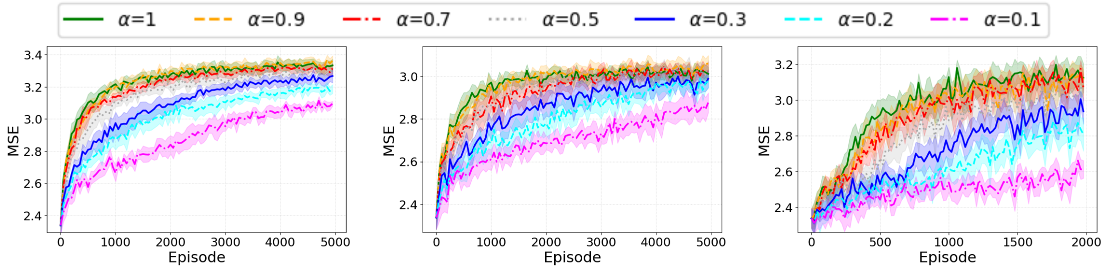


Figure C.4: *Exp. of Fig. 4.4*

Figure C.5: *Exp. of Fig. 4.5*

Figure C.6: *Exp. of Fig. 4.6*

Comparison of the performance with different values of the learning rate for various algorithms and domains.

Regularization. As for the regularization term β, the best performance for the various instances was generally reached with $\beta \in (0.3, 1)$ (the learning rate is fixed to $\alpha = 0.9$). For lower values of β, the effect of the regularization is almost negligible, while for higher values of β the agent tended to over-optimize the regularization term in place of the entropy over observations, reducing performance. As one would expect, the best value for the regularization depend on the specific POMDP instance.

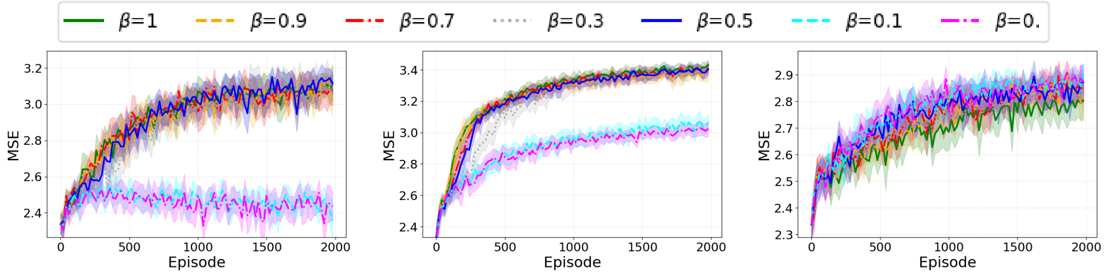


Figure C.7: $\sigma^2 = 10$

Figure C.8: $\sigma^2 = 1$

Figure C.9: $\sigma^2 = 0.25$

A comparison of different values of regularization for varying emission matrices' quality and settings with and without glasses. For the low value of regularization, the performances of Reg-MOE are equivalent to the MOE performances.

C.2 Experimental Details of Section 4.2

Environments

Here we report a visualization of the four types of domain taken into account.

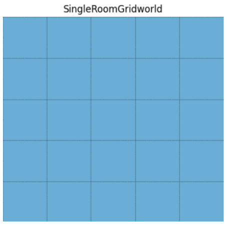


Figure C.10: *Single Room*

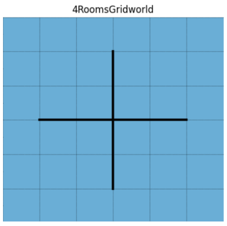


Figure C.11: *Four Rooms*

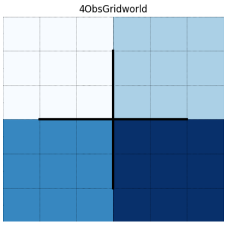


Figure C.12: *Four Rooms with 4 Observations*

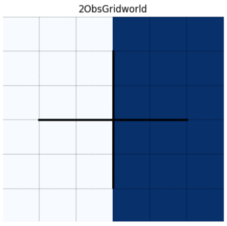


Figure C.13: *Four Rooms with 2 Observations*

Hyperparameters

The **learning rate** was selected as $\alpha = 0.3$. The **batch size** was selected to be $N = 10$ after tuning. As for the **time horizon**, $T = S$ in all the experiments. This makes the exploration task more challenging as every state can be visited at most once. The best regularization term ρ was found to be approximately equal to 0.02.

Policy Class

As already described, a plethora of deployable policy classes are possible for addressing MSE in POMDPs. In the main paper, we focused on belief-averaged policies. First, we show how this policy class is superior (or non-worse) to other possible options, being implicitly non-Markovian over observations while being memory efficient. Then, we show that belief-averaged policies perform better than (direct-parametrization) Markovian policies over belief states, even in the case when the belief states set is manageable in size.

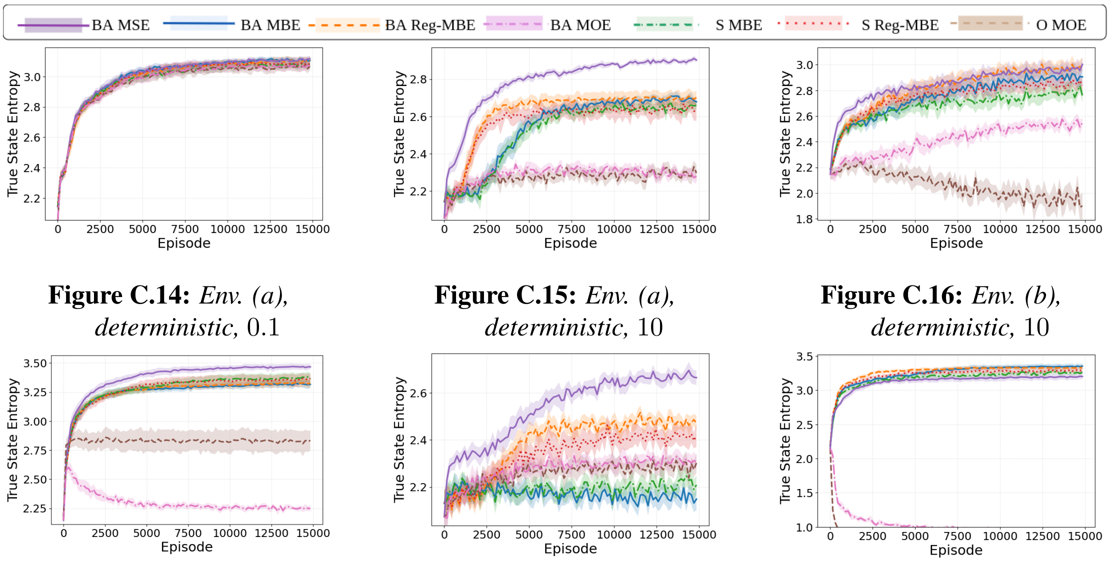


Figure C.14: *Env. (a), deterministic,* 0.1

Figure C.15: *Env. (a), deterministic,* 10

Figure C.16: *Env. (b), deterministic,* 10

Figure C.17: *Env. (c), deterministic, n.a.*

Figure C.18: *Env. (a), stochastic,* 10

Figure C.19: *Env. (d), deterministic, n.a.*

True state entropy obtained by Algorithm 1 *specialized for the feedbacks* MSE, MOE, MBE, MBE with belief regularization *(Reg-MBE) over different policy classes with direct parametrization: Markovian over observation (O), Belief Averaged (BA), Markovian over hallucinated states (S). For each plot, we report a tuple (environment, transition noise, observation variance) where the latter is* not available *(n.a.) when observations are deterministic. For each curve, we report the average and 95% c.i. over 16 runs. BA confirms to be the policy class with generally higher performance in all the considered instances.*

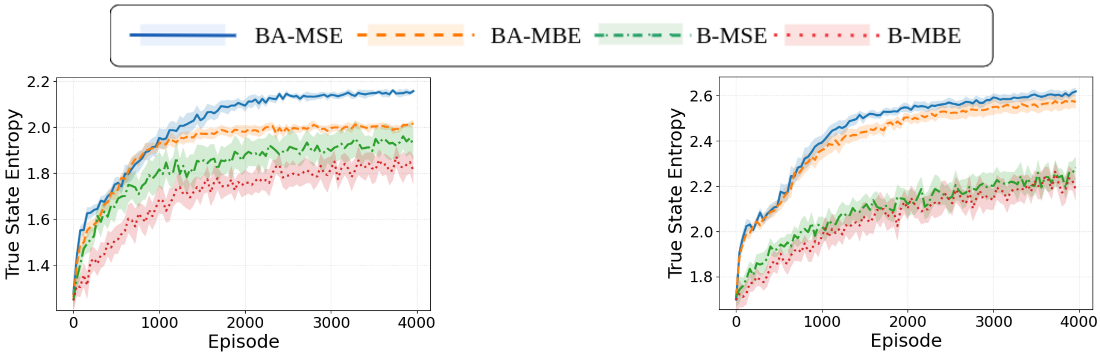


Figure C.20: *Env. (a)* $(|\mathcal{S}| = 9)$, *deterministic,* 0.2

Figure C.21: *Env. (a)* $(|\mathcal{S}| = 16)$, *deterministic,* 0.2

True state entropy obtained by Algorithm 1 with MSE and MBE employing belief averaged policies (BA) and Markovian policies over belief states (B). For each plot, we report a tuple (environment, transition noise, observation variance) where the latter is not available *(n.a.) when observations are deterministic. For each curve, we report the average and 95% c.i. over 16 runs. Limited size instances were reported since* $|\mathcal{B}| = 10^4$ *in C.20 and* $|\mathcal{B}| = 10^5$ *in C.21 leading to memory issues in the policies storage. Even in these cases, BA shows higher performances.*

C.3 Experimental Details of Section 5.4

Environments

The main empirical proof of concept was based on two environments. First, Env. (**i**), the so called *secret room* environment by Liu et al. [2021]. In this environment, two agents operate within two rooms of a 10×10 discrete grid. There is one switch in each room, one in position $(1, 9)$ (corner of first room), another in position $(9, 1)$ (corner of second room). The rooms are separated by a door and agents start in the same room deterministically at positions $(1, 1)$ and $(2, 2)$ respectively. The door will open only when one of the switches is occupied, which means that the (Manhattan) distance between one of the agents and the switch is less than 1.5. The full state vector contains x, y locations of the two agents and binary variables to indicate if doors are open *but* per-agent policies are conditioned on their respective states only and the state of the door. For Sparse-Rewards Tasks, the goal was set to be deterministically at the worst case, namely $(9, 9)$ and to provide a positive reward to both the agents of 100 when reached, which means again that the (Manhattan) distance between one of the agents and the switch is less than 1.5, a reward of 0 otherwise. The second environment, Env. (**ii**), was the MaMuJoCo *reacher* environment Peng et al. [2021]. In this environment, two agents operate the two linked joints and each space dimension is discretized over 10 bins. Per-agent policies were conditioned on their respective joint angles only. For Sparse-Rewards Tasks, the goal was set to be randomly at the worst case, namely on position $(\pm 0.21, \pm 0.21)$ on the boundary of the reachable area. Reaching the goal mean to have a tip position (not observable by the agents and not discretized) at a distance less that 0.05 and provides a positive reward to both the agents of 1 when reached, a reward of 0 otherwise.

Hyperparameters

Policies. In Environment (**i**), the policy was defined by a dense $(64, 64)$ NN. This network accepts per-agent state features as input and produces action vector probabilities through a final soft-max layer. In Environment (**ii**) instead, the policy is represented by a Gaussian characterized by a diagonal covariance matrix. It receives features of the state as input and outputs action vectors. The mean is conditioned on the state and is the terminal output of a dense $(64, 64)$ NN. The standard deviation is unconditioned, represented by a distinct trainable vector, and is initialised to -0.5. The weights are initialised using Xavier Initialization.

TRPE. A dataset of N trajectories is collected in each epoch over a T-lenght horizon (see Algorithm 5.3), leading to the reported number of samples. Throughout the experiment the number of epochs e were set equal to $e = 100$, the number of trajectories $N = 10$, the KL threshold $\delta = 6$, the maximum number of off-policy iterations set to $n_{\text{off,iter}} = 20$, the learning rate was set to $\eta = 10^{-5}$ and the number of seeds set equal to 4 due to the inherent low stochasticity of the environment.

Multi-Agent TRPO. We adopted the notation from Duan et al. [2016]. Agents have independent critics $(64, 64)$ Dense networks and in each epoch a dataset of N trajectories is gathered for a given exploration horizon T for each agent, leading to the reported number of samples. Throughout the experiment the number of epochs e were set equal to $e = 100$, the number of trajectories building the batch size $N = 20$, the KL threshold $\delta = 10^{-4}$, the maximum number of off-policy iterations set to $n_{\text{off,iter}} = 20$, the discount was set to $\gamma = 0.99$.

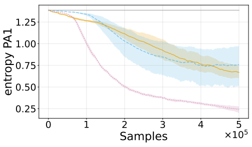


Figure C.22: *Entropy of Agent 1 Policy in TRPE Training (**i**), $T = 50$).*

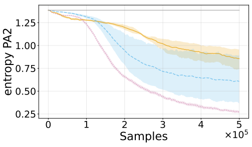


Figure C.23: *Entropy of Agent 2 Policy in TRPE Training (**i**), $T = 50$).*

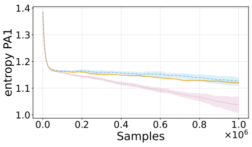


Figure C.24: *Entropy of Agent 1 Policy in TRPE Training (**ii**), $T = 100$).*

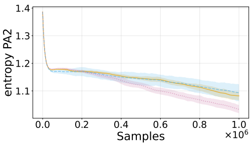


Figure C.25: *Entropy of Agent 2 Policy in TRPE Training (**ii**), $T = 100$).*

*Policiy Entropy Insights for TRPO Pretraining in Env (**i**) and Env (**ii**). **Lower Entropic Policies with Disjoint Objectives might justify the difference in pre-training performance even if the performances in training are similar**.*

Full Experimentation

We complement the main findings with two additional sets of figures that offer deeper insight into the mechanisms underlying the effectiveness of different exploration objectives.

Correlation Between Objective Type and Policy Entropy. The first set of figures demonstrates that the performance differences observed across joint, disjoint, and mixture objectives are tightly coupled with their ability to foster deterministic versus stochastic behaviors. Specifically, we observe that disjoint objectives often result in collapsed, nearly deterministic policies that fail to support effective coordinated exploration. In contrast, mixture objectives promote more diverse and stochastic behaviors across agents, enabling coverage of wider regions of the state space and better alignment with the theoretical goals of entropy maximization.

Extended Pre-Training Results Across Exploration Horizon Regimes. The second group of figures reports the full pre-training performance curves across different efficiency regimes of exploration horizons. These include both short-horizon (e.g., $T = 50$) and long-horizon (e.g., $T = 150$) setups, highlighting how mixture objectives consistently lead to better downstream fine-tuning or zero-shot performance. Additionally, we include pre-training plots for the high-dimensional MaMuJoCo Reacher environment. Even in this more complex setting, mixture-pretrained policies exhibit faster adaptation and higher entropic policies compared to other methods, supporting the robustness of our approach.

These additional visualizations further validate the theoretical claims and emphasize the practical relevance of mixture-based exploration in multi-agent task-agnostic settings.

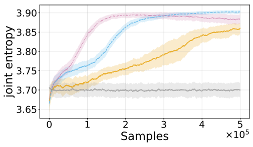


Figure C.26: *TRPE Joint Entropy (Env. (**i**), $T = 50$).*

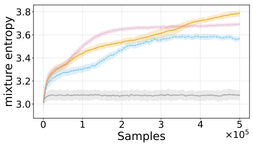


Figure C.27: *TRPE Mixture Entropy (Env. (**i**), $T = 50$).*

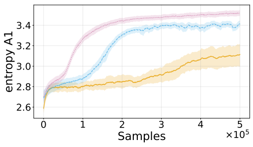


Figure C.28: *TRPE Entropy Agent 1 (Env. (**i**), $T = 50$).*

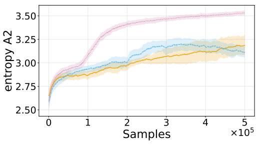


Figure C.29: *TRPE Entropy Agent 2 (Env. (**i**), $T = 50$).*

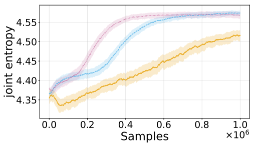


Figure C.30: *TRPE Joint Entropy (Env. (**i**), $T = 100$).*

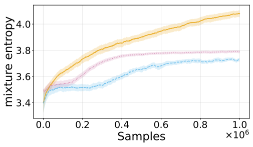


Figure C.31: *TRPE Mixture Entropy (Env. (**i**), $T = 100$).*

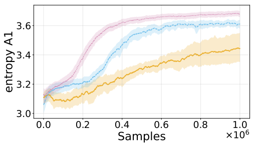


Figure C.32: *TRPE Entropy Agent 1 (Env. (**i**), $T = 100$).*

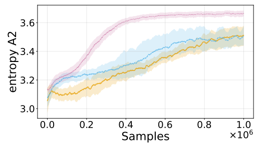


Figure C.33: *TRPE Entropy Agent 2 (Env. (**i**), $T = 100$).*

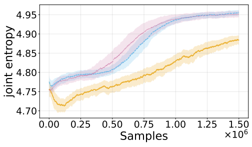


Figure C.34: *TRPE Joint Entropy (Env. (**i**), $T = 150$).*

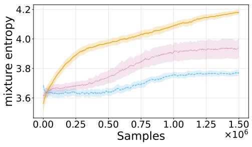


Figure C.35: *TRPE Mixture Entropy (Env. (**i**), $T = 150$).*

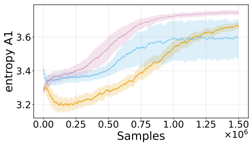


Figure C.36: *TRPE Entropy Agent 1 (Env. (**i**), $T = 150$).*

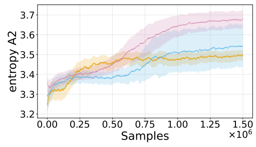


Figure C.37: *TRPE Entropy Agent 2 (Env. (**i**), $T = 150$).*

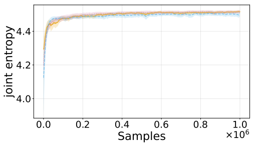


Figure C.38: *TRPE Joint Entropy (Env. (**ii**), $T = 100$).*

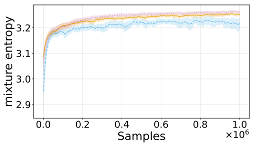


Figure C.39: *TRPE Mixture Entropy (Env. (**ii**), $T = 100$).*

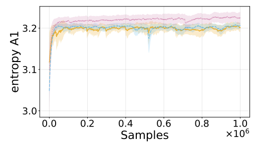


Figure C.40: *TRPE Entropy Agent 1 (Env. (**ii**), $T = 100$).*

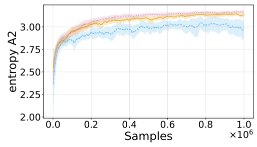


Figure C.41: *TRPE Entropy Agent 2 (Env. (**ii**), $T = 100$).*

Full Visualization of Policy Pre-Training.