POLITECNICO DI MILANO

School of Industrial and Information Engineering
Department of Electronics, Information and Bioengineering
Master of Science Degree in Computer Science and Engineering

Optimal Policy Evaluation for Policy Optimization

AI & R Lab
The Artificial Intelligence and Robotics Lab
of the Politecnico di Milano

Supervisor: Prof. Marcello Restelli
Co-supervisor: Dott. Alberto Maria Metelli

Author:
Samuele Meta, 926351

Academic Year 2020-2021

A chi mi ha sostenuto, spesso credendo in me
più di quanto non lo abbia fatto io.

Abstract

Off-policy methods are the basis of a large number of effective Policy Optimization algorithms. In this setting, Importance Sampling is typically employed as a what-if analysis tool, with the goal of estimating the performance of a target policy, given samples collected with a different behavioral policy. However, in Monte Carlo simulation, Importance Sampling represents a variance minimization approach. In this field, a suitable behavioral distribution is employed for sampling, allowing diminishing the variance of the estimator below the one achievable when sampling from the target distribution. In this thesis, Importance Sampling is analyzed in these two guises, showing the connections between the two objectives. It is shown that variance minimization can be used as a performance improvement tool, with the advantage, compared with direct off-policy learning, of implicitly enforcing a trust region. These theoretical findings are used to build a novel Policy Optimization algorithm, Policy Optimization via Optimal Policy Evaluation (PO^2PE), that employs variance minimization as an inner loop. Finally, empirical evaluations on continuous Reinforcement Learning benchmarks are presented, with a particular focus on the robustness to small batch sizes.

Estratto in lingua italiana

Il controllo automatico, la robotica, i giochi e la finanza sono solo alcuni dei problemi che possono essere modellati come un processo decisionale sequenziale, caratterizzato da un *agente* che interagisce con l'*ambiente*.
Negli ultimi anni, l'*Apprendimento per Rinforzo* [Sutton and Barto, 1998] ha riscontrato notevoli sviluppi e, al momento, rappresenta l'approccio più promettente per questa categoria di problemi. Per risolverli, ovvero per individuare il comportamento ottimale dell'agente nell'ambiente, l'apprendimento per rinforzo ricalca la modalità di apprendimento degli esseri viventi, principalmente incentrata sull'esperienza diretta. Pertanto, senza la necessità di alcun insegnante al di fuori di essa, l'apprendimento per rinforzo permette agli agenti di acquisire una competenza simulandola numerose volte. Per tradurre questa intuizione in un algoritmo, il problema è spesso espresso come un processo decisionale di Markov [Szepesvári, 2010], un formalismo matematico sufficientemente semplice e astratto da adattarsi a diversi scenari. Esso consente di descrivere rigorosamente il problema, scomponendolo nei concetti di *stato*, *azione*, *ricompensa* e *dinamica dell'ambiente*. Pertanto, lo scopo dell'agente può essere tradotto nel tentativo di raccogliere la maggior ricompensa possibile. Ciò basta ad introdurre una prima, ma efficace, classe di metodi risolutivi, nota come Programmazione Dinamica [Bellman, 1957]. Mediante questa, non è difficile determinare la soluzione ottima di un processo decisionale di Markov. Tuttavia, la programmazione dinamica richiede che la dinamica dell'ambiente sia nota, un requisito che, sfortunatamente, non è assicurabile in molti problemi reali. Ciononostante, come accennato in precedenza, la dinamica dell'ambiente può essere appresa attraverso un'interazione diretta tra l'agente e l'ambiente stesso. Nel corso degli anni, la letteratura si è estensivamente occupata delle tecniche di apprendimento per rinforzo e algoritmi sempre più efficienti sono stati proposti. Parte di questi sfrutta una funzione di valore [Sutton and Barto, 1998], uno strumento matematico che permette di stimare la convenienza di trovarsi in un certo stato ed eseguire una determinata azione. Dal momento

che l'obiettivo dell'agente è individuare le azioni più remunerative, tale funzione può guidarlo nello scoprire la politica da adottare. Perdipiù, nulla vieta all'agente di ricercare tale politica nello spazio di tutte quelle possibili, senza servirsi di una funzione di valore. Questa idea è alle fondamenta dei metodi di Ottimizzazione della Politica [Deisenroth et al., 2013], attualmente molto in voga per le numerose e desiderabili proprietà teoriche e per la loro applicabilità in scenari molto complessi. Per rendere la risoluzione di tale problema di ottimizzazione computazionalmente possibile, spesso lo spazio delle politiche viene parametrizzato. Al crescere dell'esperienza maturata dall'agente nell'ambiente, i parametri verrano modificati per indurre una politica sempre migliore. Quando la variazione dei parametri viene effettuata con l'accortezza di non allontarsi eccessivamente dal loro valore originale, si parla di algoritmi *a regione di confidenza* [Conn et al., 2000]. Tuttavia, indipendentemente dal metodo risolutivo applicato, tutti i problemi di apprendimento per rinforzo condividono la necessità di sfruttare al massimo delle possibilità gli episodi della propria esperienza. Per conseguire questo scopo, è scelta comune affidarsi all'apprendimento *off-policy*. Questo consente di utilizzare l'esperienza derivante dal seguire una politica, detta *comportamentale*, per poter stimare la performance di un'altra politica, detta *obiettivo*. Dal momento che l'esperienza proviene da una politica diversa, uno stimatore *Importance Sampling* [Owen, 2013] si rivela indispensabile. Mediante questa tecnica statistica, infatti, gli episodi generati dalla politica comportamentale vengono scalati proporzionalmente alla probabilità di essere generati da quella obiettivo. Pertanto, in un contesto off-policy, l'importance sampling assume un ruolo passivo. Eppure, in principio, esso è nato come uno strumento matematico dal ruolo attivo nel ridurre la varianza dello stimatore. In aggiunta, il problema che si prefiggeva di risolvere era opposto: fissata una politica obiettivo, lo scopo era individuare quella comportamentale che inducesse una stima a varianza minima. In tali veci, l'importance sampling è stato applicato nell'apprendimento per rinforzo soprattutto per trattare eventi rari [Frank et al., 2008; Ciosek and Whiteson, 2017], che di per sè generano un'alta varianza. Inoltre, l'intuizione di impiegare esplicitamente l'importance sampling come una tecnica di riduzione della varianza, in combinazione ad algoritmi di ottimizzazione della politica, è già stata vagliata in letteratura, ma trattando i due problemi separatamente [Hanna et al., 2017; Hanna and Stone, 2018; Hanna et al., 2019].
Lo scopo di questa tesi è approfondire la relazione che sussiste tra la riduzione della varianza e l'apprendimento off-policy, cercando di dare risposta al seguente quesito: *"Può la minimizzazione della varianza essere usata come uno strumento per l'apprendimento off-policy, rendendo vano l'impiego di*

una regione di confidenza esplicita?". Intuitivamente, assegnando maggiore probabilità agli episodi che portano ingenti ricompense, la varianza dello stimatore importance sampling può essere ridotta, dal momento che tali episodi hanno un impatto rilevante sulla media. Nel corso della trattazione, le proprietà teoriche di questa relazione verranno esaminate, con particolare enfasi sulla loro traduzione in un nuovo algoritmo di ottimizzazione della politica.

Contributi

I contributi di questa tesi sono teorici, algoritmici e sperimentali. Dopo aver introdotto le nozioni propedeutiche e aver esaminato lo stato dell'arte, la distribuzione comportamentale a minima varianza è definita. Le proprietà del processo di minimizzazione della varianza sono esaminate in due contesti, prima non applicando restrizioni sullo spazio delle distribuzioni e successivamente aggiungendole. In entrambi i casi, viene garantito un miglioramento della performance e viene determinata implicitamente una regione di confidenza. Forte di queste fondamenta teoriche, un nuovo algoritmo di ottimizzazione della politica, *Policy Optimizazion via Optimal Policy Evaluation* (PO^2PE), è presentato. Esso consiste in un ciclo interno, preposto alla valutazione della performance della politica, innestato in uno esterno, dedicato alla sua ottimizzazione. La valutazione sperimentale propone un confronto con altri algoritmi a regione di confidenza, quali POIS [Metelli et al., 2018] e TRPO [Schulman et al., 2015]. Dalle simulazioni empiriche emerge che PO^2PE è capace di migliorare i risultati delle baseline, mostrando una notevole stabilità su un grande numero di scenari.

Struttura della tesi

I contenuti della tesi sono organizzati nei seguenti sei capitoli. Il Capitolo 1 introduce il contesto, le motivazioni e i contributi della ricerca svolta. Il Capitolo 2 illustra i concetti propedeutici alla trattazione successiva, inquadrando l'apprendimento per rinforzo e i processi decisionali di Markov, fornendone le principali definizioni e discutendone le relative proprietà. Per giustificare la necessità di tecniche più avanzate, la programmazione dinamica è presentata e le sue maggiori limitazioni sono messe in luce. Infine, un'ampia discussione circa i metodi di ottimizzazione della politica consente al lettore di prendere confidenza con i fondamenti dell'argomento.
Il Capitolo 3 arricchisce ulteriormente i contenuti precedenti, focalizzandosi sullo stato dell'arte nella ottimizzazione della policy e, in particolar modo, sulle tecniche basate sulla regione di confidenza. Una concisa panoramica

delle più rilevanti, quali REPS [Peters et al., 2010], TRPO [Schulman et al., 2015], PPO [Schulman et al., 2017] e POIS [Metelli et al., 2018], permette di delineare un metro di paragone utile nei capitoli seguenti. Per poterne realmente cogliere l'essenza, tuttavia, una digressione sulla tecnica statistica dell'importance sampling è indispensabile.

Il Capitolo 4 raccoglie i contributi cardine della tesi, incentrati sull'analisi della distribuzione comportamentale a minima varianza. In primo luogo, il problema viene affrontato supponendo uno spazio delle distribuzioni non limitato, per poi adattare le considerazioni alla realistica eventualità di un contesto con restrizioni. Nella sezione finale, il pseudocodice di PO^{2}PE viene derivato e la sua logica è spiegata accuratamente.

Il Capitolo 5 è dedicato alla valutazione sperimentale di PO^{2}PE, mediante un confronto diretto con altri algoritmi basati sulla regione di confidenza, quali POIS e TRPO. La discussione sull'esito delle simulazioni, accompagnata da numerosi grafici, è preceduta dalle principali assunzioni e dalle scelte implementative. In aggiunta alle verifiche sui classici scenari dell'apprendimento per rinforzo, ulteriori test gettano luce sulle proprietà teoriche dell'algoritmo.

Il Capitolo 6 riassume i risultati ottenuti, evidenziandone i punti di forza e le maggiori criticità. Infine, fornisce indicazioni per possibili estensioni e sviluppi futuri.

L'Appendice A riporta le derivazioni e dimostrazioni omesse dalla discussione principale, mentre l'Appendice B contiene tutti i dettagli implementativi della fase sperimentale, quali l'infrastruttura utilizzata e il processo di ricerca degli iperparametri.

Ringraziamenti

Innanzitutto, un primo, doveroso, ringraziamento è da rivolgersi al Prof. Marcello Restelli, responsabile di avermi proposto un elaborato di notevole interesse, nonchè di avermi sapientemente guidato in questo percorso. Desidero inoltre ringraziare il mio correlatore, il Dott. Alberto Maria Metelli, per tutto il tempo dedicatomi e il giusto connubio tra professionalità e cordialità che lo ha sempre contraddistinto. Grazie alla loro rara passione e disponibilità, non solo potrò considerare questa esperienza un momento di crescita professionale, ma anche un bellissimo ricordo a coronare il mio percorso accademico.

Giunto a questo punto, guardandomi indietro, non posso che ripetermi una volta ancora che non sarò mai abbastanza grato ai miei genitori per i loro sacrifici, senza i quali sicuramente non mi troverei dove sono adesso. Grazie per non avermi mai fatto mancare il vostro supporto incondizionato, nonostante la distanza. Grazie per aver creduto nel mio successo in questa avventura al Politecnico, di cui siete stati i primi promotori. Naturalmente, un pensiero va anche a mio fratello, la cui positività ha sempre rallegrato i miei rientri a Firenze.

Infine, ci tengo a ringraziare tutti coloro che ho incrociato durante il mio percorso, anche se per sbaglio, anche se solo per un caffè. Grazie a chi ha condiviso un tavolo per studiare e a chi mi ha accompagnato nell'aula di un esame. Grazie a chi ha passato con me le serate in cucina dopo aver mangiato e a chi si è prestato a chiacchierate infinite. Grazie a chi si è preoccupato nei momenti difficili e a chi mi ha preso in giro in quelli spensierati. Grazie a chi c'è stato e a chi c'è tuttora. Forse senza di voi il risultato sarebbe stato lo stesso, ma avrebbe avuto tutto un altro sapore.

Mathematical Notation

Reinforcement Learning is an extremely fascinating research domain, which combines mathematics with algorithms. However, this pair is often the cause of a very heterogeneous notation across different works. Indeed, finding the right balance between the formalism and the raw concept is not always easy. Since this thesis is no exception, in the following, we report the recurrent mathematical notation and we clarify the intended usage of some symbols.

Let $\mathcal{X}$ be a set, and let $\mathfrak{F}_{\mathcal{X}}$ be a σ-algebra over $\mathcal{X}$. We indicate with $\mathscr{P}(\mathcal{X})$ the space of probability measures over $(\mathcal{X}, \mathfrak{F}_{\mathcal{X}})$. Given $P \in \mathscr{P}(\mathcal{X})$, we express with lowercase p its density function. This rule presents two exceptions throughout the thesis: the probability density function of the transition model and of the reward function of an MDP. They are respectively referred with P and R. For a subset $\mathcal{Y} \subseteq \mathbb{R}$, we denote with $\mathscr{B}(\mathcal{X}, \mathcal{Y})$ the space of measurable functions $f : \mathcal{X} \to \mathcal{Y}$. Dealing with value functions, the superscript π in V^{π} and Q^{π} specifies that, in the computation, policy π is adopted. The expected value of a function $f(x)$, taken with respect to a random variable x distributed according to the probability density p, is indicated as $\mathbb{E}_{x \sim p}[f(x)]$. Whenever there is no ambiguity, the probability density may be omitted. In the context of parametrized policies, performance $J_{\boldsymbol{\theta}}$ is short for $J(\boldsymbol{\theta})$, which in turn represents $J(\pi_{\boldsymbol{\theta}})$. Similarly, the distribution of trajectories under policy π, i.e. $p_{\pi_{\boldsymbol{\theta}}}(\tau)$, is often replaced by $p_{\boldsymbol{\theta}}$. The gradient of function f with respect to a vector of independent variables $\mathbf{x}$ is indicated with $\nabla_{\mathbf{x}} f(\mathbf{x})$. Operators are denoted by calligraphic letters, e.g. $\mathcal{I}$, and their application is indicated by square brackets, i.e. $\mathcal{I}[\cdot]$.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Automation control, robotics, games and finance are only a few of the real-world problems that can be modeled as a sequential decision making process, characterized by an *agent* interacting with the *environment*. Reinforcement Learning (RL) [Sutton and Barto, 1998] has seen significant advances in the last decades and, currently, represents the most promising approach to this class of problems. To solve them, i.e. to find the optimal behavior of the agent in the environment, RL mimes the learning process of living beings, mainly based on direct interaction with the task. Therefore, without the need for any teacher besides experience, RL agents can acquire a large variety of skills in a trial and error fashion. To translate this intuition into an algorithm, the problem is often stated in terms of a Markov Decision Process (MDP) [Szepesvári, 2010], a mathematical framework sufficiently simple and abstract to be applied to different scenarios. Such formalism allows to rigorously describe the problem, breaking it down into the concepts of *state*, *action*, *reward* and *dynamics of the environment*. This is sufficient to introduce a first, yet powerful, class of techniques, named Dynamic Programming (DP) [Bellman, 1957]. Through their application, it is possible to easily find the optimal solution of an MDP. However, they heavily rely on the assumption that the environment dynamics are known. Unfortunately, in real-life settings, this requirement often turns out to be too tight. Nevertheless, this does not represent an insurmountable hurdle because, as mentioned above, environment dynamics can still be learned through the interaction between the agent and the environment itself. Literature has extensively dealt with RL methods, proposing, year after year, novel and effective algorithms, enriching the wide taxonomy that classifies them. A first category is composed of the so-called *value-based* approaches, which leverage the outcome of a utility function. Through this mathematical tool, the agent can estimate

the "goodness", expressed in terms of future rewards, of being in a certain state and performing a certain action. Aiming to maximize the overall reward, the agent can exploit such values to determine its optimal behavior, usually referred to as *policy*. However, nothing prevents the agent from skipping this intermediate step, directly seeking the optimal policy in the policy space. Such *policy-based* vision is embraced by Policy Optimization (PO) [Deisenroth et al., 2013], a branch of Policy Search (PS) [Deisenroth et al., 2013]. Currently, it represents the most promising approach to real-world RL problems, due to the numerous desirable theoretical properties and to the ability to deal with continuous and high-dimensional scenarios. Instead of defining a utility function, PO methods describe the policy through a set of parameters, which are repeatedly updated to find the best induced policy. Nevertheless, value-based and policy-based algorithms, despite the consistent differences, both share a core problem of any RL technique, that is employing the collected samples as efficiently as possible. To achieve this result, *off-policy* learning is a common choice. Indeed, it is based on the idea of evaluating the performance of a *target* policy, while the samples are collected with a *behavioral* one. Since this introduces a mismatch, often the solution consists in adopting an Importance Sampling (IS) [Owen, 2013] estimator, which reweights each sample proportionally to the likelihood of being generated by the target policy.

Motivation

As previously mentioned, PO methods have been widely exploited in RL, proving to be a successful approach in addressing a variety of problems, such as continuous-control [Peters and Schaal, 2008; Lillicrap et al., 2016], robot manipulation [Gu et al., 2017; Chatzilygeroudis et al., 2020] and locomotion [Kohl and Stone, 2004; Duan et al., 2016]. Most of these algorithms exploit the notion of *trust region* [Conn et al., 2000], which allows seeking the successive policy "sufficiently close" to the current one. As a consequence, the PS process can rely on an accurate estimate of the next policy, limiting the variance of the IS estimator. However, in this off-policy learning (Off-PL) setting, IS is employed as a *what-if* analysis tool [Owen, 2013] and has a *passive* role, as samples have already been collected with the current behavioral policy. In this sense, the trust region is an *a-posteriori* remedy for the uncertainty injected by the IS procedure. Nevertheless, IS originated in the Monte Carlo simulation community as an *active* tool for variance minimization (Off-VM). In principle, given a fixed target policy, the goal was finding the behavioral policy yielding a minimum-variance IS estimate. In

this role, IS has already been employed in RL, mainly addressing rare events [Frank et al., 2008; Ciosek and Whiteson, 2017] which lead to high-variance estimates. Moreover, the idea of explicitly using IS as a variance reduction technique, to find the optimal behavioral policy, has been already proposed and combined with PS [Hanna et al., 2017; Hanna and Stone, 2018; Hanna et al., 2019]. However, in these works, the variance minimization (Off-VM) process and the off-policy learning (Off-PL) problem are treated separately.

Goal

The goal of this thesis is to investigate the relation between variance minimization and off-policy learning, aiming to answer the following question: *"Can Off-VM be employed as a tool for Off-PL, overcoming the need for an explicit trust region?"*. Intuitively, under the assumption of a positive reward function, the variance of the IS estimator can be reduced by assigning a larger probability to the episodes of experience with high returns, since they have a consistent impact on the mean. This provides a first hint about the connection between Off-VM and the Off-PL. Moreover, it suggests that the repeated identification of the minimum-variance policy can be employed as a tool for policy improvement. This thesis is focused on an in-depth study of the theoretical properties of this connection, with particular emphasis on how they can be exploited to define a novel PO algorithm.

Contribution

The contribution of this thesis is theoretical, algorithmic and experimental. After having introduced the necessary background and having revised the state of the art, the minimum-variance behavioral distribution is presented. The properties of the Off-VM problem are discussed in two settings: unconstrained and constrained. First, under the assumption of no restriction in the choice of the behavioral distribution, it will be shown that a performance improvement is guaranteed and an implicit trust region is enforced. Then, these results are revalued and generalized considering the constrained scenario, in which the available distributions are limited in a suitable space. Nevertheless, it is proved that the procedure still leads to a performance improvement and preserves the ability to enforce a trust region. Based on these theoretical results, a novel PO algorithm, *Policy Optimization via Optimal Policy Evaluation* (PO^{2}PE), is proposed. It leverages the intuition of repeatedly applying Off-VM as a tool for Off-PL. Therefore, it consists of an inner loop, that aims at performing the *evaluation* phase, and an outer loop, that is devoted to *optimization*. The experimental evaluation

proposes a comparison with other trust-region algorithms, such as POIS [Metelli et al., 2018] and TRPO [Schulman et al., 2015]. From the empirical simulations, it appears that PO^{2}PE outperforms the baselines in almost all the benchmarks, displaying remarkable stability across different runs.

Outline

The contents of this thesis are organized in the following five chapters. Chapter 2 introduces the RL framework and the MDP formalism, providing definitions of the main concepts and discussing their properties. To better motivate the need for more advanced approaches, DP is briefly presented and its limitations are commented. Finally, a wide discussion about PO guides the reader in revising the key intuitions behind this topic, illustrating some of the most important techniques in the field.
Chapter 3 enriches the previous contents focusing on state-of-the-art PO algorithms, which are part of the trust-region family. A concise overview of REPS [Peters et al., 2010], TRPO [Schulman et al., 2015], PPO [Schulman et al., 2017] and POIS [Metelli et al., 2018] allows us to highlight their peculiarities and to later compare them with respect to PO^{2}PE. This section is preceded by a detailed review of the IS estimator, an essential statistical tool for characterizing the aforementioned approaches.
Chapter 4 collects the core contributions of this thesis, based on the analysis of the minimum-variance behavioral distribution. At first, the problem is addressed assuming an unconstrained distribution space, then the previous considerations are adapted to the realistic eventuality of a constrained scenario. In the final section, the pseudocode of PO^{2}PE is derived and its rationale is explained in detail.
In Chapter 5, PO^{2}PE is evaluated against two relevant trust-region algorithms, namely POIS and TRPO. Before discussing the achieved results, depicted in numerous plots, the experimental setting is presented, clarifying the main assumptions and design choices. In addition to the classical benchmarks, further simulations shed light on the effects of the theoretical properties.
In Chapter 6, the results achieved along this thesis are summarised, highlighting the points of strength and limitations of the proposed approach. Finally, possible future developments and extensions of this work are suggested.
Appendix A reports the proofs and derivations omitted in the text, while Appendix B contains all the low-level experimental details, such as information about the computational infrastructure and the tuned hyperparameters.

Chapter 2

Preliminaries

In the context of Machine Learning (ML), the phrase *learning* can take multiple different meanings. In Supervised Learning, this process is carried out by analyzing a training set of labeled examples, while finding out a hidden structure in unlabeled data is the purpose of Unsupervised Learning. However, none of these approaches, even considering their combination, is suitable for describing the learning process we are most familiar with, namely that of living beings. When children learn to ride a bike, the common strategy will not involve the study of any labeled or unlabeled data, but a direct experience of the task. They will make the first rides without any prior knowledge of how to properly balance their weight, how to use the handle and how hard the brakes are. Therefore, it is very likely that the first time they will fall after few seconds, but feedback of what was profitable and what was not will be provided. Attempt after attempt, according to a *trial and error* approach, they will better understand the causal relation between actions and subsequent effects, which will allow them to adjust their behavior. Eventually, at the price of some scrapes, they will achieve the goal of learning how to correctly ride the bike, without the need for any teacher besides experience. This goal-directed process of learning from interaction can be generalized and formalized by *Reinforcement Learning* (RL) [Sutton and Barto, 1998]. In the RL framework (Figure 2.1), the typical problem can be expressed in terms of an *agent* (the child) acting in an *environment* (the world, i.e. everything outside the agent) to accomplish a *goal* (learning to ride the bike). Agent and environment interact continually, the agent making decisions and the environment responding to these actions and presenting its new representations (*states*) to the agent. The decision making process is guided by the *rewards*, i.e. the feedback provided by the environment. Intuitively, if the child distributes weight properly and

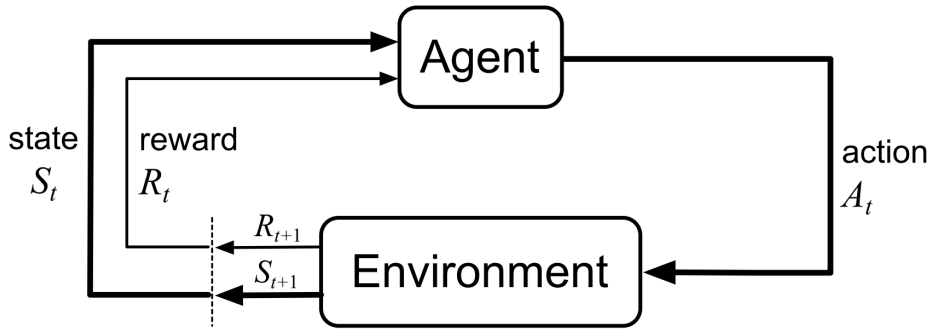

Figure 2.1: The Reinforcement Learning framework [Sutton and Barto, 1998]. At each time step t, the agent interacts with the environment, collects a reward and transitions to a new state.

maintains balance, the consequent reward will be positive, since these actions will prevent falling. Instead, an excessively sharp braking will result in a negative reward, as it can seriously jeopardize the stability of the bike. Therefore, to achieve the goal, the child will learn to behave in order to maximize the positive rewards, minimizing the negative ones. Finally, as can be experimented in many other real-life tasks, actions may not only affect the immediate future, but they may also produce a domino effect, due to modification of the environment.

Chapter Outline

The chapter is organized as follows. Section 2.1 introduces Markov Decision Processes, a mathematical framework for modeling sequential decision making problems. Section 2.2 shows how goals can be expressed in terms of rewards. Section 2.3 and Section 2.4 are devoted to the definition of the concept of policy and relative value functions. Section 2.5 explains how to solve a Markov Decision Process in basic scenarios, while how to overcome this limitation through RL is the core of Section 2.6. Section 2.7 presents Policy Search methods, whose properties are better analyzed in Section 2.8 and Section 2.9. Finally, Section 2.10 reports two successful algorithmic applications of Policy Search.

2.1 Markov Decision Processes

A Markov Decision Process (MDP) is a mathematical framework for modeling sequential decision making problems, sufficiently simple and abstract to be applied to a variety of scenarios, such as economics [Rust, 1996], manufacturing [Feyzabadi and Carpin, 2014] and robotics [Cassandra et al., 1996]. This has led to the formulation of multiple definitions, even narrowing them down to the RL domain [Sutton and Barto, 1998; Szepesvári, 2010; Puterman, 2014]. Mainly taking inspiration from [Puterman, 2014] and [Sutton and Barto, 1998], an MDP can be formally defined as follows:

Definition 2.1.1 (Markov Decision Process)**.** A Markov Decision Process is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu_0)$, where:

- $\mathcal{S}$ is a measurable set of all possible states in which the agent can be, named *state space*;
- $\mathcal{A}$ is a measurable set of all possible actions in any state, named *action space*;
- $P : \mathcal{S} \times \mathcal{A} \to \mathscr{P}(\mathcal{S})$ is the *transition model*, which assigns a probability measure over $\mathcal{S}$ for each (s, a) pair. In other terms, for each state $s \in \mathcal{S}$ and for each performed action $a \in \mathcal{A}$, $P(s'|s, a)$ represents the probability (in the sense of a density) of moving to state $s' \in \mathcal{S}$, if action a is played in state s;
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathscr{P}(\mathbb{R})$ is the *reward model*, which assigns a probability measure $R(\cdot|s, a, s')$ for each triple composed by the state s, the performed action a and the next state s', such that $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. With little abuse of notation, the expected immediate reward can be defined as $r(s, a) = \mathbb{E}_{s' \sim P(\cdot|s,a), r \sim R(\cdot|s,a,s')}[r]$;
- $\gamma \in [0, 1]$ is the *discount factor*, which determines the present value of future rewards. If $\gamma = 0$, the agent is "myopic" in being concerned only with maximizing immediate rewards. Instead, as γ approaches 1, the future rewards will be taken into account more strongly, making the agent more and more "farsighted";
- $\mu_0 \in \mathscr{P}(\mathcal{S})$ is the *starting state distribution*, from which the initial state is sampled.

Building on these elements, it is possible to fully describe the aforementioned generic RL scenario. According to the starting state distribution μ_0,

the initial state $s_0 \in \mathcal{S}$ will be drawn. Following some criteria, which will be better discussed in Section 2.3, the agent will choose an action a_0, among all the possible ones in state s_0. Playing this action will determine the next state $s_1 \sim P(\cdot|s_0, a_0)$, according to the transition model, and an immediate reward $r_0 \sim R(\cdot|s_0, a_0, s_1)$, dependent on the reward model. What has been characterized is a single iteration between agent and environment, occurring at time instant $t = 0$. However, this analysis can easily be extended to any t, repeating these simple steps. Without introducing any additional element, nothing prevents this process from being iterated an indefinite number of times. In such a case, we are dealing with *infinite-horizon* MDPs. Besides bounding the number of decision epochs to a fixed T (*finite-horizon*), another cause of a finite number of steps may be due to the presence of a terminal state, i.e. such that no other state can be reached from it and all actions provide zero reward. If the MDP presents a terminal state reachable by the agent with non-zero probability, it is said to be *episodic*. In any case, the continuous interaction between agent and environment produces a sequence of states, actions and rewards, better known as *trajectory*:

$$\tau = (s_0, a_0, r_1, s_1, a_1, r_2, \ldots).$$

Since a trajectory contains all the visited states, all the performed actions and all the collected rewards by the agent, it can be seen as its history. This leads to the last, critical, assumption in dealing with an MDP, namely the homonymous Markov property, which establishes the dependence of the future solely on the present and not on the past. More rigorously, given a Markov Chain [Meyn and Tweedie, 1993], it holds that:

$$\mathbb{P}(X_{t+1} = z|X_t = y, X_{t-1} = y_{t-1}, \cdots, X_0 = y_0) = \mathbb{P}(X_{t+1} = z|X_t = y).$$

This means that X_t is a *sufficient statistic* for the future, capturing all the information from the history. In our framework, this reflects the fact that the agent will forget its past and only leverage the current state to select the subsequent action. As strong as this assumption may be, this does not limit its applicability, such as in the game of chess, where the state of the pawns on the board is sufficient to determine the next move, regardless of all the previous ones. Moreover, whenever such property is not verified, it is possible to address the problem as a markovian one, enriching the state space with information of the past. Finally, an MDP is said to be *stationary* if $\mathbb{P}(X_{t+1} = z|X_t = y)$ is the same at any t. Paraphrasing, stationarity

holds when the dynamics of the environment, i.e. P and R, do not generate different outcomes playing the same action in the same state, but in different time steps.

2.2 Goals and Rewards

Section 2.1 discussed the temporal evolution that leads the agent to transition, through actions, from one state to another and to collect rewards. Nevertheless, we still need to clarify what role the rewards play. A reward is a simple numeric signal, that characterizes the goodness of an action. Intuitively, negative rewards can be interpreted as punishments, while positive ones as prizes. It is therefore clear that the agent is interested in maximizing the total amount of reward it receives. Unfortunately, as we know from personal experience, this is not equivalent to maximizing the immediate reward, because the actions we take now can have a chain of repercussions on the future and, sometimes, sacrificing an immediate gain may prove to be the wisest choice. Hence, the agent will act in such a way as to collect the highest amount of reward in the long run. For [Sutton and Barto, 1998], maximizing the expected value of the cumulative sum of a received scalar signal is equivalent to achieving the goal of the agent. This conjecture is known as *Reward Hypothesis* and offers both a flexible and a widely applicable formalism to shape many problems. In view of this, we can use the concept of reward to formally state the MDP goal as:

$$\max_{a_0,a_1,\dots} \mathbb{E}\left[\sum_{t=0}^{\infty} r(s_t,a_t) \;\middle|\; s_0 \sim \mu_0, s_{t+1} \sim p(\cdot|s_t,a_t) \text{ for all } t>0\right].$$

This formulation, however, has a rather important drawback, related to infinite horizon MDPs, for which the series might not converge. To address this contingency, the most common choice is to resort to the *expected cumulative discounted reward* (also known as *expected discounted return*), through the introduction of the *discounting factor* γ. As already seen in Section 2.1, the values of γ will influence not only the convergence properties, but also the value of future rewards:

$$\max_{a_0,a_1,\dots} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \;\middle|\; s_0 \sim \mu_0, s_{t+1} \sim p(\cdot|s_t,a_t) \text{ for all } t>0\right]. \quad (2.1)$$

2.3 Policies

At each decision step, the agent has to decide, given the current state, how to interact with the environment in order to maximize the expected cumulative discounted reward. The mathematical formalization of such strategy, which maps states into actions, is called policy and determines the behavior of the agent.

Definition 2.3.1 (Policy)**.** A stationary stochastic policy is any function $\pi : \mathcal{S} \to \mathscr{P}(\mathcal{A})$ that assigns a probability distribution, for each $s \in \mathcal{S}$, over the action space $\mathcal{A}$. In other terms, given a state $s_t \in \mathcal{S}$, the next action will be $a_t \sim \pi(\cdot|s_t)$.

Stationarity, as previously mentioned, is related to the independence of time while choosing a certain action in a specific state. From this point forward in the discussion, the stationary prefix will be omitted whenever the condition will hold. A particular case is represented by deterministic policies, for which the definitions reduce to $\pi : \mathcal{S} \to \mathcal{A}$ and $a_t = \pi(s_t)$, since no randomness is involved. In both cases, the definition of the policy allows us to translate the optimization problem of Equation (2.1) in terms of finding an optimal policy.

2.4 Value Functions

The previous section showed how solving an MDP is equivalent to finding an optimal policy. A useful resource supporting the agent in the search for the optimal behavior, is the so-called *state value function* [Sutton and Barto, 1998], which provides the utility of being in a certain state. Since the utility is defined according to the future rewards that can be expected (more precisely, according to the expected return), the state value function is defined with respect to the policy.

Definition 2.4.1 (State Value Function)**.** Let $\mathcal{M}$ be an MDP and π be a policy. For every state $s \in \mathcal{S}$, the state value function $V^\pi : \mathcal{S} \to \mathbb{R}$ is defined as the expected return starting from state s and following policy π:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s \right], \tag{2.2}$$

where the operator $\mathbb{E}_\pi[\cdot]$ indicates that states and actions in the expected value are chosen according to policy π. The concept of state value function is central for the *prediction problem*, often cited as *policy evaluation* (PE), i.e.

the process of computing the performance of a policy in an MDP. However, it does not provide to the agent any information about the best action to select in a given state. For this reason, to address the *control problem* - whose purpose is *policy optimization* (PO) - we introduce the *action-value function*.

Definition 2.4.2 (Action Value Function)**.** Let $\mathcal{M}$ be an MDP and π be a policy. For every state-action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, the action value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as the expected return starting from state s, taking action a and following policy π:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right]. \tag{2.3}$$

By means of some mathematical manipulation, concerning the decomposition of the expected return in immediate reward and expected return in the following state, it is possible to provide a recursive formulation of both Equation (2.2) and (2.3), which go by the name of *Bellman Expectation Equations* (BEE):

$$V^\pi(s) = \mathop{\mathbb{E}}_{a \sim \pi(\cdot|s)} \left[r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} [V^\pi(s')] \right]. \tag{2.4}$$

For the sake of brevity, all formulas will be referred to V as just above, but it is possible to adapt them to Q with minimal effort. From the BEE, a closed-form solution of V and Q can be derived, with the disadvantage of a high computational cost. In order to face this problem, it is possible to rely on the iterative application of $T^\pi : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$, the *Bellman Expectation Operator* (BEO).

$$T^\pi(V)(s) = \mathop{\mathbb{E}}_{a \sim \pi(\cdot|s)} \left[r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} [V(s')] \right] \tag{2.5}$$

Finally, it is possible to determine the optimal values V^* and Q^*, therefore the optimal policy π^*, through another operator $T^* : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}}$, named *Bellman Optimality Operator* (BOO).

$$T^*(V)(s) = \max_{a \in \mathcal{A}} \left[r(s, a) + \gamma \mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} [V(s')] \right] \tag{2.6}$$

Since the above concepts do not play a central role in the development of this thesis, but are only propaedeutic to subsequent definitions, we refer the reader to [Bellman, 1957; Sutton and Barto, 1998] for an extensive review.

2.5 Dynamic Programming

Dynamic Programming (DP) [Bellman, 1957; Bellman and Dreyfus, 2015] is a general technique, not necessarily limited to the scope of RL, which is applied whenever a problem verifies the following properties:

1. possibility to be decomposed in *overlapping subproblems*, such that solutions to subproblems can be reused;
2. presence of an *optimal substructure*, i.e. the resolution of the subproblems implies the resolution of the problem.

Since the BEE provides a recursive decomposition and V (or Q) can store and reuse solutions, given a finite MDP one can use DP to find an optimal policy [Bertsekas and Tsitsiklis, 1996]. However, this class of algorithms is of limited utility in RL, because the assumption of full knowledge of the transition and reward models turns out to be unrealistic in most of the scenarios. Moreover, the great computational expense makes it problematic to scale to MDPs with a very large state space. Nevertheless, we will provide a brief description of the two most relevant DP algorithms, as they represent an essential foundation for the next methods.

2.5.1 Policy Iteration

Policy Iteration (PI) [Howard, 1960] is one of the first approaches to MDPs resolution. Its name comes from the idea of iterating between two phases, called *policy evaluation* and *policy improvement*, until convergence.

- In the first phase, for a given policy π (starting from a random one) its value function V^π will be evaluated. As already seen, this computation can be carried out through the BEE (Equation (2.4)) or through the repeated application of the BEO (Equation (2.5)), depending on the complexity of the problem. In the latter case, according to [Puterman and Shin, 1978], often full convergence is not necessary and a smaller number of applications may be sufficient. In the literature, this approach is known as *Modified Policy Iteration*.
- In the second phase, a new policy is greedily built by selecting, in each state, the action that looks best after one step of lookahead according to V^π.

Proofs of the monotonical improvement of successive policies and of the convergence of the algorithm in a finite number of steps can be found in [Puterman, 2014].

2.5.2 Value Iteration

Value Iteration (VI) [Bellman, 1957] is an algorithm based on the iterative application of the BOO (Equation (2.6)), starting from a random state value function. Once at convergence, the optimal policy is greedily retrieved, following the same criteria of PI. The most important difference between these two approaches lies in the fact that VI focuses on the state value function only, without explicitly representing the policy. As a consequence, intermediate state value functions may not correspond to any policy and single iterations of VI are less computationally demanding with respect to PI. On the other hand, VI tends to converge in a greater number of iterations. As before, proof of the convergence of the algorithm is provided in [Puterman, 2014].

2.6 Reinforcement Learning Algorithms

Introducing DP allowed to describe a straightforward class of algorithms to solve MDPs, under the assumptions already mentioned in Section 2.5. Unfortunately, the requirement of complete knowledge of the transition model P and of the reward model R often turns out to be too tight. In fact, in real-world problems, the environment dynamics can be represented by a very computationally expensive model or can even be completely unknown. These limitations are overcome by introducing RL algorithms, which can estimate, implicitly or explicitly, the transition and reward models through sampling, i.e. the continuous interaction between agent and environment. For example, SARSA [Rummery and Niranjan, 1994] and Q-Learning [Watkins and Dayan, 1992] exploit this idea and represent the RL version of PI and VI, respectively. Since RL algorithms will be the cornerstone topic of this thesis, we now introduce some nomenclature that will prove useful in future chapters.

2.6.1 Dichotomies

Model-based vs. Model-free

Although the transition model P and the reward model R are not available in principle, there is nothing to preclude the agent from estimating them, exploiting its own experience. Once a representation of the environment dynamics has been reconstructed from the samples, exact techniques such as DP can be applied to solve the MDP, if the complexity is not excessive. In general, different methods can be employed to derive an approximate

value function and/or approximately optimal policy. This is the core idea of *model-based* RL algorithms [Deisenroth and Rasmussen, 2011; Nagabandi et al., 2018; Wang et al., 2019]. Instead, in *model-free* RL algorithms [Mnih et al., 2015; Schulman et al., 2015; Lillicrap et al., 2016; Duan et al., 2016], the concept of model is totally absent, as the agent does not even try to build it from experience. In this case, samples become crucial for a direct estimate of the value function and/or the optimal policy.

On-policy vs. Off-policy

In *on-policy* settings [Williams, 1992; Rummery and Niranjan, 1994; Jaksch et al., 2010], there is a correspondence between the policy used to interact with the environment and the policy which is being learned. Instead, when a policy is used to collect samples and another one is optimized, we talk about *off-policy* methods. In this case, we refer to the former policy as *behavioral policy* and to the latter as *target policy*. Despite some caveats, better illustrated in Chapter 3, off-policy learning opens up major possibilities in RL, which are widely described in the core sections of this thesis.

On-line vs. Off-line

On-line algorithms [Watkins and Dayan, 1992; Jaksch et al., 2010; Schulman et al., 2017] interleave the learning and sample collection phases, performing them at the same time. As a consequence, the agent can update its estimates about the value function and/or policy as soon as new data are available. On the other hand, *off-line* algorithms [Ernst et al., 2005; Lange et al., 2012; D'Oro et al., 2020] leverage already collected samples, performing the learning phase only at a later stage.

Tabular vs. Function Approximation

In *tabular* RL approaches [Watkins and Dayan, 1992; Rummery and Niranjan, 1994], information about value functions can be represented and stored in tables, i.e. vectors of possibly different dimensionality such as arrays, matrices or tensors. It immediately follows that memory availability becomes a stringent requirement. Therefore, for problems with a high cardinality of $\mathcal{S}$ and $\mathcal{A}$ or in case of continuous spaces, one must turn to *function approximation* [Munos, 2005; Scherrer, 2014]. This last describes value functions and/or policies in a function space through various representations, such as linear regression, neural networks and decision trees.

Value-based vs. Policy-based vs. Actor-critic

Value-based algorithms [Watkins and Dayan, 1992; Rummery and Niranjan, 1994; Munos, 2005] are focused on estimating the optimal value function, in order to recover the optimal policy from it. Whereas, *policy-based* algorithms [Williams, 1992; Baxter and Bartlett, 2001; Pirotta et al., 2013a] bypass the first step and directly search the optimal policy. Finally, methods that learn approximations to both policy and value functions are named *actor-critic* [Sutton and Barto, 1998; Lillicrap et al., 2016], where "actor" is a reference to the learned policy and "critic" to the learned value function.

2.7 Policy Search

In the previous sections, we explained that solving an MDP is equivalent to finding the optimal policy π^* and some possible resolution techniques were introduced. All of them share one fundamental characteristic, namely the strict dependence on the value function. However, there exists a family of methods, known as Policy Search (PS) [Deisenroth et al., 2013], that, in contrast to value-based algorithms, typically avoids learning a value function, directly seeking the best policy in the policy space Π. Since the process involves mathematical optimization and this thesis will not address global search approaches, from now on, PS will also be referred to as Policy Optimization (PO) [Deisenroth et al., 2013]. To define a computationally feasible optimization problem, the policy space is described by restricted policy classes and, among them, *parametric policies* are of particular interest. Therefore, let $\Theta \subseteq \mathbb{R}^n$ be a parameter space for some $n \in \mathbb{N}$, the space of parametrized policies can be defined as:

$$\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^n\},$$

where the elements of Θ are n-dimensional real vectors, named *policy parameters*. It is evident that, changing the parameter space Θ, it is possible to explore the policy space. For this reason, many different parametrizations, both for deterministic and stochastic policies, have been proposed in the literature [Deisenroth et al., 2013; Sutton and Barto, 1998]. The most simple parametrization one could think of is represented by the linear one:

$$\pi_{\boldsymbol{\theta}}(s) = \boldsymbol{\theta}^T \phi(x),$$

where $\phi(x)$ is an appropriate *basis function* vector. This definition shows how the policy only depends linearly on the parameters, allowing to deter-

ministically draw an action $a = \pi_{\boldsymbol{\theta}}(s)$. However, specifying the right basis function by hand is generally a difficult task, often limiting their application domain. When a continuous d-dimensional action space $\mathcal{A} = \mathbb{R}^d$ is considered, the most common parametric policy is the multi-variate *Gaussian*:

$$\pi_{\boldsymbol{\theta}}(a|s) = \mathcal{N}(a|\mu_{\boldsymbol{\theta}}(s), \Sigma_{\boldsymbol{\theta}}(s)) = \frac{1}{\sqrt{2\pi|\Sigma_{\boldsymbol{\theta}}(s)|}} \exp\left(-\frac{\|a - \mu_{\boldsymbol{\theta}}(s)\|^2}{2\Sigma_{\boldsymbol{\theta}}(s)}\right),$$

where $\mu_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}^d$ is the mean, $\Sigma_{\boldsymbol{\theta}} : \Theta \times \mathcal{S} \to \mathbb{R}^{d \times d}$ is the covariance matrix and the operator $|\cdot|$ represents the determinant. Furthermore, the mean can be linearly parametrized as:

$$\mu_{\boldsymbol{\theta}}(s) = \boldsymbol{\theta}_\mu \phi(s),$$

and a diagonal covariance matrix is typically used, i.e. $\Sigma_{\boldsymbol{\theta}} = \mathrm{diag}(\exp(2\boldsymbol{\theta}_\Sigma))$. This latter choice is particularly convenient, because implies that each action variable follows an independent Gaussian distribution.

Taxonomy

Once a proper parametrization has been identified, the problem of finding the optimal policy $\pi_{\boldsymbol{\theta}^*}$ can be addressed through a large variety of approaches. To describe their taxonomy (Figure 2.2), a first important distinction between *model-free* and *model-based* algorithms has to be made. As presented in Section 2.6, the first category does not explicitly require a system model, but only the ability to sample trajectories. The latter, instead, employs the collected data to learn a model. In *stochastic trajectory generation* methods, the model can be used as a simulator for sampling trajectories and can be easily combined with model-free PS. Alternatively, *deterministic trajectory prediction* does not perform any sampling, but relies on an analytical prediction of the trajectory distribution. In both cases, different update strategies are possible, such as *Policy Gradient* (PG) [Williams, 1992; Peters and Schaal, 2006, 2008], *Expectation-Maximization* (EM) [Kober and Peters, 2009; Neumann, 2011] and *Information-Theoretic Insights* (Inf.Th.) [Peters et al., 2010; Daniel et al., 2012]. From now on, the discussion will mainly focus on PG methods, since they are the most relevant for the subsequent development in this thesis.

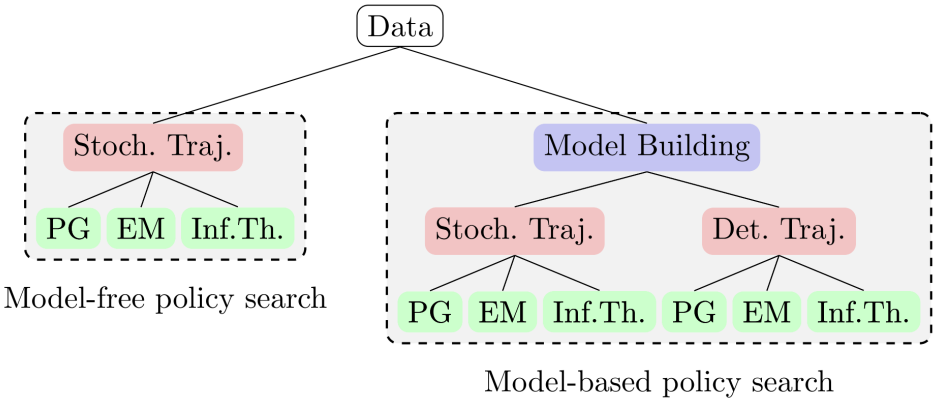

Figure 2.2: A taxonomy of Policy Search methods [Deisenroth et al., 2013]. Model-free PS (left sub-tree) directly uses trajectories for updating the policy, while model-based PS (right sub-tree) employs data to learn a model. In both model-free and model-based algorithms, the update are based on either policy gradients (PG), expectation-maximization (EM) or information-theoretic (Inf.Th.) insights.

2.8 Policy Gradient

Model-free policy search (Mf-PS) [Deisenroth et al., 2013] methods perform policy updates directly based on sampled trajectories τ, taking into account the collected immediate rewards $R(\tau)$, expressed as:

$$R(\tau) = \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t). \tag{2.7}$$

Such updates aim to modify the parameters $\boldsymbol{\theta}$, so that trajectories with higher rewards become more likely when following the new policy. Therefore, the purpose is to increase the *expected accumulated reward*:

$$J_{\boldsymbol{\theta}} = \mathbb{E}[R(\tau)|\boldsymbol{\theta}] = \int_{\mathcal{T}} R(\tau) p_{\boldsymbol{\theta}}(\tau) \mathrm{d}\tau, \tag{2.8}$$

where $\mathcal{T}$ is the trajectory space and $p_{\boldsymbol{\theta}}(\tau)$ represents the distribution of trajectories τ under policy $\pi_{\boldsymbol{\theta}}$. Given a stochastic policy, the latter is defined as:

$$p_{\boldsymbol{\theta}}(\tau) = \mu_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t) \pi_{\boldsymbol{\theta}}(a_t|s_t).$$

Algorithm 2.1 formalizes the generic Mf-PS method, highlighting three important steps during the procedure. The first one, *exploration*, is at the heart of one of the main challenges in RL, the so-called exploration-exploitation

Algorithm 2.1 Model-Free Policy Search

repeat
  Explore: generate trajectories $\tau^{(i)}$ using policy π_k
  Evaluate: assess quality of trajectories or actions
  Update: compute π_{k+1} given trajectories $\tau^{(i)}$ and evaluations
until policy converges $\pi_{k+1} \approx \pi_k$

tradeoff [Sutton and Barto, 1998]. To obtain high rewards, the agent must prefer actions that it has tried in the past and found to be profitable. But, to discover them, it has to experience actions that it has not selected before. Moreover, exploration can be categorized into *step-based* and *episode-based* strategies. The former chooses an exploratory action in each time step, while the latter changes $\boldsymbol{\theta}$ only at the start of the episode by sampling from a higher-level policy. This choice will also influence how the policies will be *evaluated*. Indeed, step-based evaluation strategies decompose the trajectory τ in its atomic steps, evaluating the single actions. In comparison, episode-based evaluation directly uses the returns of the whole trajectories to evaluate the quality of the policy parameters $\boldsymbol{\theta}$. Finally, as seen in Section 2.7, the *update* will be performed according to the optimization method employed by the PS algorithm. Since we are focusing on PG techniques, $\boldsymbol{\theta}$ will be modified towards the direction of maximum growth of the expected return $J_{\boldsymbol{\theta}}$, given by its gradient $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$. This approach is named *gradient ascent* and obeys the following update rule:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}, \tag{2.9}$$

where $\boldsymbol{\theta}_k$ and $\boldsymbol{\theta}_{k+1}$ represent the policy parameters at two subsequent instants of time, α is the *learning rate*, a user-specified real number, and the policy gradient is given by:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \int_{\mathcal{T}} R(\tau) \nabla p_{\boldsymbol{\theta}}(\tau) \, \mathrm{d}\tau.$$

These two operations, in addition to sample collection, form the backbone of a generic PG method (Algorithm 2.2). Such approach, under mild conditions on the step size and the regularity of $J_{\boldsymbol{\theta}}$, guarantees convergence to a (local) optimum [Bottou, 1998]. Since in Mf-PS the transition model and the reward model are unknown, the gradient needs to be estimated by the collected samples. Thus, the main problem of PG methods reduces to identifying a good estimator of the policy gradient [Peters and Schaal, 2008].

Algorithm 2.2 Policy Gradient

Initialize policy parameters $\boldsymbol{\theta}_0$
for $k = 0, \ldots,$ convergence **do**
Generate trajectories using policy $\boldsymbol{\theta}_k$
Estimate the gradient:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \int_{\tau} R(\tau) \nabla p_{\boldsymbol{\theta}}(\tau) \, \mathrm{d}\tau$$

Update policy parameters:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$$

end for
return optimal policy $\pi_{\boldsymbol{\theta}^*}$

2.9 Policy Gradient Theorem

A convenient expression for the policy gradient is provided by the Policy Gradient Theorem (PGT) [Sutton et al., 1999]. Since its formulation is particularly relevant for successive methods, a brief derivation will be shown. Starting from the definition of gradient, the following equivalences hold:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \mathbb{E}[R(\tau)|\boldsymbol{\theta}] = \nabla_{\boldsymbol{\theta}} \int_{\mathcal{T}} R(\tau) p_{\boldsymbol{\theta}}(\tau) \mathrm{d}\tau \tag{2.10}$$

$$= \int_{\mathcal{T}} R(\tau) \nabla_{\boldsymbol{\theta}} p_{\boldsymbol{\theta}}(\tau) \mathrm{d}\tau \tag{2.11}$$

$$= \int_{\mathcal{T}} R(\tau) p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau), \tag{2.12}$$

where in Equation (2.11) the gradient operator is brought inside the integral and in Equation (2.12) the *"likelihood ratio trick"* is exploited. It consists in the identity $\nabla p_{\boldsymbol{\theta}}(y) = p_{\boldsymbol{\theta}}(y) \nabla \log p_{\boldsymbol{\theta}}(y)$, easily provable applying the chain rule to compute the derivative of $\log p_{\boldsymbol{\theta}}(y)$:

$$\nabla \log p_{\boldsymbol{\theta}}(y) = \frac{\nabla p_{\boldsymbol{\theta}}(y)}{p_{\boldsymbol{\theta}}(y)}. \tag{2.13}$$

Moreover, Equation (2.12) can also be expressed as:

$$\int_{\mathcal{T}} R(\tau) p_{\boldsymbol{\theta}}(\tau) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau) = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} \Big[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau) R(\tau) \Big].$$

In case of step-based algorithms, the term $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau)$ will result in:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau) &= \nabla_{\boldsymbol{\theta}} \log \left(\mu_0(s_0) \prod_{t=1}^{T} P(s_{t+1}|s_t, a_t) \pi_{\boldsymbol{\theta}}(a_t|s_t) \right) \\ &= \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t), \end{aligned}$$

where the product transforms into a sum because of the logarithm and all the terms which do not depend on $\boldsymbol{\theta}$ disappear during differentiation. Hence, the final result is given by:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} \Big[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) R(\tau) \Big]. \tag{2.14}$$

Equation (2.14) reveals that, under the assumption of stochastic policies, PG does not depend on the transition model P, apart from the expectation.

2.10 Likelihood-ratio policy gradients

Likelihood-ratio policy gradients [Peters and Schaal, 2008] are a family of methods become popular in the early 1990s, which takes the name from the so-called "likelihood-ratio trick" (Equation (2.13)). Such techniques allow evaluating $\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$ resorting to estimators that can be computed from samples, approximating Equation (2.14) to:

$$\widehat{\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}} = \frac{1}{n} \sum_{j=1}^{n} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\tau_j) R(\tau_j),$$

where $\tau_j \sim p_{\boldsymbol{\theta}}$. Furthermore, since the gradient has to be estimated through sampling, often this operation is performed relying on Monte Carlo (see Section 3.1), like in the example above. As it was discussed in Section 3.2, such estimator is inherently inclined to high variances, thus a *baseline* can be introduced to mitigate this effect:

$$\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} \Big[\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t|s_t) \Big(R(\tau) - b \Big) \Big]. \tag{2.15}$$

Algorithm 2.3 REINFORCE Algorithm

Input: policy parametrization $\boldsymbol{\theta}$, dataset $\mathcal{D} = \{\tau_1, \tau_2, \ldots, \tau_n\}$
Compute returns:

$$R^{(i)} = \sum_{t=0}^{T} \gamma^t r_t^{(i)}$$

for each dimension h of $\boldsymbol{\theta}$ **do**
Estimate the optimal baselines:

$$b_h = \frac{\sum_{i=1}^{n} \left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_t^{(i)} | s_t^{(i)}) \right)^2 R^{(i)}}{\sum_{i=1}^{n} \left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_t^{(i)} | s_t^{(i)}) \right)^2}$$

Estimate the gradient element:

$$\widehat{\nabla_{\boldsymbol{\theta}_h} J_{\boldsymbol{\theta}}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_t^{(i)} | s_t^{(i)}) (R^{(i)} - b_h)$$

end for
return gradient estimate $\widehat{\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}}$

It can be proved that this contribution leaves the estimator unbiased [Deisenroth et al., 2013]. Since the baseline is a free parameter, it can be chosen in order to minimize the variance of the gradient estimate. Such baseline will be referred as optimal baseline. Depending on how the gradient estimation is performed and on how the baseline is derived, different methods are implemented.

2.10.1 REINFORCE

The concepts behind the PGT were used by one of the first PG algorithms, REINFORCE [Williams, 1992]. The relative estimator can be defined by rephrasing Equation (2.15), simply taking its Monte Carlo version:

$$\hat{\nabla}_{\boldsymbol{\theta}}^{RF} J_{\boldsymbol{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}} \left(a_t^{(i)} | s_t^{(i)} \right) \right) \left(\sum_{t=0}^{T-1} \gamma^t r_t^{(i)} - b \right),$$

obtained considering n independent trajectories. As previously said, to reduce the variance of such estimator, the optimal baseline b^{RF} can be determined. This will depend on the h-th element of the gradient, so it will

Algorithm 2.4 G(PO)MDP Algorithm

Input: policy parametrization $\boldsymbol{\theta}$, dataset $\mathcal{D} = \{\tau_1, \tau_2, \dots, \tau_n\}$
for each dimension h of $\boldsymbol{\theta}$ **do**
  for each time step $t = 0, \dots, T-1$ **do**
    Estimate the optimal time-dependent baseline:

$$b_{h,t} = \frac{\sum_{i=1}^{n} \left(\sum_{j=0}^{t} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}} \left(a_j^{(i)} | s_j^{(i)} \right) \right)^2 \gamma^t r_t^{(i)}}{\sum_{i=1}^{n} \left(\sum_{j=0}^{t} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}} \left(a_j^{(i)} | s_j^{(i)} \right) \right)^2}$$

  end for
  Estimate the gradient element:

$$\widehat{\nabla_{\boldsymbol{\theta}_h} J_{\boldsymbol{\theta}}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \sum_{j=0}^{t} \left(\nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}} \left(a_j^{(i)} | s_j^{(i)} \right) \right) \left(\gamma^t r_t^{(i)} - b_{h,t} \right)$$

end for
return gradient estimate $\widehat{\nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}}$

be computed separately for each dimension. It can be proved [Deisenroth et al., 2013] that the optimal baseline corresponds to:

$$b_h^{RF} = \frac{\mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}} \left[\left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_t | s_t) \right)^2 R(\tau) \right]}{\mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}} \left[\left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_t | s_t) \right)^2 \right]}.$$

Through this definitions it is possible to write the pseudocode of this technique (Algorithm 2.3).

2.10.2 G(PO)MDP

Gradient of a (Partially Observable) Markov Decision Process (G(PO)MDP) [Baxter and Bartlett, 2000, 2001] tries to further reduce the variance of the REINFORCE estimator. Indeed, the latter does not fully exploit the causality between actions and rewards, described in Equation (2.15). More explicitly, the independence between rewards and *future* decisions is not taken into account. Indeed, REINFORCE uses the returns $R(\tau)$ to evaluate single actions, instead of choosing a step-based evaluation strategy. Thus, it can be seen that the reward at a given $t \in \mathbb{N}$ is independent of all the actions performed at $t' > t$. Therefore, the policy gradient can be rewritten as:

$$\nabla^G_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\tau \sim p_{\boldsymbol{\theta}}} \left[\sum_{t=0}^{T-1} \left(\sum_{j=0}^{t} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_j | s_j) \right) (r_t - b_t) \right].$$

Similarly to REINFORCE, to reduce the variance of the corresponding estimator, a baseline can be introduced. This time, however, it will be time-dependent, to reflect the aforementioned properties:

$$b^G_{h,t} = \frac{\mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}} \left[\left(\sum_{j=0}^{t} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_h | s_h) \right)^2 \gamma^t r_t \right]}{\mathbb{E}_{\tau \sim p_{\boldsymbol{\theta}}} \left[\left(\sum_{j=0}^{t} \nabla_{\boldsymbol{\theta}_h} \log \pi_{\boldsymbol{\theta}}(a_h | s_h) \right)^2 \right]}.$$

Finally, we can define the G(PO)MDP estimator, which will be at the basis of the homonymous algorithm (Algorithm 2.4):

$$\hat{\nabla}^G_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{t=0}^{T-1} \left(\sum_{j=0}^{t} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_j^{(i)} | s_j^{(i)}) \right) \left(\gamma^t r_t^{(i)} - b_t \right) \right).$$

Chapter 3

State of the art

In recent years, PO methods have been widely exploited in RL, proving to be a successful approach in addressing a variety of problems, such as continuous-control [Peters and Schaal, 2008; Lillicrap et al., 2016], robot manipulation [Gu et al., 2017; Chatzilygeroudis et al., 2020] and locomotion [Kohl and Stone, 2004; Duan et al., 2016]. Most of these techniques employ the notion of *trust region* [Conn et al., 2000], which allows seeking the successive policy "sufficiently close" to the current one. This preference towards small policy updates gave rise to many algorithms, including *Relative Entropy Policy Search* (REPS) [Kober and Peters, 2009], *Trust Region Policy Optimization* (TRPO) [Schulman et al., 2015], *Proximal Policy Optimization* (PPO) [Schulman et al., 2017] and *Policy Optimization via Importance Sampling* (POIS) [Metelli et al., 2018]. Moreover, at the basis of a large number of trust-region methods there is a fundamental statistical tool, namely *Importance Sampling* (IS) [Owen, 2013]. Indeed, IS allows evaluating the performance of a policy through samples collected with a different one, paving the way for *off-policy* learning.

Chapter Outline

The chapter is organized as follows. Section 3.2 introduces the basic Importance Sampling estimator, highlighting its main properties. Section 3.3 and Section 3.4 show how, considering multiple distributions, such estimator can be extended. Section 3.5 and Section 3.6 lay the groundwork for the successive topics, describing the practical implications of applying Importance Sampling to RL. Finally, Section 3.7 provides an extensive overview of the main trust-region methods, briefly presenting some state-of-the-art algorithms.

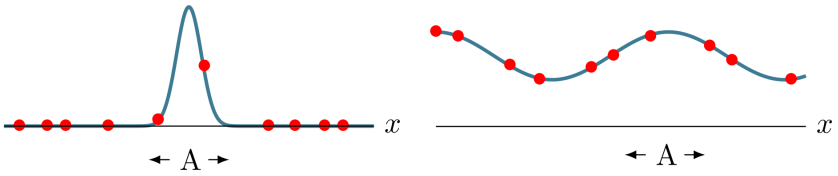

Figure 3.1: Graphical representation of a function modeling a rare event (left) and of a regular function (right). When dealing with rare events, the probability of collecting samples in A will be very small and the final estimate will be predominantly based on points such that $f(x) = 0$. Instead, if the function we want to study is quite regular, having such region A will not be so harmful to MC.

3.1 Monte Carlo Simulations

The Monte Carlo (MC) [Owen, 2013] method is a powerful and flexible approach to learn about a system by simulating it with random sampling. Besides being a simple and direct method, sometimes MC can even prove to be the only feasible solution to a problem. A classical application of MC consists in estimating the value of an integral: a sufficient number of points needs to be sampled within the integration domain, in order to return their average as the final result. Consider now a measurable set A and a random variable X, such that $\mathbb{P}(X \in A)$ is small, and let $f(x)$ be nearly zero outside region A. Intuitively, if we follow an MC approach, the vast majority of the samples - if not all of them - will be outside A. As a consequence, if the problem consists in studying $\mathbb{E}[f(X)]$, relying on MC can lead to a poor estimate. It is clear that some samples from the interesting or important region must be picked. However, this does not merely mean forcing the process to collect more data in A. In fact, MC is based on the assumption that the samples are randomly drawn, so this would result in introducing a strong bias and in an overestimation of the expected value. Moreover, in most situations, we do not even have any prior knowledge of the function to be estimated. A widespread solution to this problem is provided by Importance Sampling (IS) [He and Owen, 2014]. As we will see in the next section, IS relies on sampling from a distribution that overweights the important region, hence the name. Having oversampled, we then need to introduce an adjustive factor to account for the previous bias. Given these premises, IS can bring enormous gains and allow working with rare events. However, it can also backfire, yielding an infinite variance estimate when MC would have had a finite variance.

3.2 Basic Importance Sampling

Let X be a random variable, f the integrand, P a probability measure and p its density function, such that $p(x) = 0$ outside the integration domain. Suppose we want to estimate the aforementioned quantity

$$\mu = \mathbb{E}_P[f(X)] = \int f(x)p(x)\ \mathrm{d}x. \tag{3.1}$$

If we introduce a new probability measure Q with relative density function q, also defined over the integration interval, we can rewrite Equation (3.1) as:

$$\mu = \int f(x)p(x)\ \mathrm{d}x = \int \frac{f(x)p(x)}{q(x)} q(x)\ \mathrm{d}x = \mathbb{E}_Q\left[\frac{f(X)p(X)}{q(X)}\right], \tag{3.2}$$

where $q(x) > 0$ whenever $f(x)p(x) \neq 0$. Furthermore, the possibility of dividing by zero is not a cause for concern, since samples are drawn from q and $q(x) = 0$ implies that the event will happen with zero probability. Starting from Equation (3.2), we can define the IS estimator:

$$\hat{\mu}_{P/Q} = \frac{1}{n}\sum_{i=1}^{n} \frac{p(x_i)}{q(x_i)} f(x_i) = \frac{1}{n}\sum_{i=1}^{n} w_{P/Q}(x_i) f(x_i), \tag{3.3}$$

where the ratio $w_{P/Q}(x_i) = p(x_i)/q(x_i)$ is the corrective factor, also known as *importance weight* (or likelihood ratio or Radon-Nikodym derivative). This estimator is well-known to be unbiased [Owen, 2013], i.e.

$$\mathbb{E}_{x_i \sim Q}[\hat{\mu}_{P/Q}] = \mathbb{E}_{x_i \sim P}[f(x)],$$

but it might suffer from large variance. In fact, bad choices for q can lead to $\sigma_q^2 = \infty$ and even good ones can result in very high σ_q^2, due to small values in the denominator. Not by chance, importance sampling is considered the hardest variance reduction method to use well. However, under the assumption of $f(x) \geqslant 0$, good choices for q can yield $\sigma_q^2 \ll \sigma_p^2$, up to consider the extreme case of zero variance:

$$q(x) = \frac{f(x)p(x)}{\mu}. \tag{3.4}$$

Obviously, such distribution is not feasible because it requires knowledge of μ. Nevertheless, Equation (3.4) suggests that optimal values of q can be found when it is nearly proportional to $f(x)p(x)$. Further properties of the IS estimator and several of its transformations have been extensively studied

in the literature [Ionides, 2008; Owen, 2013; Papini et al., 2019; Metelli et al., 2020; Kuzborskij et al., 2021].

3.3 Mixture Importance Sampling

Resuming what was discussed in the previous section, IS can be a very powerful statistical tool or backfire according to the chosen q. As a consequence, determining the correct q plays a crucial role and often this task turns out to be anything but trivial. To maximize the chances of success, instead of sampling from a single q, samples can be drawn from multiple distributions q_j, for $j = 1, \dots, J$, corresponding to different extreme scenarios. In literature, this method is known as *Mixture Importance Sampling* [Owen, 2013; He and Owen, 2014] and the mixture distribution q_α is defined as:

$$q_\alpha = \sum_{j=1}^{J} \alpha_j q_j, \tag{3.5}$$

where α_j is the probability to sample from q_j and is subject to $\alpha_j \geqslant 0$ and $\sum_{j=1}^{J} \alpha_j = 1$. In some Bayesian contexts, mixtures are very convenient to sample from, since combinations of unimodal densities can provide a flexible approximation to multimodal densities. Similarly, we might seek a mixture describing peaks in $f(x)p(x)$, since the ideal q is proportional not only to p, as already seen in Equation (3.4). Furthermore, mixtures allow us to avoid too light tails for q. In fact, if p is one mixture component, then the tails of q cannot be much lighter than those of p. Finally, we can extend the basic IS notation to the mixture case:

$$\hat{\mu}_{P/Q_{1:J}} = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)p(x_i)}{q_\alpha(x_i)} = \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)p(x_i)}{\sum_{j=1}^{J} \alpha_j q_j(x_i)}. \tag{3.6}$$

According to this definition, we have to compute $q_j(x_i)$ for all the J distributions, regardless of which proposal density generated x_i. Equation (3.6) does not indeed even take into account which component delivered x_i.

3.4 Multiple Importance Sampling

Multiple Importance Sampling (MIS) [Veach and Guibas, 1995; Owen, 2013] is a large family of estimators generalizing what we have discussed so far. This method follows the core idea of mixture importance sampling, so samples will be again collected from J different distributions. The difference is that, now, sampling will take place independently, generating n_j data points

for each q_j, for a total of $n = \sum_{j=1}^{J} n_j$ observations. Therefore, the resulting estimator is given by:

$$\hat{\mu}^{\beta}_{P/Q_{1:J}} = \sum_{j=1}^{J} \frac{1}{n_j} \sum_{i=1}^{n_j} \beta_j(x_{ij}) \frac{p(x_{ij})}{q_j(x_{ij})} f(x_{ij}) =$$

$$= \sum_{j=1}^{J} \frac{1}{n_j} \sum_{i=1}^{n_j} \beta_j(x_{ij}) w_{P/Q_j} f(x_{ij}), \tag{3.7}$$

where we assume $q_j(x) > 0$ whenever $\beta_j(x)p(x)f(x) \neq 0$. Moreover, $\beta_j(x)$ has to be a *partition of the unity*, i.e. a collection of weight functions such that $\beta_j(x) \geqslant 0$ for $j = 1, \ldots, J$ and $\sum_{j=1}^{J} \beta_j(x) = 1$. Under these conditions, the MIS estimator is still unbiased [Owen, 2013]. A straightforward, but inefficient, choice for β is $\beta_j(x) = n_j/n$, which assigns the same importance to all the samples. Many other variants - such as the *cutoff*, *maximum* and *power heuristics* - have been proposed in the literature [Owen, 2013]. Among them, the most common and studied one is the *balance heuristic* [Veach and Guibas, 1995], thanks to its theoretical properties:

$$\beta^{BH}_j(x) = \frac{n_j q_j(x)}{\sum_{k=1}^{J} n_k q_k(x)}. \tag{3.8}$$

Thus, combining Equation (3.7) and Equation (3.8) we can derive the formulation of the MIS estimator with balance heuristic:

$$\hat{\mu}^{BH}_{P/Q_{1:J}} = \frac{1}{n} \sum_{j=1}^{J} \sum_{i=1}^{n_j} \frac{p(x_{ij})}{\sum_{k=1}^{J} \frac{n_k}{n} q_k(x_{ij})} f(x_{ij}) =$$

$$= \frac{1}{n} \sum_{j=1}^{J} \sum_{i=1}^{n_j} w^{BH}_{P:Q_{1:J}}(x_{ij}) f(x_{ij}). \tag{3.9}$$

The choice of the coefficient functions according to the balance heuristic has been proved [Veach and Guibas, 1995] to be nearly optimal in terms of variance of the estimator $\hat{\mu}_{P/Q_{1:J}}$. Finally, exploiting balance heuristics has the advantage of canceling out q_j in Equation (3.9). This means that the weight of a given sample x_{ij} does not depend on which component of the mixture it comes from.

3.5 Importance Sampling and RL

Throughout this chapter, we presented IS independently of the RL framework, always referring to generic distributions p and q. We now import notation from the off-policy setting, where we name *behavioral* the policy used to collect the samples and *target* the policy which is being learned. From what we have seen, importance sampling is an *active* tool for *variance minimization* (Off-VM), as originally defined in the Monte Carlo simulation community [Hesterberg, 1988; Hammersley, 2013]. Indeed, the target policy is fixed, while the best behavioral policy is searched. Once found, this yields an IS estimate with the lowest possible variance [Kahn and Marshall, 1953]. In this guise, IS has been employed in RL mainly to address rare events, which lead to high-variance estimates when tackled on policy [Frank et al., 2008; Ciosek and Whiteson, 2017]. Nevertheless, we can also exploit IS for estimating $\mathbb{E}[f(x)]$ under multiple alternative distributions for X. This approach is known as *what-if* simulation [Owen, 2013] and it is often employed as a *passive* analysis tool in off-policy learning (Off-PL). In this scenario, instead, the roles of the policies are reversed. More explicitly, the behavioral policy is fixed and we look for the best target policy, whose performance we aim to estimate. The idea of using IS as a variance reduction technique, with the specific goal of finding an optimal behavioral policy, was already proposed by [Hanna et al., 2017] for evaluation and subsequently combined with policy gradient learning [Hanna and Stone, 2018; Hanna et al., 2019]. However, in these works, Off-VM and Off-PL are treated separately. The purpose of this thesis is to investigate the relation between them, trying to combine this dualism and to leverage Off-VM as a tool for Off-PL.

3.5.1 What-if simulations

In order to properly discuss the ambivalent nature of IS, its usage as *what-if* tool will be better clarified in this section. In the Off-PL setting, as mentioned above, a target policy π is learned by exploiting the samples collected through a behavioral policy q. Moreover, the role of function $f(x)$ can be played by the cumulative reward $R(\tau)$, already defined in Equation (2.7). Given these definitions, $J_\pi = \mathbb{E}[R(\tau)|\pi]$, that is the expected reward following policy π, can be estimated in an off-policy fashion as:

$$\hat{J}_\pi = \frac{1}{n}\sum_{i=1}^{n} \frac{p_\pi(\tau_i)}{p_q(\tau_i)} R(\tau_i), \tag{3.10}$$

where the ratio $p_\pi(\tau_i)/p_q(\tau_i)$ is the importance weight. Furthermore, expressing it in terms of policies instead of distribution of trajectories, it holds that:

$$\hat{J}_\pi = \frac{1}{n}\sum_{i=1}^{n} \frac{\prod_{t=0}^{T-1} \pi\left(a_t^{(i)}|s_t^{(i)}\right)}{\prod_{t=0}^{T-1} q\left(a_t^{(i)}|s_t^{(i)}\right)} R(\tau_i). \tag{3.11}$$

Therefore, adopting IS, the trajectories are collected following the behavioral policy and the updates of the target policy are performed according to the RL method. For example, in a PS setting, Equation (3.10) can be used for an off-policy estimate of $J_{\boldsymbol{\theta}}$, whose gradient is then needed in the update rule stated in Equation (2.9).

3.5.2 Per-Decision Importance Sampling

In the previous section, it was shown how to determine an IS estimate of the expected reward following policy π, employing a unique importance weight for the whole trajectory τ. However, this estimator can be further refined by observing that the reward at a certain time step t does not depend on actions and states visited after t. This is the rationale behind *Per-Decision Importance Sampling* (PDIS) [Precup et al., 2000], a technique that reweights the reward at time t, limiting the importance weight to consider the products of policy ratios up to t. Indeed, starting from Equation (3.11), the ratios successive to time step t can be expanded as:

$$\frac{\pi\left(a_{t-1}^{(i)}|s_{t-1}^{(i)}\right)}{q\left(a_{t-1}^{(i)}|s_{t-1}^{(i)}\right)} \frac{\pi\left(a_t^{(i)}|s_t^{(i)}\right)}{q\left(a_t^{(i)}|s_t^{(i)}\right)} \frac{\pi\left(a_{t+1}^{(i)}|s_{t+1}^{(i)}\right)}{q\left(a_{t+1}^{(i)}|s_{t+1}^{(i)}\right)} \cdots \frac{\pi\left(a_{T-1}^{(i)}|s_{T-1}^{(i)}\right)}{q\left(a_{T-1}^{(i)}|s_{T-1}^{(i)}\right)} R(s_t^{(i)}|a_t^{(i)}).$$

As reported in [Sutton and Barto, 1998], the off-policy estimators rely on the expected values of these terms and each sub-term is a product of a random reward by a random importance ratio. Thus, of all these factors, only the first and the last are correlated, while the others are random variables whose expected value is 1. Therefore, the previous chain for future actions and states reduces to:

$$\frac{\pi\left(a_{t-1}^{(i)}|s_{t-1}^{(i)}\right)}{q\left(a_{t-1}^{(i)}|s_{t-1}^{(i)}\right)} R(s_t^{(i)}|a_t^{(i)}).$$

If this procedure is applied to every time step, the final formulation of the PDIS estimator can be derived as:

$$\hat{J}_\pi = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T-1}\frac{\prod_{j=0}^{t-1}\pi\left(a_j^{(i)}|s_j^{(i)}\right)}{\prod_{j=0}^{t-1}q\left(a_j^{(i)}|s_j^{(i)}\right)}R\left(s_t^{(i)}|s_t^{(i)}\right).$$

3.6 Rényi divergence

In this section, we introduce a mathematical background that will be useful in later chapters, namely the *Rényi divergence* [Rényi, 1961; van Erven and Harremoës, 2014].

Definition 3.6.1 (Rényi divergence)**.** Let P and Q be two probability measures such that P is absolutely continuous w.r.t. Q, we define, for every $\alpha \in [0, \infty]$, the α-Rényi divergence as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1}\log\int p(x)^\alpha q(x)^{1-\alpha}\,\mathrm{d}x.$$

However, in many circumstances, it will be more convenient to refer to its exponentiated version, $d_\alpha(P\|Q) = \exp\{D_\alpha(P\|Q)\}$ [Cortes et al., 2010].

Therefore, the Rényi divergence is a dissimilarity index between probability measures P and Q. Some edge cases which require an additional comment are the following:

- for $\alpha = 1$, $D_1(P\|Q) = D_{KL}(P\|Q)$. We recall that $D_{KL}(P\|Q)$ is the *Kullback-Leibler divergence*, defined as:

$$D_{KL}(P\|Q) = \int p(x)\ln\frac{p(x)}{q(x)}\,\mathrm{d}x;$$

- for $\alpha = \infty$, $D_\infty(P\|Q) = \ln \operatorname{ess\,sup}_{x\sim Q}\left\{\frac{p(x)}{q(x)}\right\}$. We recall that ess sup is the essential supremum of a measurable function f.

Since the concept of importance weight was defined as $w_{P/Q}(x) = \frac{p(x)}{q(x)}$, its magnitude can be interpreted as how much P and Q are dissimilar. In light of Definition 3.6.1, Rényi divergence can be related with the variance and the essential supremum of important weights [Cortes et al., 2010]:

$$\underset{x\sim Q}{\mathbb{V}\mathrm{ar}}\left[w_{P/Q}(x)\right] = d_2(P\|Q) - 1,$$

$$\operatorname*{ess\,sup}_{x\sim Q}\{w_{P/Q}(x)\} = d_\infty(P\|Q).$$

Finally, as studied by [Papini et al., 2019; Metelli et al., 2020], the variance of the IS estimator can be bounded for every $\alpha \in [1, \infty]$ and these results can be extended to MIS:

$$\operatorname*{\mathbb{V}ar}_{x\sim Q}[\hat{\mu}_{P/Q}] \leqslant \frac{1}{n}\|f\|^2_{Q,\frac{2\alpha}{\alpha-1}} d_{2\alpha}(P\|Q)^{2-\frac{1}{\alpha}},$$

$$\operatorname*{\mathbb{V}ar}_{x\sim Q_{1:J}}[\hat{\mu}^{BH}_{P/Q_{1:J}}] \leqslant \frac{1}{n}\|f\|^2_{\Phi,\frac{2\alpha}{\alpha-1}} d_{2\alpha}(P\|\Phi)^{2-\frac{1}{\alpha}}.$$

where $\Phi = \sum_{k=1}^{J} \frac{n_k}{n} Q_k$ is a finite mixture, $x \sim Q_{1:J}$ abbreviates $x_j \sim Q_j$ for all $j = 1, \dots, J$ all independent and f is a function with bounded $\frac{2\alpha}{\alpha-1}$-moment under Φ, i.e. $\|f\|^2_{\Phi,\frac{2\alpha}{\alpha-1}} < +\infty$.

3.7 Trust-region methods

PG methods have proven to be effective approaches to address continuous control tasks, especially in presence of continuous action spaces. Nevertheless, the estimated policy gradient direction has a *local* nature and PG algorithms only employ first-order information. As a consequence of this, the update rule $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha \nabla_{\boldsymbol{\theta}} J_{\boldsymbol{\theta}}$ offers no guarantee on the successive induced distribution. Thus, some updates may lead to performance oscillations and unexpected behavior [Kakade and Langford, 2002]. To mitigate this problem, *trust-region* [Conn et al., 2000; Schulman et al., 2015] methods propose to perform the optimization of $\boldsymbol{\theta}$ in a constrained region, i.e. a sufficiently small neighborhood of the current parametrization. During learning, if the ratio between achieved and predicted reduction of the objective function is positive enough, the trust region will be expanded. On the contrary, in case of a poor approximation, the region will contract.

3.7.1 REPS

Relative Entropy Policy Search (REPS) [Peters et al., 2010; Daniel et al., 2012] is an information-theoretic approach that rephrases the policy search problem into an optimization problem, directly optimizing in the space of distributions and allowing a closed-form solution. The key idea consists in bounding the loss of information, measured via relative entropy, between the observed data distribution q and the data distribution p_π, induced by

the new policy π. This constraint, enforced through the KL divergence $D_{KL}(p_\pi \| q)$, prevents p_π from being too dissimilar with respect to q.
Importing notation from Section 2.1, it can be said that the agent, after a sequence of state transitions, may converge to a stationary state distribution $\mu_\pi(s)$ for which:

$$\mu_\pi(s') = \sum_{s,a} \mu_\pi(s)\pi(a|s)P(s'|s,a), \quad \forall s' \in \mathcal{S}, \tag{3.12}$$

holds under mild conditions [Sutton et al., 1999]. Moreover, the γ-*discounted future state distribution* can be defined as:

$$\mu_\pi(s) = \sum_{t \geqslant 0} \gamma^t \, \mathbb{P}(s_t = s)$$

Since the goal of the agent is to find a policy that maximizes the expected return, the optimization problem can be expressed as:

$$\max_{\pi, \mu_\pi} J(\pi) = \sum_{s,a} \mu^\pi(s)\pi(a|s)r(s,a), \tag{3.13}$$

$$\text{s.t. } \epsilon \geqslant \sum_{s,a} \mu_\pi(s)\pi(a|s) \log \frac{\mu_\pi(s)\pi(a|s)}{q(s,a)}, \tag{3.14}$$

$$\sum_{s'} \mu_\pi(s')\phi(s') = \sum_{s,a,s'} \mu_\pi(s)\pi(a|s)P(s'|s,a)\phi(s'), \tag{3.15}$$

$$1 = \sum_{s,a} \mu_\pi(s)\pi(a|s), \tag{3.16}$$

where Equation (3.13) is the goal, Equation (3.14) represents the divergence constraint and ϵ is the maximal information loss. Moreover, two additional conditions that must hold are stated in Equation (3.15) and Equation (3.16). The former is a feature-based proxy of the recursive formulation in Equation (3.12), where ϕ is a feature function. The latter, instead, ensures that p_π is a distribution. Such optimization problem can be efficiently solved by the method of Lagrangian multipliers. From the Lagrangian, it is possible to obtain the closed-form solution for the new policy:

$$\pi(a|s) = \frac{q(s,a) \exp\left(\frac{1}{\eta}\delta_\theta(s,a)\right)}{\sum_b q(s,b) \exp\left(\frac{1}{\eta}\delta_\theta(s,b)\right)},$$

where $\delta_\theta(s,a) = r(s,a) + \sum_{s'} P(s'|s,a)\theta^T\phi(s') - \theta^T\phi(s)$ and the Lagrangian parameters $\eta \in [0,\infty]$ and θ can be computed by solving the dual problem:

$$g(\theta,\eta) = \eta \log\left(\sum_{s,a} q(s,a)\exp\left(\epsilon + \frac{1}{\eta}\delta_\theta(s,a)\right)\right).$$

3.7.2 TRPO

Trust Region Policy Optimization (TRPO) [Schulman et al., 2015] is a state-of-the-art algorithm for continuous control, which has shown impressive results in combination with high-performing policies. As it belongs to the trust-region family, TRPO constrains the space of the candidate policies, ensuring that the following one lies in the proximity of the current parametrization. The formal definition of the optimization problem starts from a useful identity, which expresses the expected return of a policy $\pi_{\boldsymbol{\theta}'}$ in terms of advantage over $\pi_{\boldsymbol{\theta}}$ [Kakade and Langford, 2002]:

$$J_{\boldsymbol{\theta}'} = J_{\boldsymbol{\theta}} + \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}'}\\ a\sim\pi_{\boldsymbol{\theta}}}}[A_{\boldsymbol{\theta}}(s,a)],$$

where the term $A_{\boldsymbol{\theta}}(s,a)$ represents the *advantage function* and is defined as $A_{\boldsymbol{\theta}}(s,a) = Q_{\boldsymbol{\theta}}(s,a) - V_{\boldsymbol{\theta}}(s)$. Since TRPO aims to find a policy $\pi_{\boldsymbol{\theta}'}$ which is an improvement with respect to $\pi_{\boldsymbol{\theta}}$, it has to seek for the maximum difference between $J_{\boldsymbol{\theta}'}$ and $J_{\boldsymbol{\theta}}$. This intent, resorting to IS, can be mathematically expressed as:

$$\max_{\boldsymbol{\theta}'} J_{\boldsymbol{\theta}'} - J_{\boldsymbol{\theta}} = \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}'}\\ a\sim\pi_{\boldsymbol{\theta}}}}[A_{\boldsymbol{\theta}}(s,a)] \tag{3.17}$$

$$= \max_{\boldsymbol{\theta}'} \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}}\\ a\sim\pi_{\boldsymbol{\theta}}}}\left[\frac{\pi_{\boldsymbol{\theta}'}(a,s)}{\pi_{\boldsymbol{\theta}}(a|s)}A_{\boldsymbol{\theta}}(s,a)\right], \tag{3.18}$$

where the passage from Equation (3.17) to Equation (3.18) is performed by replacing the discounted future state distribution under $\pi_{\boldsymbol{\theta}'}$ with the one under $\pi_{\boldsymbol{\theta}}$. This approximation allows a consistent reduction in the complexity of the problem and it is justified in virtue of the constraint. Thus, making the latter explicit, the final optimization problem can be stated as:

$$\begin{aligned} \max_{\boldsymbol{\theta}'} \quad & \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}} \\ a\sim\pi_{\boldsymbol{\theta}}}} \left[\frac{\pi_{\boldsymbol{\theta}'}(a,s)}{\pi_{\boldsymbol{\theta}}(a|s)} A_{\boldsymbol{\theta}}(s,a) \right] \\ \text{s.t.} \quad & \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}} \\ a\sim\pi_{\boldsymbol{\theta}}}} \left[D_{KL}(\pi_{\boldsymbol{\theta}} \| \pi_{\boldsymbol{\theta}'}) \right] \leqslant \delta. \end{aligned} \tag{3.19}$$

A solution to this problem can be efficiently found using the conjugate gradient algorithm, after applying a linear approximation to the objective and a quadratic one to the constraint [Schulman et al., 2017]. However, if a penalty is used instead of the constraint, an alternative formulation of the same problem can be provided:

$$\max_{\boldsymbol{\theta}'} \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}} \\ a\sim\pi_{\boldsymbol{\theta}}}} \left[\frac{\pi_{\boldsymbol{\theta}'}(a,s)}{\pi_{\boldsymbol{\theta}}(a|s)} A_{\boldsymbol{\theta}}(s,a) - \beta D_{KL}(\pi_{\boldsymbol{\theta}} \| \pi_{\boldsymbol{\theta}'}) \right].$$

Nevertheless, TRPO is expressed in terms of the hard constraint, rather than the penalty, because it is difficult to determine a value of β that performs well across different problems. Indeed, if the characteristics change over the course of learning, finding the right β can be challenging even within a single scenario. Finally, the complexity of the optimization problem is aggravated by the fact that TRPO guarantees monotonic improvements. This concept is related to the field of *safe learning* [Pirotta et al., 2013b], where particular attention is paid to update the policy with the precaution of not reducing the performances of the next policies.

3.7.3 PPO

Although TRPO often performs quite well, it needs to calculate an estimate of the Kullback-Leibler divergence. This adds complexity and computing time to the implementations of the method, making it inefficient and difficult to scale up for large problems. *Proximal Policy Optimization* (PPO) [Schulman et al., 2017] methods are, at the same time, significantly simpler to implement and very effective in dealing with a wide range of challenging tasks. As it was shown in Equation (3.19), TRPO maximizes a surrogate objective, also referred to by the symbol $\mathcal{L}(\boldsymbol{\theta})$. Let $\rho_t(\boldsymbol{\theta'})$ denote the probability ratio:

$$\rho_t(\boldsymbol{\theta'}) = \frac{\pi_{\boldsymbol{\theta}'}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)},$$

then $\rho_t(\boldsymbol{\theta}) = 1$ and Equation (3.19) can be rewritten as:

$$\mathcal{L}^{TRPO}(\boldsymbol{\theta}') = \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}} \\ a\sim\pi_{\boldsymbol{\theta}}}} \left[\frac{\pi_{\boldsymbol{\theta}'}(a,s)}{\pi_{\boldsymbol{\theta}}(a|s)} A_{\boldsymbol{\theta}}(s,a) \right] = \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}} \\ a\sim\pi_{\boldsymbol{\theta}}}} \left[\rho_t(\boldsymbol{\theta}') A_{\boldsymbol{\theta}}(s,a) \right].$$

Since the absence of the constraint would lead to an excessively large policy update, a penalization for policies that move $\rho_t(\boldsymbol{\theta}')$ away from 1 is added. Thus, the surrogate objective is rephrased as follows:

$$\mathcal{L}^{PPO}(\boldsymbol{\theta}') = \mathop{\mathbb{E}}_{\substack{s\sim\mu_{\boldsymbol{\theta}} \\ a\sim\pi_{\boldsymbol{\theta}}}} \left[\min \left(\rho_t(\boldsymbol{\theta}') A_{\boldsymbol{\theta}}(s,a),\ \mathrm{clip}\big(\rho_t(\boldsymbol{\theta}'), 1-\epsilon, 1+\epsilon\big) A_{\boldsymbol{\theta}}(s,a) \right) \right],$$

where ϵ is a hyperparameter. The first term corresponds to $\mathcal{L}^{TRPO}(\boldsymbol{\theta}')$, while the second one clips the probability ratio and removes the incentive for moving ρ_t outside of the interval $[1-\epsilon, 1+\epsilon]$. Therefore, the clipping penalizes large deviations from the current policy $\pi_{\boldsymbol{\theta}}$, *implicitly* defining a trust region.

3.7.4 POIS

Policy Optimization via Importance Sampling (POIS) [Metelli et al., 2018, 2020] is a model-free actor-only policy optimization algorithm. To efficiently leverage the information contained in the collected trajectories, it mixes on-line and off-line optimization, relying on IS to perform multiple gradient steps with the same batch of data. Therefore, given a set of trajectories $\{\tau_i\}_{i=1}^n$, sampled through a *behavioral* policy $\pi_{\boldsymbol{\theta}}$, it aims to estimate a *target* policy $\pi_{\boldsymbol{\theta}'}$. To achieve this result, the performance J will be evaluated with the following estimator:

$$\hat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=0}^{T-1} \gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) r\left(s_t^{(i)}, a_t^{(i)}\right),$$

where $r(s_t^{(i)}, a_t^{(i)})$ is uniformly bounded, i.e. $|r(s_t^{(i)}, a_t^{(i)})| \leqslant R_{\max}$, and the per-decision importance weight $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i)$ is given by:

$$w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) = \prod_{l=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}\left(a_l^{(i)}|s_l^{(i)}\right)}{\pi_{\boldsymbol{\theta}}\left(a_l^{(i)}|s_l^{(i)}\right)}.$$

As proved by [Metelli et al., 2020], such estimator is unbiased and its variance can be bounded according to:

$$\mathbb{V}\mathrm{ar}_{\pi_{\boldsymbol{\theta}}}\left[\hat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] \leqslant \frac{R_{\max}^2}{n}\sum_{t=0}^{T-1} c_t d_2(p_{\boldsymbol{\theta}'}\|p_{\boldsymbol{\theta}}),$$

where $p_{\boldsymbol{\theta}'}$ and $p_{\boldsymbol{\theta}}$ represent the distribution over trajectories $p(\cdot|\boldsymbol{\theta}')$ and $p(\cdot|\boldsymbol{\theta})$, respectively induced by the target policy $\pi_{\boldsymbol{\theta}'}$ and the behavioral policy $\pi_{\boldsymbol{\theta}}$. Instead, the term c_t is defined as:

$$c_t = \begin{cases} \dfrac{\gamma^t\left(\gamma^t + \gamma^{t+1} - 2\gamma^T\right)}{1-\gamma} & \text{if } \gamma < 1 \\ 2T - 2t - 1 & \text{if } \gamma = 1 \end{cases}$$

Moreover, particular attention must be paid to the Rényi divergence, which cannot be directly computed. In fact, it requires the approximation of an integral over the trajectory space and knowledge of the transition model P, which is not provided in a model-free setting. Thus, it can be estimated from samples as follows:

$$\hat{d}_2(p_{\boldsymbol{\theta}'}\|p_{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{l=0}^{t}\frac{\pi_{\boldsymbol{\theta}'}\left(a_l^{(i)}|s_l^{(i)}\right)}{\pi_{\boldsymbol{\theta}}\left(a_l^{(i)}|s_l^{(i)}\right)}\right)^2.$$

From these elements, finally, it is possible to define the surrogate objective function, in which the estimated performance $\hat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ is penalized by a function of its variance bound:

$$\mathcal{L}_\lambda(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{t=0}^{T-1}\gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) r\left(s_t^{(i)}, a_t^{(i)}\right)}_{\hat{J}(\boldsymbol{\theta}'/\boldsymbol{\theta})} - \lambda\underbrace{\sqrt{\frac{1}{n}\sum_{t=0}^{T-1} c_t \hat{d}_2\left(p_{\boldsymbol{\theta}'}\|p_{\boldsymbol{\theta}}\right)}}_{\text{penalization}},$$

where $\lambda = R_{\max}\sqrt{(1-\delta)/\delta}$ is a regularization parameter. At each *online* iteration j, the current policy $\pi_{\boldsymbol{\theta}_0^j}$ collects n trajectories interacting with the environment. These are used to update $\mathcal{L}_\lambda(\boldsymbol{\theta}'/\boldsymbol{\theta})$, according to the above formulation. Then, at each offline iteration k, the parameters are updated via gradient ascent:

$$\boldsymbol{\theta}_{k+1}^j = \boldsymbol{\theta}_k^j + \alpha_k \mathcal{G}(\boldsymbol{\theta}_k^j)^{-1}\nabla_{\boldsymbol{\theta}_k^j}\mathcal{L}(\boldsymbol{\theta}_k^j/\boldsymbol{\theta}_0^j),$$

where $\alpha_k > 0$ is the step size determined through a line search and $\mathcal{G}(\boldsymbol{\theta}_k^j)$ is a positive semi-definite matrix, e.g. the Fisher Information Matrix.

Chapter 4

Policy Optimization via Optimal Policy Evaluation

In the previous chapter, several state-of-the-art algorithms were presented, stressing the relevance of the IS estimator in their final formulation. However, IS was employed as a *what-if* analysis tool with a *passive role*, as samples had already been collected with the current behavioral policy. Therefore, the trust region could be considered an *a-posteriori* remedy for the uncertainty injected by the IS procedure. Nevertheless, as widely discussed in Chapter 3, IS originated in the Monte Carlo simulation community as an *active* tool for variance minimization. In the following sections, the relation between Off-VM and Off-PL will be investigated, showing how the repeated identification of the minimum-variance behavioral policy can result in a tool for policy improvement.

Chapter Outline

The chapter is organized as follows. Section 4.1 introduces the problem of the minimum-variance behavioral distribution, outlining the three main research questions of this thesis. Section 4.2, assuming no restriction on the class of distributions, focuses on how such distribution can be employed in off-policy learning settings. More specifically, it shows that the repeated search for the minimum-variance behavioral distribution leads to a performance improvement, converge of the process and the enforcement of an implicit trust region. Section 4.3 covers the same steps, but constraining the class of distributions according to the transition model of the MDP. Finally, Section 4.4 presents the pseudocode of PO^{2}PE, a novel algorithm that exploits variance-minimization for off-policy learning.

4.1 Minimum-variance behavioral distribution

In this section, the problem of finding a minimum-variance IS estimate $\hat{\mu}_{P/Q}$, already introduced in Section 3.2, is revised and further investigated. Let $f \in \mathscr{B}(\mathcal{X}, [0, \infty))$ be a non-negative measurable function, $P \in \mathscr{P}(\mathcal{X})$ the fixed target distribution and $\mathcal{Q} \in \mathscr{P}(\mathcal{X})$ the sought-after behavioral distribution. Note that, in the RL framework, the previous restriction on f is without loss of generality, since it is always possible to define a non-negative reward function, simply translating the original one. Furthermore, at this preliminary stage, no limitations on the possible forms of Q will be enforced. The problem of the minimum-variance estimate and the relative solution are well known in the literature [Kahn, 1950; Kahn and Marshall, 1953], formally stated as:

$$\min_{Q \in \mathscr{P}(\mathcal{X})} \left\{ \underset{x \sim Q}{\mathbb{V}\mathrm{ar}} \left[\frac{p(x)}{q(x)} f(x) \right] \right\} \implies q^*(x) = \frac{p(x) f(x)}{\mathbb{E}_{x \sim P}[f(x)]}, \quad \forall x \in \mathcal{X}. \tag{4.1}$$

The resulting $\hat{\mu}_{P/Q^*}$ estimator is non-stochastic and equal to the quantity to estimate, i.e. $\hat{\mu}_{P/Q^*} = \mathbb{E}_{x \sim P}[f(x)]$. As for Equation (3.4), the construction of the zero-variance distribution Q^* is infeasible, since it would require knowledge of $\mathbb{E}_{x \sim P}[f(x)]$. Moreover, the non-stochasticity of the estimator will induce a simultaneous minimization of the absolute central moments of any order. However, the most remarkable property is that Q^* represents a *performance improvement* with respect to P. In other words, the expectation of f under the minimum-variance behavioral distribution Q^* is larger than the expectation of f under the target distribution P [Owen, 2013]:

$$\underset{x \sim Q^*}{\mathbb{E}}[f(x)] - \underset{x \sim P}{\mathbb{E}}[f(x)] = \frac{\mathbb{V}\mathrm{ar}_{x \sim P}[f(x)]}{\mathbb{E}_{x \sim P}[f(x)]} \geqslant 0. \tag{4.2}$$

In Equation (4.2) it is possible to observe the direct relation that subsists between the magnitude of the improvement, that is the first term, and the reduction in variance $\mathbb{V}\mathrm{ar}_{x \sim P}[f(x)]$. This better motivates what was suggested in Section 3.5, where both natures of IS were compared. More in detail, since there is a connection between finding the minimum-variance behavioral distribution (Off-VM) and finding a target distribution that maximizes $\mathbb{E}_{x \sim P}[f(x)]$ (Off-PL), Off-VM can be employed as a performance improvement tool, exploiting the repeated resolution of the problem stated in Equation (4.1). Indeed, given a target policy and a positive reward function, one way to reduce the variance of the IS estimator is to assign a higher probability to the trajectories that have a large impact on the mean, i.e.

those with high returns. Given these premises, the aforementioned technique leveraging Off-VM for Off-PL will be studied along three different directions:

(Q1) Does this procedure always generate a distribution that entails a *performance improvement?*

(Q2) Does this procedure *converge* to a (local) maximum of f?

(Q3) Can we quantify the divergence between two consecutive distributions, i.e. does this procedure enforce a *trust region*?

In Section 4.2 these questions will be answered under the assumption of no limitations in determining the behavioral distribution, i.e. $Q \in \mathscr{P}(\mathcal{X})$. Instead, in Section 4.3, the same problem will be analyzed confining the choice of Q, i.e. $Q \in \mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$.

4.2 Unconstrained probability distribution space

Equation (4.2) shows that Q^* is a performance improvement with respect to P. However, this formulation can be further generalized by introducing a non-negative monotonic strictly-increasing function $h : [0, \infty) \to [0, \infty)$. Indeed, when f is composed with h, being the latter strictly-increasing, the composition $h \circ f$ presents the same maxima as f. Thus, the *policy improvement operator* $\mathcal{I}_{h \circ f} : \mathscr{P}(\mathcal{X}) \to \mathscr{P}(\mathcal{X})$ can be defined as:

$$(\mathcal{I}_{h \circ f}[P])(x) = \frac{p(x)h(f(x))}{\mathbb{E}_{x \sim P}[h(f(x))]}, \quad \forall x \in \mathcal{X}.$$

Such operator takes as input a target distribution $P \in \mathscr{P}(\mathcal{X})$ and the composition $h \circ f \in \mathscr{B}(\mathcal{X}), [0, \infty))$, returning $Q^* = \mathcal{I}_{h \circ f}[P]$, i.e. the minimum-variance behavioral distribution for the IS estimate of $\mathbb{E}_{x \sim P}[h(f(x))]$.

4.2.1 Performance Improvement

Given the previous definitions, **(Q1)** can be finally answered. In fact, an iterative application of the operator will generate distributions tending to assign larger probability mass to points $x \in \mathcal{X}$ with high values of $f(x)$. Moreover, following [Ghosh et al., 2020], it can be proved that $\mathcal{I}_{h \circ f}[P]$ is a performance improvement with respect to P, whenever h is increasing.

Proposition 4.2.1 (Proposition 9 of [Ghosh et al., 2020])**.** Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0, \infty))$ and $h : [0, \infty) \to [0, \infty)$ monotonic increasing. Then, $\mathcal{I}_{h \circ f}[P]$ is a performance improvement with respect to P:

$$\mathop{\mathbb{E}}_{x \sim \mathcal{I}_{h \circ f}[P]}[f(x)] - \mathop{\mathbb{E}}_{x \sim P}[f(x)] = \frac{\mathbb{C}\mathrm{ov}_{x \sim P}[h(f(x)), f(x)]}{\mathbb{E}_{x \sim P}[h(f(x))]} \geqslant 0.$$

Another consequence of h being monotonic increasing is represented by $\mathbb{C}\mathrm{ov}_{x \sim P}[h(f(x)), f(x)] \geqslant 0$, as stated in [Cuadras, 2002].

4.2.2 Convergence Properties

Established that the operator $\mathcal{I}_{h \circ f}[P]$ involves a performance improvement, now **(Q2)** is addressed, further studying the repeated application of $\mathcal{I}_{h \circ f}[P]$, to determine its convergence properties. Suppose to start from a target distribution $P \in \mathscr{P}(\mathcal{X})$ and an initial behavioral distribution $Q_0 = P$. The iterative application of the operator $\mathcal{I}_{h \circ f}$ will generate a sequence of distributions $(Q_k)_{k \in \mathbb{N}}$ such that $Q_k = \mathcal{I}_{h \circ f}[Q_{k-1}] = (\mathcal{I}_{h \circ f})^k[P]$, for every $k \in \mathbb{N}_{\geqslant 0}$.

Theorem 4.2.2. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0, \infty))$ and $h : [0, \infty) \to [0, \infty)$ monotonic strictly increasing. Then, the following statements hold:

(i) P is a fixed point of $\mathcal{I}_{h \circ f}$, i.e. $\mathcal{I}_{h \circ f}[P] = P$ iff $\mathbb{V}\mathrm{ar}_{x \in P}[f(x)] = 0$;

(ii) let $\mathcal{X}^* = \mathrm{argmax}_{x \in \mathrm{supp}(P)}\{f(x)\}$ be the set of maxima of f restricted to the support of P. If $\mathcal{X}^*$ is non-empty and measurable, the repeated application of $\mathcal{I}_{h \circ f}$ converges to a distribution $Q_\infty = \lim_{k \to \infty} (\mathcal{I}_{h \circ f})^k[P]$ with support $\mathcal{X}^*$. In particular:

$$\mathop{\mathbb{E}}_{x \sim Q_\infty}[f(x)] = \max_{x \in \mathrm{supp}(P)}\{f(x)\}.$$

We recall that the support of P is the set of all points to which P assigns a non-zero probability. In other words, Theorem 4.2.2 states that, under certain conditions, the operator $\mathcal{I}_{h \circ f}$ admits fixed points and the sequence $(Q_k)_{k \in \mathbb{N}}$ converges to a distribution Q_∞. A first, important, observation is that none of the previous properties depends on function h, as long as it respects the assumption of being non-negative and monotonically-increasing. Moreover, following point (i), any deterministic policy will be a fixed point of $\mathcal{I}_{h \circ f}$. Finally, according to point (ii), we are performing a global optimization of f. Indeed, if a distribution P that assigns non-zero probability to all points in $\mathcal{X}$ is selected, i.e. $\mathrm{supp}(P) = \mathcal{X}$, the iterated application of $\mathcal{I}_{h \circ f}$ converges to the distribution Q_∞, such that $\mathbb{E}_{x \sim Q_\infty}[f(x)] = \max_{x \in \mathcal{X}}\{f(x)\}$.

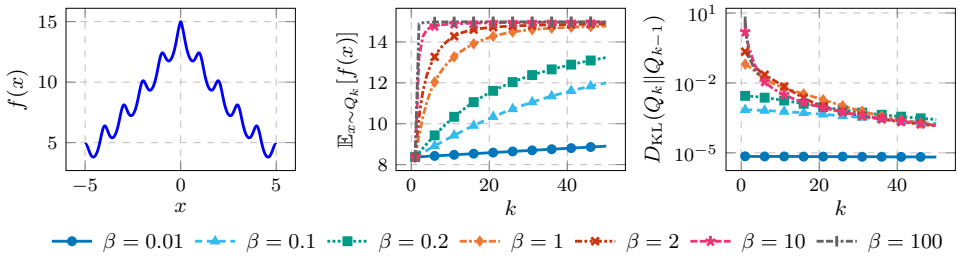

Figure 4.1: The Ackley function (left), the expectation of the distribution $Q_k = (\mathcal{I}_{h\circ f})^k[P]$ (center) and the KL-divergence (right) between two consecutive distributions Q_{k-1} and Q_k, with $h = (\cdot)^\beta$.

4.2.3 Implicit Trust Region

In the previous sections, it was shown how the optimization of function f can be cast to an iterative procedure that yields a performance improvement. The rationale behind this formulation lies in **(Q3)**.

Theorem 4.2.3. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Then, for every $a \in [0,\infty]$, it holds that:

$$D_\alpha(\mathcal{I}_{h\circ f}[P]\|P) = \frac{1}{\alpha-1}\log\frac{\mathbb{E}_{x\sim P}[h(f(x))^\alpha]}{\mathbb{E}_{x\sim P}[h(f(x))]^\alpha}.$$

This means that it is possible to naturally control the divergence between two consecutive distributions Q_k and $Q_{k+1} = \mathcal{I}_{h\circ f}[Q_k]$, resulting in the enforcement of an *implicit* trust region. A particular case is obtained for $\alpha = 1$:

$$D_{KL}(\mathcal{I}_{h\circ f}[P]\|P) = \frac{\mathbb{C}\mathrm{ov}_{x\sim P}[h(f(x)), \log h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]}.$$

Instead, for $\alpha = 2$ it holds that:

$$D_2(\mathcal{I}_{h\circ f}[P]\|P) = \log\frac{\mathbb{E}_{x\sim P}[h(f(x))^2]}{\mathbb{E}_{x\sim P}[h(f(x))]^2} \leqslant \frac{\mathbb{V}\mathrm{ar}_{x\sim P}[h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]^2}.$$

As can be observed from these equations, the divergence between P and $\mathcal{I}_{h\circ f}$ is large when the variance of $h(f(x))$ is. Furthermore, as stated in the theorem, it is possible to control the Rényi divergences of *any* order $\alpha \in [0,\infty)$. This is a remarkable result, since the commonly used trust regions, like KL-divergence [Schulman et al., 2015], are unable to control higher-order divergences, that can still be infinite. Finally, it can be noticed that the increasing function h plays the role of a *regularizer*, controlling the width of the trust region.

Example Let f be (a slight variation of) the one-dimensional Ackley function [Ackley, 2012], defined as:

$$f(x) = -5 + 20\exp(-0.1414|x|) + \exp(0.5(\cos(2\pi x) + 1)) + e.$$

Moreover, let $(h \circ f)(x) = f(x)^\beta$ be the class of increasing functions, assuming $\beta \geqslant 0$, and $P = \text{Uni}([-5, 5])$ an initial uniform distribution. Figure 4.1, on the left, shows a graphical representation of function f, while, in the middle, displays the expectation of the distribution $Q_k = (\mathcal{I}_{h\circ f})^k[P]$. From the latter plot, it can be observed that convergence to the global optimum ($x^* = 0$ and $f(x^*) = 15$) if faster for high powers of β. Finally, the graph on the right illustrates the KL-divergence between two consecutive distributions Q_{k-1} and Q_k, as a function of the number of applications k and for different β values in $h = (\cdot)^\beta$. It reveals that a faster convergence is also related to larger trust regions.

4.3 Constrained probability distribution space

In Section 4.2, the three questions were addressed without enforcing any restriction on the class of distributions that can be played, allowing a free choice of Q in the whole space $\mathscr{P}(\mathcal{X})$. This analysis can be easily extended to multi-armed bandit problems [Slivkins, 2019], where the agent is described by a single state and a set of actions, of which it aims to estimate the rewards. Since any distribution over the actions can be played, there is no need for further considerations. Unfortunately, this is no longer true for the classical MDP problem, in which the trajectory distributions must be subject to the transition model P. As a consequence of this, when a class of distributions $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ is considered, the distribution $\mathcal{I}_{h\circ f}[P]$ might not belong to $\mathcal{Q}$, even if $P \in \mathcal{Q}$. Moreover, recalling from Section 4.1 that $\mathcal{I}_{h\circ f}$ generates a non-stochastic estimator which minimizes all the absolute central α-moments, there may exist multiple different distributions in $\mathcal{Q}$ complying with this property:

$$\min_{Q\in\mathcal{Q}} \left\{ \mathop{\mathbb{E}}_{x\sim Q} \left[\left| \frac{p(x)}{q(x)} h(f(x)) - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))] \right|^\alpha \right] \right\}. \tag{4.3}$$

It can be observed that, for $\alpha = 2$, Equation (4.1) reduces to Equation (4.3). However, apart from this particular case, for general values of $\alpha \in [0, \infty]$ the optimization might not be so straightforward. Indeed, Equation (4.3) is not differentiable in the interval $\alpha \in (0, 2)$. A solution to this problem can be obtained by performing a *moment projection* through the α-Rényi

divergence. Such projection is a reasonable surrogate for minimizing the absolute central α-moments of Equation (4.3), as shown by the following proposition.

Proposition 4.3.1. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Then, for every $a \in (1,\infty)$, it holds that:

$$\underbrace{\mathop{\mathbb{E}}_{x\sim Q}\left[\left|\frac{p(x)}{q(x)}h(f(x)) - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]\right|^{\alpha}\right]}_{\text{absolute central } \alpha\text{-moment}} \leqslant \underbrace{\mathop{\mathbb{E}}_{x\sim Q}\left[\left(\frac{p(x)}{q(x)}h(f(x))\right)^{\alpha}\right]}_{\text{(non-central) } \alpha\text{-moment}}.$$

Moreover, the second term can also be expressed as:

$$\mathop{\mathbb{E}}_{x\sim Q}\left[\left(\frac{p(x)}{q(x)}h(f(x))\right)^{\alpha}\right] = e^{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q)} \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha}.$$

Thus, combining the two partial results, the property can be finally stated:

$$\mathop{\mathbb{E}}_{x\sim Q}\left[\left|\frac{p(x)}{q(x)}h(f(x)) - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]\right|^{\alpha}\right] \leqslant e^{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q)} \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha}.$$

Since the subset of distributions $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ has been considered, whenever $\mathcal{I}_{h\circ f}[P] \notin \mathcal{Q}$, the result of the operator can be replaced by the corresponding moment projection, perfromed through the α-Rényi divergence:

$$Q^{\dagger} \in \operatorname*{arg\,min}_{Q\in\mathcal{Q}} \left\{D_\alpha\left(\mathcal{I}_{h\circ f}[P]\|Q\right)\right\}.$$

4.3.1 Performance Improvement

Assuming an unconstrained scenario, Proposition 4.2.1 states that, whenever h is strictly-increasing, $\mathcal{I}_{h\circ f}[P]$ is a performance improvement with respect to P under function f. This can also be extended to the composition between f and any strictly-increasing function. However, when a subset $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ is considered, the guarantee of a performance improvement is no longer always valid for f. Therefore, the previous proposition can be adapted to the constrained case considering a *specific* monotonic transformation of f, which depends on h and α.

Theorem 4.3.2. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$, $Q \in \mathcal{Q}$ and $\alpha \in [0,\infty]$, it holds that:

$$\mathop{\mathbb{E}}_{x\sim Q}[h(f(x))^\alpha] - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))^\alpha] \geqslant \frac{\mathbb{E}_{x\sim P}[h(f(x))]^\alpha}{\alpha - 1}\left(e^{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[P]\|P)} - e^{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q)}\right).$$

In particular, for $\alpha = 1$ (Proposition 6 of [Ghosh et al., 2020]):

$$\mathop{\mathbb{E}}_{x\sim Q}[h(f(x))] - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))] \geqslant \mathop{\mathbb{E}}_{x\sim P}[h(f(x))](D_{KL}(\mathcal{I}_{h\circ f}[P]\|P) - D_{KL}(\mathcal{I}_{h\circ f}[P]\|Q)).$$

This answers **(Q1)**, because, according to the theorem, if the α-moment of the transformed function $h \circ f$ is minimized, a performance improvement can be guaranteed on the function $(\cdot)^\alpha \circ h \circ f$. Such result holds under the assumption that:

$$D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q) \leqslant D_\alpha(\mathcal{I}_{h\circ f}[P]\|P). \tag{4.4}$$

However, Equation (4.4) is always verified when $P \in \mathcal{Q}$ and $Q = Q^\dagger$, since $Q^\dagger$ was defined as the minimizer of the second divergence term. A particular choice of h is $h = (\cdot)^{1/\alpha}$, for which the theorem guarantees the performance improvement for the function f directly. In all other cases, it can be guaranteed for a monotonic transformation of f only.

4.3.2 Convergence Properties

This section will attempt to answer **(Q2)**, taking into account the additional constraint due to the transition model. A sequence of distributions $(Q_k)_{k\in\mathbb{N}}$ can still be obtained using Equation (4.3) as an iterate:

$$Q_{k+1} \in \arg\min_{Q\in\mathcal{Q}}\{D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q)\}.$$

Nevertheless, in contrast to the unconstrained setting, the guarantee of convergence to any fixed-point distribution Q_∞ does not hold anymore. This is caused by the minimization step, which might yield multiple solutions, as it was discussed for Equation (4.3). Thus, theoretical guarantees can only be provided in terms of the final divergence value.

Theorem 4.3.3. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ and suppose that $h \circ f$ is bounded from above. The iterate $Q_{k+1} \in \arg\min_{Q\in\mathcal{Q}}\{D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q)\}$, where possible ties are broken arbitrarly, satisfies:

(i) the sequence of divergences $D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_k)$ is convergent;

(ii) the sequence of expectations $\mathbb{E}_{x\sim Q_k}[h(f(x))^\alpha]$ is non-decreasing in $k \in \mathbb{N}$ and converges to a stationary point of $\mathbb{E}_{x\sim Q}[h(f(x))^\alpha]$ with respect to $Q \in \mathcal{Q}$.

Note that, both for (i) and (ii), Theorem 4.3.2 can be employed to derive the relative sequences $D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_k)$ and $\mathbb{E}_{x\sim Q_k}[h(f(x))^\alpha]$. This theorem highlights that, even in the constrained scenario, convergence to a *stationary point* $\mathbb{E}_{x\sim Q}[h(f(x))^\alpha]$ is achieved. Moreover, if $\mathcal{Q}$ is a parametric space, following the notation introduced in section 2.7, it can be defined as:

$$\mathcal{Q}_\Theta = \{Q_{\boldsymbol{\theta}} \in \mathscr{P}(\mathcal{X}) : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d\}.$$

As for a general PG method maximizing $h(f(x))^\alpha$ [Papini et al., 2019], stopping is guaranteed when:

$$\mathop{\mathbb{E}}_{x\sim Q_{\boldsymbol{\theta}}}\left[\nabla_{\boldsymbol{\theta}} \log q_{\boldsymbol{\theta}}(x) h(f(x))^\alpha\right] = 0.$$

Therefore, the convergence to a fixed point is lost constraining the distribution space. However, this property is still valid under the assumption that the iterate in Equation (4.3) admits a unique solution for every P. In such a case, the procedure will converge to a distribution

$$Q_\infty = \arg\min_{Q\in\mathcal{Q}}\{D_\alpha(\mathcal{I}_{h\circ f}[Q]\|Q)\}.$$

4.3.3 Implicit Trust Region

Finally, **(Q3)** will be analyzed in light of the new scenario. In Section 4.2.3, it was proved that the α-Rényi divergence between $\mathcal{I}_{h\circ f}[P]$ and P is bounded. In the constrained case it is possible to control the trust region as well, under a particular form of convexity [van Erven and Harremoës, 2014] of Q. This is formalized by the following theorem.

Theorem 4.3.4. Let $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ be a $(1-\alpha)$-convex set (Definition 4 of

[van Erven and Harremoës, 2014]), $P \in \mathcal{Q}$, $Q^\dagger \in \arg\min_{Q \in \mathcal{Q}}\{D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q)\}$ and $\alpha \in [0, \infty]$, then it holds that:

$$D_\alpha(Q^\dagger\|P) \leqslant D_\alpha(\mathcal{I}_{h\circ f}[P]\|P) - D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q^\dagger).$$

More in detail, it is always guaranteed that the trust region induced by $Q^\dagger$ is tighter with respect to the one induced by $Q^* = \mathcal{I}_{h\circ f}[P]$, which was computed in Theorem 4.2.3. Therefore, it follows that:

$$D_\alpha(Q^\dagger\|P) \leqslant D_\alpha(\mathcal{I}_{h\circ f}[P]\|P).$$

4.4 PO^{2}PE

After having introduced the preliminary concepts of PO and IS, paying special attention to the problem of the minimum-variance behavioral distribution, finally, *Policy Optimization via Optimal Policy Evaluation* (PO^{2}PE) can be presented. PO^{2}PE is a novel PO algorithm, based on the intuition of repeatedly applying Off-VM as a tool for Off-PL. Following such approach, PO^{2}PE is able to both improve its performances and to implicitly control the divergence between two consecutive policies, enforcing a trust region without any need for divergence constraints. For generality of presentation, PO^{2}PE will be particularized for a parametric policy space $\mathcal{Q}_\Theta$, defined as in Section 2.7:

$$\mathcal{Q}_\Theta = \{Q_{\boldsymbol{\theta}} \in \mathscr{P}(\mathcal{X}) : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d\}.$$

In the next sections, it will be better explained how the theoretical results of Section 4.2 and Section 4.3 can be translated into the pseudocode of Algorithm 4.1.

Sample Collection

Let $q_{\boldsymbol{\theta}} = p(\cdot|\boldsymbol{\theta})$ be the trajectory distribution induced by the parametrized policy $\pi_{\boldsymbol{\theta}}$, f the trajectory return $R(\tau)$ and i a generic and fixed iteration of the outer loop. At each inner iteration $j \in [J]$, the current behavioral policy $\pi_{\bar{\boldsymbol{\theta}}_{i,j}}$ is employed to independetly collect n trajectories $\{\tau_l\}_{l\in[n]}$. This sampling phase leads to the creation of a dataset $\mathcal{D}_{i,j} = \{(\tau_l, \mathcal{R}(\tau_l))\}_{l\in[n]}$, consisting of pairs that associate each trajectory to the return gathered along them. To efficiently exploit the observed data, the dataset $\mathcal{D}_{i,j}$ will not be discarded at the end of iteration j. Instead, it will be saved in a *memory* and later employed to perform an off-policy estimate of the next

Algorithm 4.1 PO^{2}PE

Input: divergence order α, function h, function f, distribution space $\mathcal{Q}_{\boldsymbol{\theta}}$ initial parameter $\boldsymbol{\theta}_1 \in \Theta$, batch size n
Output: final parameter $\boldsymbol{\theta}_{I+1} \in \Theta$

for $i = 1, \dots, I$ **do** Optimization
  $\bar{\boldsymbol{\theta}}_{i,1} = \boldsymbol{\theta}_i$
  for $j = 1, \dots, J$ **do** Evaluation
    Collect n samples $\mathcal{D}_{i,j} = \{(x_l, f(x_l))\}_{l \in [n]}$ with $Q_{\bar{\boldsymbol{\theta}}_{i,j}}$
    Find $\bar{\boldsymbol{\theta}}_{i,j+1}$ by minimizing $D_\alpha(\mathcal{I}_{h \circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}})$ using $(\mathcal{D}_{i,k})_{k \in [j]}$
  end for
  $\boldsymbol{\theta}_{i+1} = \bar{\boldsymbol{\theta}}_{i,J+1}$
end for

behavioral policy $\bar{\boldsymbol{\theta}}_{i,j+1}$. Such memory is modeled after a hyperparameter C, namely the *capacity*, that determines the number of behavioral policies that can be stored. Until $j \leqslant C$, new behavioral policies can be added to the memory without any concern, while an alternative strategy has to be chosen when the capacity is exceeded. Intuitively, since the inner loop aims to repeatedly minimize the absolute central α-moment, the lastest C policies are the most significant ones. This consideration results in a *first-in first-out* implementation, which discards the oldest policies to leave room for the newest ones. However, in Section 5.2, it will better discussed how, from empirical evidence, a single inner iteration seems to be a good enough approximation.

Sample-based Optimization

In the previous section, it was mentioned that the samples collected so far $(\mathcal{D}_{i,k})_{k \in [j]}$ are saved in a memory and later exploited for off-policy learning. In this section, the problem of finding the next behavioral distribution parameter $\bar{\boldsymbol{\theta}}_{i,j+1}$ will be discussed in more detail. According to Algorithm 4.1, the successive $\bar{\boldsymbol{\theta}}_{i,j+1}$ has to be determined through the minimization of $D_\alpha(\mathcal{I}_{h \circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}})$. Instead of a direct estimate of this quantity, Proposition 4.3.1 can be employed to refer to the (non-central) α-moment.
Furthermore, recalling the notion of *mixture* from Section 3.3, the mixture

of the distributions experienced so far in the inner loop can be defined as follows:

$$\Phi_{i,j} = \frac{1}{j} \sum_{k \in [j]} q_{\boldsymbol{\theta}_{i,k}}.$$

Therefore, since samples coming from different behavioral distributions were preserved, an MIS estimator can be used:

$$\hat{d}_\alpha(\mathcal{I}_{h \circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}}; \Phi_{i,j}) = \frac{1}{nj} \sum_{k \in [j]} \sum_{l \in [n]} \underbrace{\frac{q_{\boldsymbol{\theta}}(x_{k,l})}{\Phi_{i,j}(x_{k,l})}}_{\text{(a)}} \underbrace{\frac{q_{\boldsymbol{\theta}_i}(x_{k,l})^\alpha}{q_{\boldsymbol{\theta}}(x_{k,l})^\alpha} h(f(x))^\alpha}_{\text{(b)}}. \quad (4.5)$$

Such estimator is composed of two terms, referred as (a) and (b) in the above formula. The first takes into account the fact that a mixture $\Phi_{i,j}$ is being used to estimate an expectation under $q_{\boldsymbol{\theta}}$. Instead, the second one, is the α-moment, that is the variable whose expected value is to be estimated. The equivalence between $\hat{d}_\alpha$ and the α-moment can be easily proved, as shown in [Papini et al., 2019]. To perform the minimization of Equation (4.5), a variance correction is introduced to mitigate the effect of finite samples [Metelli et al., 2018], theoretically grounded in the following result.

Theorem 4.4.1. Let $\mathcal{Q}_\Theta \subseteq \mathscr{P}(\mathcal{X})$ be a set of parametric distributions and let $\boldsymbol{\theta}, \boldsymbol{\theta}_i \in \Theta$. If $\|h \circ f\|_\infty \leqslant \overline{m}$, then, if all samples are independent, for every $\delta \in [0, 1]$, with probability at least $1 - \delta$ it holds that:

$$\mathop{\mathbb{E}}_{x \sim \boldsymbol{\theta}} \left[\left(\frac{q_{\boldsymbol{\theta}_i}(x)}{q_{\boldsymbol{\theta}}(x)} h(f(x)) \right)^\alpha \right] \leqslant \widehat{d}_\alpha \left(\mathcal{I}_{h \circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}}; \Phi_{i,j} \right) + \overline{m}^\alpha \sqrt{\frac{2 \log \frac{1}{\delta}}{nj} \int_{\mathcal{X}} \frac{q_{\boldsymbol{\theta}_i}(x)^{2\alpha}}{\Phi_{i,j}(x) q_{\boldsymbol{\theta}}(x)^{2(\alpha-1)}} \mathrm{d}x},$$

where the integral within the square root is an upper bound to the variance of the α-moment estimator $\widehat{d}_\alpha \left(\mathcal{I}_{h \circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}}; \Phi_{i,j} \right)$. When all involved distributions are Gaussians, a closed-form tight bound on this quantity can be provided (Appendix A). Furthermore, when $\boldsymbol{\theta} = \boldsymbol{\theta}_i$, it reduces to the exponentiated Rényi divergence [Metelli et al., 2020]. Recalling that, in probability theory, *concentration inequalities* [Boucheron et al., 2003] deal with deviations of functions of independent random variables from their expectation, this formalism can be extended to the previous formulation. Differently from results available in the literature about the concentration of the IS estimator, it provides an exponential concentration inequality (dependence on delta of the form $\log(1/\delta)$), instead of a polynomial concentration (dependence of the form $1/\delta$). This is due to the fact that the considered

random variables are bounded to zero from below, allowing the application of stronger Bernestein's concentration inequalities [Boucheron et al., 2009].

For the sake of notational simplicity, let the bound derived in Theorem 4.4.1 be denoted by:

$$\mathcal{L}_i = \widehat{d}_\alpha \left(\mathcal{I}_{h\circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}}; \Phi_{i,j}\right) + \overline{m}^\alpha \sqrt{\frac{2\log\frac{1}{\delta}}{nj} \int_{\mathcal{X}} \frac{q_{\boldsymbol{\theta}_i}(x)^{2\alpha}}{\Phi_{i,j}(x) q_{\boldsymbol{\theta}}(x)^{2(\alpha-1)}} \mathrm{d}x}.$$

Once it has been computed, it can be employed for the *offline optimization*. This procedure consists of a loop of k iterations, aiming to effectively update the parameters. Indeed, following the classical PG approach, the direction of maximum growth can be obtained through the gradient of $\mathcal{L}_i$. Therefore, the policy update formula is given by:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha \cdot \epsilon \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_i,$$

where α is defined as $\alpha = 1/\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_i\|$ and ϵ is initialized to 1. The product $\alpha \cdot \epsilon$ realizes a normalized constant step size, which balances the tradeoff between fast convergence and the stability of the updates. In fact, whenever the step size is too large, the value of ϵ is halved until a correct update is performed.

Evaluation and Optimization

After the detailed analysis of sample collection and sample-based optimization, it is now possible to describe a high-level picture of the algorithm. At each outer iteration i, $q_{\boldsymbol{\theta}_i}$ represents a target distribution of which the inner loop aims at performing the `Evaluation` of the performance. To achieve this result, for every inner iteration $j \in [J]$, sample collection and sample-based optimization are both involved. At first $\mathcal{D}_{i,j}$ is sampled with the current behavioral distribution $q_{\bar{\boldsymbol{\theta}}_{i,j}}$, then it is employed with all data collected so far $(\mathcal{D}_{i,k})_{k\in[j]}$ to compute the next behavioral distribution $q_{\bar{\boldsymbol{\theta}}_{i,j+1}}$. As already discussed, this process has the goal of minimizing the absolute central α-moment. The outer loop, instead, aims to perform the `Optimization` of the target distribution $q_{\boldsymbol{\theta}_i}$. Finally, at the end of each outer iteration $i \in [I]$, the target distribution $q_{\boldsymbol{\theta}_{i+1}}$ is updated with the last behavioral distribution $q_{\bar{\boldsymbol{\theta}}_{i,J+1}}$ by the inner loop.

Chapter 5

Experimental Results

As already discussed in the previous chapters, the idea of employing Off-VM as a tool for Off-PL is a novelty in the context of RL. In the literature, the most related works [Hanna et al., 2017; Hanna and Stone, 2018; Hanna et al., 2019] propose to find the optimal behavioral policy for evaluation, in order to subsequently combine it with PG learning. However, they treat Off-VM and Off-PL separately. As a consequence, no algorithm shares the same properties of PO^{2}PE and can be directly compared to it. Nevertheless, being PO^{2}PE a PO algorithm, we can still make a comparison with other methods belonging to this family. Among them, TRPO and POIS are chosen as significant benchmarks for the experimental phase. The former, indeed, besides being a state-of-the-art method, exploits an explicitly defined trust region. Therefore, testing PO^{2}PE and TRPO on different scenarios can shed light on the effects of PO^{2}PE's implicit one. The latter, instead, shares many characteristics with PO^{2}PE, such as the rationale behind the bound derivation and the offline optimization. Naturally, the main difference consists in the reversed roles of behavioral and target policies. Furthermore, additional empirical simulations are conducted to deepen the consequences of some theoretical properties, such as modeling the transformation function h and reducing the batch size n.

Chapter Outline

The chapter is organized as follows. Section 5.1 presents, through plots and observations, the experimental results of testing PO^{2}PE, POIS and TRPO on the most common RL benchmarks. In Section 5.2, their performances are analyzed after a progressive reduction of the batch size. Section 5.3 is devoted to the effects of using a power function as transformation function. Finally, Section 5.4 shows the potentialities of the PDIS estimator.

5.1 Benchmarks

In the following paragraphs, we provide a brief description of the environments in which PO^{2}PE was tested and a comparison with TRPO and POIS. All the experiments have been carried out employing Gaussian linear policies, with mean linear in the state variables and constant variance uniform over the state space. This latter choice is derived from previous simulations, in which a learnable variance was preventing the algorithm from converging to optimal policies. Thus, variance was treated as a hyperparameter, whose tuning turned out to be quite immediate, since its possible values could be restricted to few candidates (see Appendix B for more details). Another important remark is about environments characterized by negative reward functions. Indeed, as discussed in Chapter 4, an important theoretical requirement for PO^{2}PE is the non-negativity of function f, which is crucial to perform the optimization of the α-moment. To overcome this problem, the observed returns were shifted and transformed into non-negative values. This procedure was carried out independently at each algorithm iteration, by subtracting the minimum return among the collected ones. Since the optimization is performed in terms of expected value of the return, this transformation guarantees that the optimum is left unchanged. Moreover, to ensure a fair comparison with TRPO and POIS, the performance and other metrics were computed starting from the original value of returns.

Cartpole

The *Cartpole balancing* is one of the most common RL benchmarks [Barto et al., 1983], here considered in its continuous-action variant. The problem consists of a cart which can horizontally move along a track, aiming to keep balanced the pole in a vertical position. Such a system has to obey the physics laws, given that the cart has a mass of 1 kg and the pole has a mass of 0.1 kg and is 0.5 m long. The dynamics can be described through the cart's position x, the pole's angle θ and their derivatives, respectively representing the cart's speed and pole's angular velocity. Clearly, when the cart runs off the track or the angle becomes excessive, the episode terminates in a failure. Instead, for each step in which the pole was correctly balanced, a reward of 10 will be provided. Therefore, fixed a horizon of 500 time steps, the ideal (undiscounted) return is 5000. Bearing in mind this optimal value, we can better analyze the performances of the algorithms in the Cartpole environment, as shown in Figure 5.1. It is immediately evident that both PO^{2}PE and POIS converge to the optimal policy, succeeding in balancing

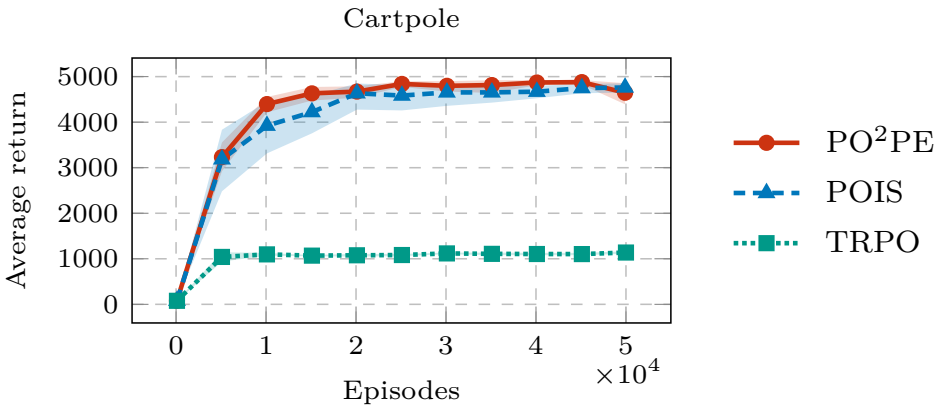

Figure 5.1: Average return as a function of the number of episodes for the Cartpole environment. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (20 runs $\pm$ 95% boostrapped c.i.)

the pole for all the time steps. However, PO^{2}PE manages to achieve this result faster and proving to be very stable across different runs. As it will be reiterated later, this is a key property of the algorithm, which exploits the induced implicit trust region to avoid hazardous updates. More importantly, this behavior is confirmed even with a drastic reduction of the batch size n, as will be illustrated in detail in Section 5.2. Furthermore, PO^{2}PE significantly outperforms TRPO, which converges to a suboptimal policy and might suffer the usage of linear policies.

Mountain Car

The *Mountain Car* environment [Moore, 1991] is characterized by an underpowered car in the middle of a valley, limited on both sides by a hill. The goal of the agent is to learn how to properly climb the hill on the right, given that the engine is not enough to overcome gravity. To achieve this result, the agent can exploit the hill on the left to swing up and down, building momentum. Therefore, the problem consists in learning how to leverage potential energy to reach the target. The state of the agent is described by its position x and speed $\dot{x}$, while the actions represent both forward and backward engine traction. Differently from Cartpole, in this environment the reward function only assumes negative values. Indeed, the reward for a state-action pair is defined as $r(s, a) = 1 - h$, where h is the *height* and 0.6 is the altitude of the target. Since PO^{2}PE requires a non-negative reward function, as previously discussed, it was tested shifting the returns by subtracting their minimum observed value. However, we remind that this

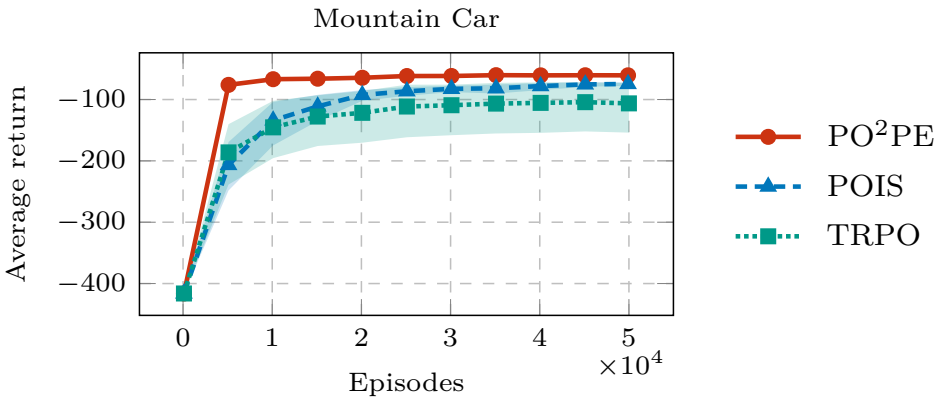

Figure 5.2: Average return as a function of the number of episodes for the Mountain Car environment. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (20 runs $\pm$ 95% boostrapped c.i.)

transformation only affects the optimization problem, leaving untouched performances and metrics. Thus, in Figure 5.2, we can evaluate the effects of this choice, along with the performances of the other two baselines. Even in this scenario, PO^{2}PE confirms its main property, showing once more a very limited variance across a consistent number of runs. This phenomenon can be also observed in POIS, but only when the growth becomes stable. Despite a larger confidence interval, also TRPO reaches a comparable average return. Finally, since all the algorithms converge to a similar final value, the initial steep trend of PO^{2}PE becomes even more desirable.

Inverted Double Pendulum

The *Inverted Double Pendulum* [Furuta et al., 1978] can be seen as a variation of the Cartpole involving a two-link pole, which is composed of two pendulum rods interconnected in a joint. Therefore, the state will be described by the position x of the cart, the joint angles θ_1 and θ_2 of the double pendulum and their derivatives $\dot{\theta_1}$ and $\dot{\theta_2}$. Nevertheless, the goal of the agent is left unchanged, since it should learn how to keep the two-link pole in an upright position, eventually rebalancing it through horizontal forces applied to the cart. Referring to Figure 5.3, the three algorithms can be compared in such a scenario. Even if PO^{2}PE confirms the previous analysis on the variance, this time, what catches the attention is its learning curve. Indeed, unlike in other environments, PO^{2}PE clearly beats both POIS and TRPO, more than doubling their average return.

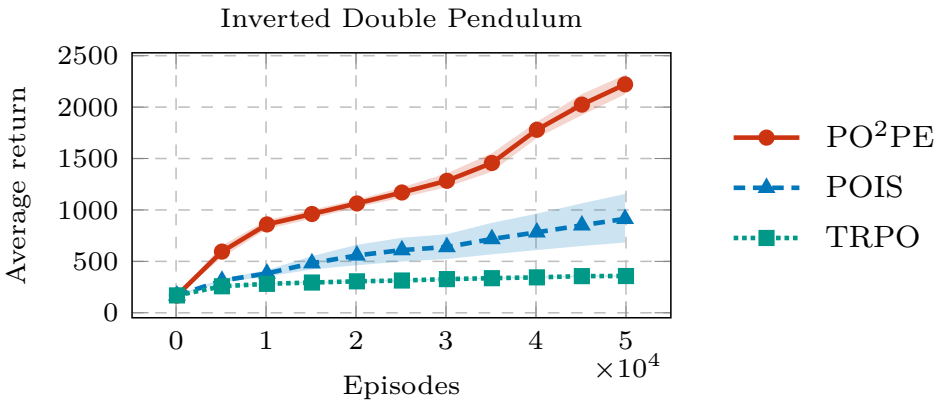

Figure 5.3: Average return as a function of the number of episodes for the Inverted Double Pendulum environment. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (20 runs $\pm$ 95% boostrapped c.i.)

Swimmer

The *Swimmer* [Coulom, 2002] implements a snake robot, composed of three links and two joints, which is immersed in a viscous fluid and should swim forward as fast as possible. Differently from the previous control problems, characterized by a rather small dimensionality, in this environment the state collects the position of the center of mass, the angles and the velocities of the joints in a 13-dimensional vector. As shown in Figure 5.4, despite the higher complexity of this problem, PO^{2}PE remains a reliable choice, converging to the best average return with the fastest curve. POIS is not as competitive as in other environments, while TRPO does not deviate significantly.

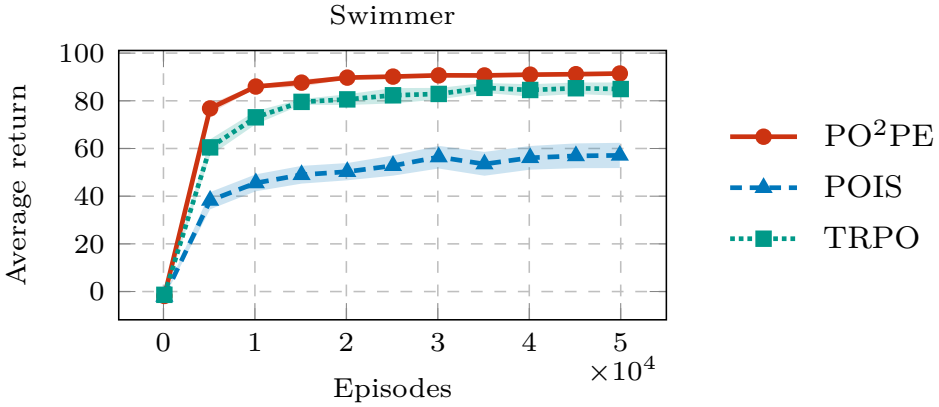

Figure 5.4: Average return as a function of the number of episodes for the Swimmer environment. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (20 runs $\pm$ 95% boostrapped c.i.)

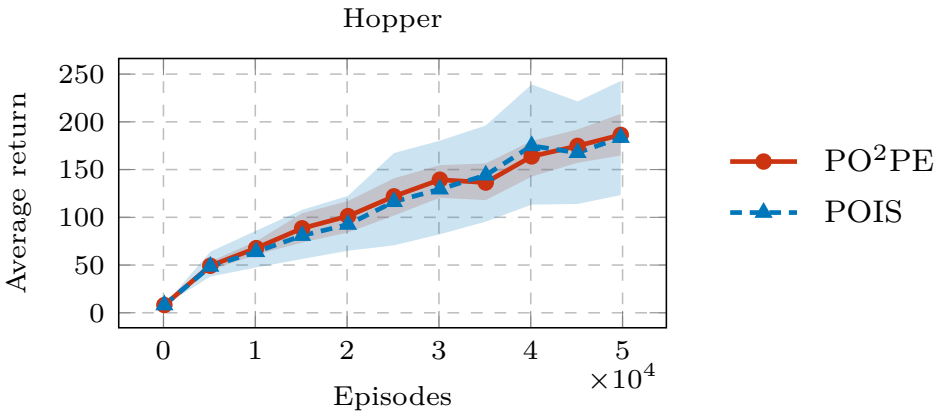

Figure 5.5: Average return as a function of the number of episodes for the Hopper environment. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (10 runs $\pm$ 95% boostrapped c.i.)

Hopper

The *Hopper* [Duan et al., 2016] is a monopod robot, composed of four rigid links (torso, upper leg, lower leg and foot) and three joints, which should hop forward as fast as possible. Differently from the Swimmer, the agent needs to learn a stable hopping gait and avoid falling. This requires a larger amount of exploration, because it can stuck in the suboptimal policy of diving forward. The problem can be described by a 20-dimensional vector, which includes joint angles, joint velocities, the coordinates of the center of mass and constraint forces. The reward function, instead, consists of a bonus for being alive and a positive contribution of the forward velocity. In Figure 5.5, we can see that PO^{2}PE and POIS converge to the same policy. However, once more, variance breaks the tie in favour of PO^{2}PE, even if it displays a larger confidence interval than usual.

Half-Cheetah

The *Half-Cheetah* [Duan et al., 2016] is a biped cheetah robot, composed of nine links (torso, two legs and head) and six joints, which should run forward as fast as possible. Again, the problem is quite complex, since a 20-dimensional vector is required to define joint angles, joint velocities and the coordinates of the center of mass. Similarly to Hopper, the reward depends on the forward velocity. Figure 5.6 confirms the previous analysis, suggesting that complex problems may require a different representation than the one given by linear policies. Indeed, for the first time, PO^{2}PE displays a larger variance than POIS, despite the better trend.

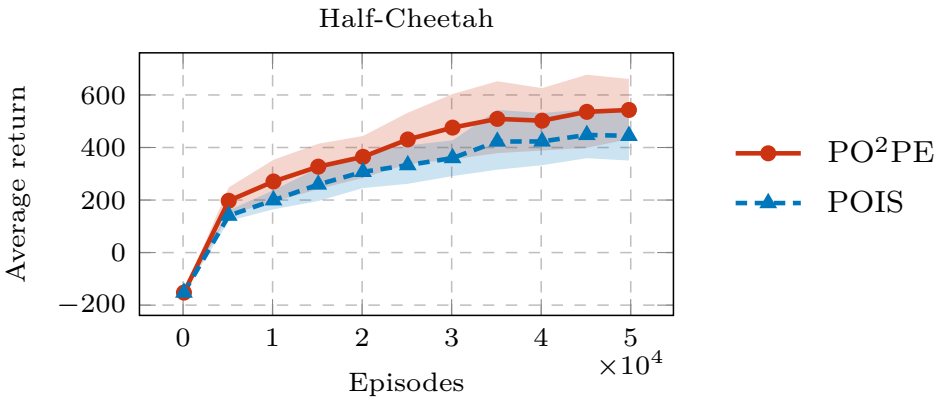

Figure 5.6: Average return as a function of the number of episodes for the Half-Cheetah environment. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (10 runs $\pm$ 95% boostrapped c.i.)

5.2 Robustness to Small Batch Sizes

Since the previous experiments revealed that PO^{2}PE is robust with respect to different runs, showing a small variance across them, this property was further investigated by varying the batch size. Starting from a batch size $n = 100$, its value was progressively reduced to 50 and 11, simulating the agent in the Cartpole environment. Furthermore, these experiments were conducted changing also the hyperparameter J, i.e. the number of inner iterations performed by PO^{2}PE. In Figure 5.1, the curve with $n = 100$ (and $J = 1$ for PO^{2}PE) was already shown for all the algorithms under examination. It is immediately evident that TRPO only converges to a suboptimal policy, even if is not affected by a large variance. Instead, as already discussed in the previous section, POIS and PO^{2}PE both succeed in reaching the optimal behavior. However, PO^{2}PE proves to be a preferable choice pursuant to the very strict confidence interval. Moving to $n = 50$ (Figure 5.7), this factor becomes largely evident and is paired to a relevant drop in average return for POIS, while TRPO gains no benefit or complication from the change. Moreover, a closer look at PO^{2}PE reveals that the value of J does not affect achieving the optimal policy. Instead, it only has consequences on the convergence speed, which slows down for a higher number of inner iterations. This suggests that one inner iteration can determine a good enough estimate to update the target policy. The same considerations can be extended for $n = 11$, where the trend of PO^{2}PE and TRPO is basically unchanged, while POIS suffers even more the reduction of the batch size.

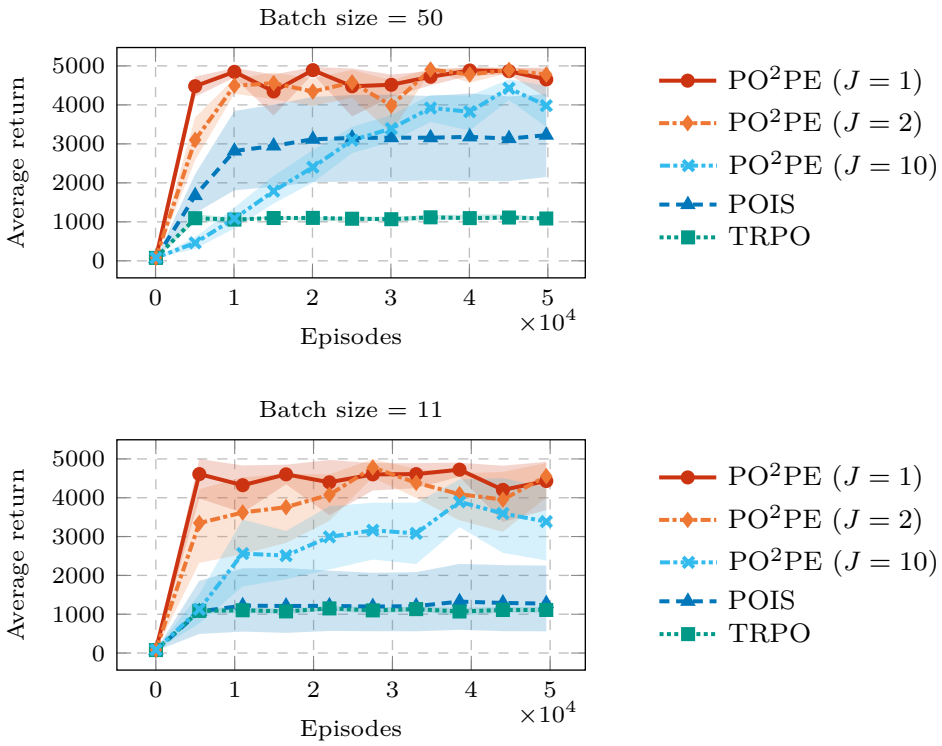

Figure 5.7: Average return as a function of the number of episodes for the Cartpole environment. The algorithms are tested with $\alpha = 2$, h =Id, batch size $n = 50$ (top) and $n = 11$ (bottom). Moreover, multiple values of J are evaluated (10 runs $\pm$ 95% boostrapped c.i.)

5.3 Effect of the Function h

In Chapter 4, PO^{2}PE was introduced as an algorithm consisting of two nested loops, showing how the outer one was devoted to optimization, while the inner one was focused on evaluation. Moreover, during the evaluation phase, the computation of the next behavioral distribution $q_{\bar{\theta}_{i,j+1}}$ was expressed in terms of two hyperparameters: the moment order α and the transformation function h. In this section, the role of the transformation function h will be better discussed. Until now, in all previous experiments, h has been considered to be the *identity function*, i.e. $h = \mathrm{Id}$. This has been a transparent choice, since the algorithm was reaching satisfactory performances and desirable properties, also with respect to TRPO and POIS. However, to further detail the characteristics of PO^{2}PE, the effects of using $h = (\cdot)^\beta$, i.e. the power function, were studied. In particular, four values of β were selected. Two of them, $\beta = 0.5$ and $\beta = 2$, are comparable to the basic scenario, while $\beta = 0.1$ and $\beta = 4$ represent two extreme powers.

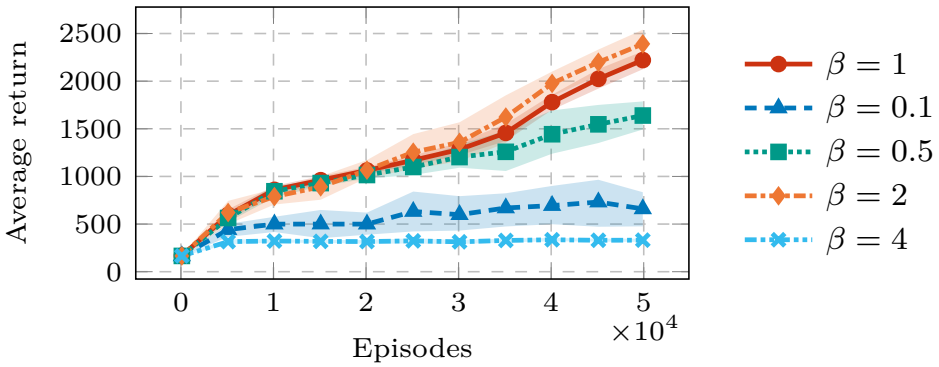

Figure 5.8: Average return of PO^{2}PE as a function of the number of episodes in the Inverted Double Pendulum for different choices of $h = (\cdot)^\beta$ (5 runs $\pm$ 95% boostrapped c.i.)

The results of the experiments, conducted in the Inverted Double Pendulum environment according to the previous criteria, are illustrated in Figure 5.8. The red curve outlines the reference trend, to which the others must be compared. It can be noticed that, for values of β close to 1, the performances are not very dissimilar, or can even be slightly better. This is the case of the orange curve ($\beta = 2$), which succeeds to reach a higher average return, in the face of a small increase in variance. All things considered, also $\beta = 0.5$ displays a similar growth - especially in the first part - struggling to keep up only when the number of episodes exceeds 3×10^4. Instead, extreme powers of β seem to have a negative effect on learning, causing a degradation in performances. Both for $\beta = 0.1$ and $\beta = 4$, the average return fails to reach the previous values and the curve flattens almost immediately. In light of these results, it is clear that modeling h as a power function with the appropriate β can be a useful technique, especially in problems where $h = \mathrm{Id}$ is not achieving optimal returns. Indeed, the optimization of a power of the return, within certain limits, still allows the convergence to a (near-)optimal policy and, eventually, provides better performances.

5.4 PDIS-PO^{2}PE

In all the previous simulations, the repeated identification of the minimum-variance policy was performed resorting to the classical IS estimator. However, as discussed in Section 3.5.2, this approach can be further refined, taking into account the time dependence between rewards and future states and actions. When the reward at time t is only reweighted according to the policy ratios up to that time step, ignoring all the successive ones, we are

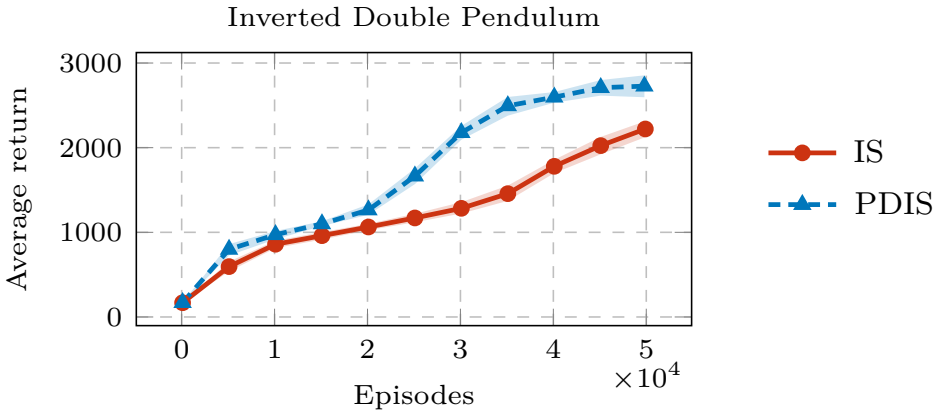

Figure 5.9: Average return as a function of the number of episodes for the Inverted Double Pendulum environment, employing a PDIS estimator. The algorithms are tested with batch size $n = 100$, $\alpha = 2$, h =Id and $J = 1$ (10 runs $\pm$ 95% boostrapped c.i.)

employing the so-called PDIS estimator. Since we are no longer computing a unique importance weight for the whole trajectory, we can expect more accurate estimates than before. Figure 5.9 shows a direct comparison between IS and PDIS estimators on the Inverted Double Pendulum environment. The latter not only has a slightly faster growth in the first stage, but also converges to a higher average return. Considering that the gap between the two trends is not negligible, PDIS-PO^{2}PE turns out to be a significant variant of the standard algorithm. Therefore, its application in other contexts represents an interesting line of research that should be further investigated for future works.

Chapter 6

Conclusions

This thesis provided a theoretical, algorithmic and experimental contribution to the Policy Optimization field. In recent years, such research area has been widely explored and mainly focused on trust-region methods, which proved to be a successful approach in addressing a variety of problems. However, the proposed work deviated from this paradigm, aiming to overcome the need for an explicitly defined trust region. To achieve this result, Importance Sampling was not merely considered a passive tool for Off-PL, but its original nature as an active technique for Off-VM was rediscovered. This shift in perspective paved the way for a novel PO algorithm, which leverages the repeated identification of the minimum-variance behavioral distribution to induce policy improvements.

In Chapter 4, we conducted a detailed study of such distribution, with particular emphasis on its theoretical properties. Starting from the well-known formulation of the minimum-variance estimate problem [Kahn, 1950; Kahn and Marshall, 1953], the *policy improvement operator* was derived. Initially, under the assumption of an unconstrained probability distribution space, we formally showed that the operator guarantees a performance improvement and that, through its repeated application, it converges to a fixed point. Moreover, we pointed out that casting the optimization problem to an iterative procedure has the advantage of enforcing an *implicit* trust region. However, these considerations could not be directly extended to the classical MDP problem, where the trajectory distributions must be subject to the transition model P. In light of this restriction, we retraced the previous steps, re-evaluating and generalizing the previous findings. Despite the performance improvement is no longer valid, it can still be preserved considering monotonic transformations of function f. Similarly, convergence to a fixed point is lost constraining the distribution space, but, even so, a

stationary point can be reached. Furthermore, in the restricted case, it is possible to control the trust region as well, assuming a particular form of convexity of distribution Q. Finally, we presented *Policy Optimization via Optimal Policy Evaluation* (PO^{2}PE), a PO algorithm grounded on these theoretical foundations. We showed that PO^{2}PE consists of an inner loop, that aims at performing the *evaluation* phase, and an outer loop, that is devoted to *optimization*.

Then, we described the experimental setting and reported the outcomes of the empirical simulations. In particular, PO^{2}PE was tested on a variety of environments and compared to other PO methods, such as TRPO and POIS, proving its remarkable performances and, more importantly, its stability across different runs. Moreover, additional tests were conducted to deepen the consequences of reducing the batch size n and modeling the transformation function h. We concluded by introducing the *Per-Decision Importance Sampling* (PDIS) estimator, a refined version of the classical one, laying the groundwork for future works and analysis.

Although we discussed its effects only on the Inverted Double Pendulum environment, potentially PDIS can lead to drastic improvements in many other scenarios. Extending its usage in combination with different configurations of the hyperparameters, e.g. different batch sizes and inner iterations, would be an interesting line of research. Another facet that can be further investigated is the *shift return* strategy. Indeed, we opted for a basic approach, but more advanced techniques may lead to consistent benefits, especially in environments that involve both positive and negative rewards.

Finally, throughout this thesis, we employed Gaussian linear policies and we proved their effectiveness in performing a wide spectrum of tasks. However, experiments conducted on more complex environments, such as Swimmer, Hopper and Half-Cheetah, revealed some limitations of this choice. In fact, in these scenarios, PO^{2}PE was characterized by a larger confidence interval, indicative of less stability across multiple runs. To overcome this problem, it might be beneficial to resort to a nonlinear representation of the policy. Therefore, in future research and experiments, the most promising and natural extension of this work is given by the application of deep neural policies.

Appendix A

Proofs and Derivations

In this appendix, we report the proofs and derivations we have omitted in the main chapters.

A.1 Proofs of Chapter 4

Proposition 4.2.1 (Proposition 9 of [Ghosh et al., 2020])**.** Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic increasing. Then, $\mathcal{I}_{h\circ f}[P]$ is a performance improvement with respect to P:

$$\mathop{\mathbb{E}}_{x\sim\mathcal{I}_{h\circ f}[P]}[f(x)] - \mathop{\mathbb{E}}_{x\sim P}[f(x)] = \frac{\mathbb{C}\mathrm{ov}_{x\sim P}[h(f(x)), f(x)]}{\mathbb{E}_{x\sim P}[h(f(x))]} \geqslant 0.$$

Proof.

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\mathcal{I}_{h\circ f}[P]}[f(x)] - \mathop{\mathbb{E}}_{x\sim P}[f(x)] &= \int_{\mathcal{X}} \frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]} f(x)\mathrm{d}x - \mathop{\mathbb{E}}_{x\sim P}[f(x)] \\
&= \frac{\mathbb{E}_{x\sim P}[h(f(x))f(x)] - \mathbb{E}_{x\sim P}[f(x)]\mathbb{E}_{x\sim P}[h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]} \\
&= \frac{\mathbb{C}\mathrm{ov}_{x\sim P}[h(f(x)), f(x)]}{\mathbb{E}_{x\sim P}[h(f(x))]}
\end{aligned}$$

where the definitions of $\mathcal{I}_{h\circ f}$ and covariance have been exploited. The result is obtained by recalling that h is increasing and the covariance between two increasing functions of the same random variable (i.e., h and the identity function) is non-negative [Cuadras, 2002]. ■

Theorem 4.2.2. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Then, the following statements hold:

(i) P is a fixed point of $\mathcal{I}_{h\circ f}$, i.e. $\mathcal{I}_{h\circ f}[P] = P$ iff $\mathbb{V}\mathrm{ar}_{x\in P}[f(x)] = 0$;

(ii) let $\mathcal{X}^* = \mathrm{argmax}_{x\in\mathrm{supp}(P)}\{f(x)\}$ be the set of maxima of f restricted to the support of P. If $\mathcal{X}^*$ is non-empty and measurable, the repeated application of $\mathcal{I}_{h\circ f}$ converges to a distribution $Q_\infty = \lim_{k\to\infty}(\mathcal{I}_{h\circ f})^k[P]$ with support $\mathcal{X}^*$. In particular:

$$\mathop{\mathbb{E}}_{x\sim Q_\infty}[f(x)] = \max_{x\in\mathrm{supp}(P)}\{f(x)\}.$$

Proof. We start with (i). First of all, it can be observed that, since h is monotonically strictly-increasing, it holds that $\mathbb{V}\mathrm{ar}_{x\sim P}[f(x)] = 0$ if and only if $\mathbb{V}\mathrm{ar}_{x\sim P}[h(f(x))] = 0$. P is a fixed point of $\mathcal{I}_{h\circ f}$, i.e., $P = \mathcal{I}_{h\circ f}[P]$ a.s. if and only if for all $x \in \mathcal{X}$ it holds a.s.:

$$p(x) = \frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]},$$

that occurs if and only if either $p(x) = 0$ ($x \notin \mathrm{supp}(P)$) or $h(f(x)) = \mathbb{E}_{x\sim P}[h(f(x))]$.

- ($\Rightarrow$) Whenever $p(x)$ is not zero, function $h(f(x))$ is a constant in $\mathrm{supp}(P)$ and, consequently, its variance under P is zero.

- ($\Leftarrow$) Suppose that $\mathbb{V}\mathrm{ar}_{x\sim P}[h(f(x))] = 0$, then $h(f(x)) = \mathbb{E}_{x\sim P}[h(f(x))]$ almost surely and, consequently $\dfrac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]} = p(x)$ almost surely.

Let us now consider (ii). First of all, we can easily observe that for every $k \in \mathbb{N}$:

$$(\mathcal{I}_{h\circ f})^k[P](x) = \frac{p(x)f(x)^k}{\mathbb{E}_{x\sim P}[f(x)^k]}.$$

Let $f^* = \max_{x\in\mathrm{supp}(P)}\{f(x)\}$, consider the function $g_k(x) = p(x)\left(\frac{f(x)}{f^*}\right)^k$ and the limit:

$$\lim_{k\to\infty} g_k(x) = \lim_{k\to\infty} p(x)\left(\frac{f(x)}{f^*}\right)^k = \begin{cases} p(x) & \text{if } x \in \mathcal{X}^* \\ 0 & \text{otherwise} \end{cases}.$$

Thus, we have:

$$Q_\infty = \lim_{k\to\infty} \left(\mathcal{I}_{h\circ f}\right)^k [P](x) = \lim_{k\to\infty} \frac{p(x)f(x)^k}{\int_{\mathcal{X}} p(x)f(x)^k \mathrm{d}x}$$

$$= \lim_{k\to\infty} \frac{g_k(x)}{\int_{\mathcal{X}} g_k(x)\mathrm{d}x} = \begin{cases} \frac{p(x)}{\int_{\mathcal{X}^*} p(x)\mathrm{d}x} & \text{if } x \in \mathcal{X}^* \\ 0 & \text{otherwise} \end{cases}.$$

Thus, the support of Q_∞ is given by $\mathcal{X}^*$. Consequently, the expectation of f under Q_∞ is given by:

$$\mathop{\mathbb{E}}_{x\sim Q_\infty}[f(x)] = \int_{\mathcal{X}} q^*_\infty(x)f(x)\mathrm{d}x = f^*.$$

■

Theorem 4.2.3. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Then, for every $a \in [0,\infty]$, it holds that:

$$D_\alpha(\mathcal{I}_{h\circ f}[P]\|P) = \frac{1}{\alpha-1}\log\frac{\mathbb{E}_{x\sim P}[h(f(x))^\alpha]}{\mathbb{E}_{x\sim P}[h(f(x))]^\alpha}.$$

This means that it is possible to naturally control the divergence between two consecutive distributions Q_k and $Q_{k+1} = \mathcal{I}_{h\circ f}[Q_k]$, resulting in the enforcement of an *implicit* trust region. A particular case is obtained for $\alpha = 1$:

$$D_{KL}(\mathcal{I}_{h\circ f}[P]\|P) = \frac{\mathbb{C}\mathrm{ov}_{x\sim P}[h(f(x)), \log h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]}.$$

Instead, for $\alpha = 2$ it holds that:

$$D_2(\mathcal{I}_{h\circ f}[P]\|P) = \log\frac{\mathbb{E}_{x\sim P}[h(f(x))^2]}{\mathbb{E}_{x\sim P}[h(f(x))]^2} \leqslant \frac{\mathbb{V}\mathrm{ar}_{x\sim P}[h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]^2}.$$

Proof. Let us consider the following derivation:

$$J := \int_{\mathcal{X}} \left(\left(I_{h\circ f}[P]\right)(x)\right)^\alpha p(x)^{1-\alpha}\,\mathrm{d}x = \int_{\mathcal{X}} \left(\frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]}\right)^\alpha p(x)^{1-\alpha}\mathrm{d}x$$

$$= \frac{\mathbb{E}_{x\sim P}[h(f(x))^\alpha]}{\mathbb{E}_{x\sim P}[h(f(x))]^\alpha}.$$

By observing that $D_\alpha\left(I_{h\circ f}[P]\|P\right) = \frac{1}{\alpha-1}\log J$, we obtain the result. For $\alpha = 1$, we provide an independent derivation:

$$D_{\mathrm{KL}}(I_{h\circ f}[P]\|P) = \int_{\mathcal{X}} \frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]} \log \frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]p(x)} \mathrm{d}x$$

$$= \frac{\mathbb{E}_{x\sim P}[h(f(x))\log h(f(x))] - \mathbb{E}_{x\sim P}[h(f(x))]\,\mathbb{E}_{x\sim P}[\log h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]}$$

$$= \frac{\mathbb{C}\mathrm{ov}_{x\sim P}[h(f(x)), \log h(f(x))]}{\mathbb{E}_{x\sim P}[h(f(x))]},$$

where we exploited the definition of covariance in the last line. ■

Proposition 4.3.1. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Then, for every $a \in (1,\infty)$, it holds that:

$$\underbrace{\mathop{\mathbb{E}}_{x\sim Q}\left[\left|\frac{p(x)}{q(x)}h(f(x)) - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]\right|^{\alpha}\right]}_{\text{absolute central } \alpha\text{-moment}} \leqslant \underbrace{\mathop{\mathbb{E}}_{x\sim Q}\left[\left(\frac{p(x)}{q(x)}h(f(x))\right)^{\alpha}\right]}_{\text{(non-central) } \alpha\text{-moment}}.$$

Moreover, the second term can also be expressed as:

$$\mathop{\mathbb{E}}_{x\sim Q}\left[\left(\frac{p(x)}{q(x)}h(f(x))\right)^{\alpha}\right] = e^{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q)} \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha}.$$

Thus, combining the two partial results, the property can be finally stated:

$$\mathop{\mathbb{E}}_{x\sim Q}\left[\left|\frac{p(x)}{q(x)}h(f(x)) - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]\right|^{\alpha}\right] \leqslant e^{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[P]\|Q)} \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha}.$$

Proof. First of all, we observe that since $\mathbb{E}_{x\sim Q}\left[\frac{p(x)}{q(x)}h(f(x))\right] = \mathbb{E}_{x\sim P}[h(f(x))]$, for $\alpha \geqslant 1$, the absolute central α-moment is smaller or equal than the (non-

central) α-moment. Thus, for $\alpha \geqslant 1$, we have:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x\sim Q}\left[\left|\frac{p(x)}{q(x)}h(f(x)) - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]\right|^{\alpha}\right] \leqslant \mathop{\mathbb{E}}_{x\sim Q}\left[\left(\frac{p(x)}{q(x)}h(f(x))\right)^{\alpha}\right] \\
&= \int_{\mathcal{X}} \left(\frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]}\right)^{\alpha} q(x)^{1-\alpha}\mathrm{d}x \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha} \\
&= \int_{\mathcal{X}} ((\mathcal{I}_{h\circ f}[P])(x))^{\alpha}\, q(x)^{1-\alpha}\mathrm{d}x \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha} \\
&= \exp\left\{(\alpha-1)\frac{1}{\alpha-1}\log\int_{\mathcal{X}} ((\mathcal{I}_{h\circ f}[P])(x))^{\alpha}\, q(x)^{1-\alpha}\mathrm{d}x\right\} \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^{\alpha}.
\end{aligned}$$

By applying the definition of Rényi divergences, we get the result. ■

Theorem 4.3.2. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$, $Q \in \mathcal{Q}$ and $\alpha \in [0,\infty]$, it holds that:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{x\sim Q}[h(f(x))^{\alpha}] - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))^{\alpha}] \geqslant \\
&\qquad \frac{\mathbb{E}_{x\sim P}[h(f(x))]^{\alpha}}{\alpha-1}\left(e^{(\alpha-1)D_{\alpha}(\mathcal{I}_{h\circ f}[P]\|P)} - e^{(\alpha-1)D_{\alpha}(\mathcal{I}_{h\circ f}[P]\|Q)}\right).
\end{aligned}$$

Proof. Let us consider the following derivation:

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim Q}[h(f(x))^{\alpha}] &= \int_{\mathcal{X}} q(x)h(f(x))^{\alpha}\mathrm{d}x \\
&= \int_{\mathcal{X}} p(x)\frac{q(x)}{p(x)}h(f(x))^{\alpha}\mathrm{d}x \\
&= \int_{\mathcal{X}} p(x)h(f(x))^{\alpha}\mathrm{d}x + \int_{\mathcal{X}} p(x)\left(\frac{q(x)}{p(x)} - 1\right)h(f(x))^{\alpha}\mathrm{d}x \\
&\geqslant \int_{\mathcal{X}} p(x)h(f(x))^{\alpha}\mathrm{d}x + \frac{1}{\alpha-1}\int_{\mathcal{X}} p(x)\left(1 - \left(\frac{p(x)}{q(x)}\right)^{\alpha-1}\right)h(f(x))^{\alpha}\mathrm{d}x
\end{aligned} \tag{A.1}$$

$$
\begin{aligned}
&= \mathop{\mathbb{E}}_{x\sim P}[h(f(x))^\alpha] + \frac{1}{\alpha-1}\int_{\mathcal{X}} p(x)h(f(x))^\alpha \mathrm{d}x \\
&\qquad\qquad - \frac{1}{\alpha-1}\int_{\mathcal{X}} p(x)\left(\frac{p(x)}{q(x)}\right)^{\alpha-1} h(f(x))^\alpha \mathrm{d}x \\
&= \mathop{\mathbb{E}}_{x\sim P}[h(f(x))^\alpha] + \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^\alpha \frac{1}{\alpha-1}\int_{\mathcal{X}} A \ \mathrm{d}x \\
&\qquad\qquad - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^\alpha \frac{1}{\alpha-1}\int_{\mathcal{X}} B \ \mathrm{d}x \\
&= \mathop{\mathbb{E}}_{x\sim P}[h(f(x))^\alpha] + \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^\alpha \frac{1}{\alpha-1}\exp\left\{(\alpha-1)\frac{1}{\alpha-1}\log\int_{\mathcal{X}} A \ \mathrm{d}x\right\} \\
&\qquad\qquad - \mathop{\mathbb{E}}_{x\sim P}[h(f(x))]^\alpha \frac{1}{\alpha-1}\exp\left\{(\alpha-1)\frac{1}{\alpha-1}\log\int_{\mathcal{X}} B \ \mathrm{d}x\right\} \\
&= \mathop{\mathbb{E}}_{x\sim P}[h(f(x))^\alpha] + \frac{\mathbb{E}_{x\sim P}[h(f(x))]^\alpha}{\alpha-1}\exp\left\{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}\|P)\right\} \\
&\qquad\qquad - \frac{\mathbb{E}_{x\sim P}[h(f(x))]^\alpha}{\alpha-1}\exp\left\{(\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}\|Q)\right\},
\end{aligned}
$$

where the terms A and B are defined as follows:

$$
A = \left(\frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]}\right)^\alpha p(x)^{1-\alpha},
$$

$$
B = \left(\frac{p(x)h(f(x))}{\mathbb{E}_{x\sim P}[h(f(x))]}\right)^\alpha q(x)^{1-\alpha}.
$$

where line (A.1) derived from Lemma A.2.1. The second inequality was provided in Proposition 6 of [Ghosh et al., 2020]. ■

Theorem 4.3.3. Let $P \in \mathscr{P}(\mathcal{X})$, $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ and suppose that $h \circ f$ is bounded from above. The iterate $Q_{k+1} \in \arg\min_{Q\in\mathcal{Q}}\{D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q)\}$, where possible ties are broken arbitrarly, satisfies:

(i) the sequence of divergences $D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_k)$ is convergent;

(ii) the sequence of expectations $\mathbb{E}_{x\sim Q_k}[h(f(x))^\alpha]$ is non-decreasing in $k \in$

$\mathbb{N}$ and converges to a stationary point of $\mathbb{E}_{x\sim Q}[h(f(x))^\alpha]$ with respect to $Q \in \mathcal{Q}$.

Proof. Let us consider the sequence of distributions $(Q_k)_{k\in\mathbb{N}}$, generated by the iterate in Equation (4.3), where possible ties are broken with an arbitrary (possibly with a tie-breaking rule T_k different for every k). From Theorem 4.3.2, we have for every $k \in \mathbb{N}$:

$$\mathop{\mathbb{E}}_{x\sim Q_{k+1}}[h(f(x))^\alpha] - \mathop{\mathbb{E}}_{x\sim Q_k}[h(f(x))^\alpha] \geqslant \frac{\mathbb{E}_{x\sim Q_k}[h(f(x))]^\alpha}{\alpha-1}\left(e^A - e^B\right) \geqslant 0,$$

where terms A and B are defined as:

$$A = (\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_k)$$

$$B = (\alpha-1)D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_{k+1})$$

and we simply exploited that $Q_k \in \arg\min_{Q\in\mathcal{Q}}\{D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q)\}$. Thus, $\mathbb{E}_{x\sim Q_k}[h(f(x))^\alpha]$ is a non-decreasing function of k. Since $h\circ f$ is bounded, it must be that $\lim_{k\to\infty}\mathbb{E}_{x\sim Q_k}[h(f(x))^\alpha] = \mu_\infty < \infty$, that proves convergence. Furthermore, being convergent, for $k \to \infty$ it must be that:

$$\mathop{\mathbb{E}}_{x\sim Q_k}[h(f(x))^\alpha] = \mathop{\mathbb{E}}_{x\sim Q_{k+1}}[h(f(x))^\alpha],$$

and consequently:

$$D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_k) = D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q_{k+1}).$$

Therefore, even if the tie-braking rule prescribes to select $Q_{k+1} \neq Q_k$ we could select Q_k instead, since it lead to the same divergence value. Consequently, being Q_k a solution, we can assert that it is a stationary point of the function $D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|\cdot)$ (as well as Q_{k+1}):

$$\begin{aligned}
0 &= \nabla_{q(\cdot)} D_\alpha(\mathcal{I}_{h\circ f}[Q_k]\|Q)|_{Q=Q_k} \\
&= \frac{1}{(\alpha-1)C}\nabla_{q(\cdot)}\int_{\mathcal{X}} h(f(x))^\alpha q_k(x)^\alpha q(x)^{1-\alpha}\mathrm{d}x|_{Q=Q_k} \\
&= -\frac{1}{C}\int_{\mathcal{X}} h(f(x))^\alpha q_k(x)^\alpha q(x)^{-\alpha}\mathrm{d}x|_{Q=Q_k}
\end{aligned}$$

$$= -\frac{1}{C}\int_{\mathcal{X}} h(f(x))^{\alpha}\mathrm{d}x.$$

where term C is defined as:

$$C = e^{(\alpha-1)D_{\alpha}(\mathcal{I}_{h\circ f}[Q_k]\|Q)} \mathop{\mathbb{E}}_{x\sim Q_k}[h(f(x))].$$

We observe that the latter expression is zero if and only if the gradient of $\mathbb{E}_{x\sim Q}[h(f(x))^{\alpha}]$ w.r.t. Q is zero. Indeed:

$$\nabla_{q(\cdot)} \mathop{\mathbb{E}}_{x\sim Q}[h(f(x))^{\alpha}] = \int_{\mathcal{X}} h(f(x))^{\alpha}\mathrm{d}x.$$

Thus, the process converges to a stationary point of $\mathbb{E}_{x\sim Q_k}[h(f(x))^{\alpha}]$. ■

Theorem 4.3.4. Let $f \in \mathscr{B}(\mathcal{X}, [0,\infty))$ and $h : [0,\infty) \to [0,\infty)$ monotonic strictly increasing. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ be a $(1-\alpha)$-convex set (Definition 4 of [van Erven and Harremoës, 2014]), $P \in \mathcal{Q}$, $Q^{\dagger} \in \arg\min_{Q\in\mathcal{Q}}\{D_{\alpha}(\mathcal{I}_{h\circ f}[P]\|Q)\}$ and $\alpha \in [0,\infty]$, then it holds that:

$$D_{\alpha}(Q^{\dagger}\|P) \leqslant D_{\alpha}(\mathcal{I}_{h\circ f}[P]\|P) - D_{\alpha}(\mathcal{I}_{h\circ f}[P]\|Q^{\dagger}).$$

Proof. The proof is a simple application of Lemma A.2.2, by taking $Q \leftarrow P$, $Q^{*} \leftarrow Q^{\dagger}$, and $P \leftarrow \mathcal{I}_{h\circ f}[P]$. ■

Theorem 4.4.1. Let $\mathcal{Q}_{\Theta} \subseteq \mathscr{P}(\mathcal{X})$ be a set of parametric distributions and let $\boldsymbol{\theta}$,$\boldsymbol{\theta}_i \in \Theta$. If $\|h\circ f\|_{\infty} \leqslant \overline{m}$, then, if all samples are independent, for every $\delta \in [0,1]$, with probability at least $1-\delta$ it holds that:

$$\mathop{\mathbb{E}}_{x\sim\boldsymbol{\theta}}\left[\left(\frac{q_{\boldsymbol{\theta}_i}(x)}{q_{\boldsymbol{\theta}}(x)}h(f(x))\right)^{\alpha}\right] \leqslant \widehat{d}_{\alpha}\left(\mathcal{I}_{h\circ f}[Q_{\boldsymbol{\theta}_i}]\|Q_{\boldsymbol{\theta}};\Phi_{i,j}\right) + \overline{m}^{\alpha}\sqrt{\frac{2\log\frac{1}{\delta}}{nj}\int_{\mathcal{X}}\frac{q_{\boldsymbol{\theta}_i}(x)^{2\alpha}}{\Phi_{i,j}(x)q_{\boldsymbol{\theta}}(x)^{2(\alpha-1)}}\mathrm{d}x},$$

Proof. We start observing that each addendum of $\widehat{d}_{\alpha}\left(\mathcal{I}_{h\circ f}[Q_{\boldsymbol{\theta}_i}]\|Q_{\boldsymbol{\theta}};\Phi_{i,j}\right)$ is non negative. Since all terms are i.i.d., we can apply unilateral Bernstein's inequality [Maurer et al., 2003] that allows achieving an exponential concentration. Thus, for every $\delta \in [0,1]$, with probability at least $1-\delta$ it holds that:

$$\mathop{\mathbb{E}}_{x\sim\boldsymbol{\theta}}\left[\left(\frac{q_{\boldsymbol{\theta}_i}(x)}{q_{\boldsymbol{\theta}}(x)}h(f(x))\right)^{\alpha}\right] \leqslant \widehat{d}_{\alpha}\left(\mathcal{I}_{h\circ f}[Q_{\boldsymbol{\theta}_i}]\|Q_{\boldsymbol{\theta}};\Phi_{i,j}\right) + \sqrt{2\mathop{\mathbb{V}\mathrm{ar}}_{x_i\sim\Phi_{i,j}}\left[\widehat{d}_{\alpha}\left(\mathcal{I}_{h\circ f}[Q_{\boldsymbol{\theta}_i}]\|Q_{\boldsymbol{\theta}};\Phi_{i,j}\right)\right]\log\frac{1}{\delta}}.$$

Thus, it remains to provide a bound on the variance term. We exploit the fact that $h(f(x)) \leqslant \overline{m}$ and that each addendum represents an i.i.d. random variable:

$$\begin{aligned}
&\underset{x_i \sim \Phi_{i,j}}{\mathbb{V}\mathrm{ar}} \left[\widehat{d}_\alpha \left(\mathcal{I}_{h \circ f}[Q_{\boldsymbol{\theta}_i}] \| Q_{\boldsymbol{\theta}}; \Phi_{i,j} \right) \right] \\
&\leqslant \frac{1}{(nj)^2} \sum_{k \in [j]} \sum_{l \in [n]} \underset{x_{k,l} \sim \Phi_{i,j}}{\mathbb{E}} \left[\left(\frac{q_{\boldsymbol{\theta}_i}(x_{k,l})^\alpha}{\Phi_{i,j}(x_{k,l}) q_{\boldsymbol{\theta}}(x_{k,l})^{\alpha-1}} h(f(x))^\alpha \right)^2 \right] \\
&\leqslant \frac{\overline{m}^{2\alpha}}{(nj)^2} \sum_{k \in [j]} \sum_{l \in [n]} \underset{x_{k,l} \sim \Phi_{i,j}}{\mathbb{E}} \left[\left(\frac{q_{\boldsymbol{\theta}_i}(x_{k,l})^\alpha}{\Phi_{i,j}(x_{k,l}) q_{\boldsymbol{\theta}}(x_{k,l})^{\alpha-1}} \right)^2 \right] \\
&= \frac{\overline{m}^{2\alpha}}{nj} \underset{x \sim \Phi_{i,j}}{\mathbb{E}} \left[\left(\frac{q_{\boldsymbol{\theta}_i}(x)^\alpha}{\Phi_{i,j}(x) q_{\boldsymbol{\theta}}(x)^{\alpha-1}} \right)^2 \right].
\end{aligned}$$

■

A.2 Technical Lemmas

Lemma A.2.1. For every $x \geqslant 0$ and $\alpha \in (0,1) \cup (1, \infty)$, it holds that:

$$x - 1 \geqslant \frac{1}{\alpha - 1} \left(1 - \frac{1}{x^{\alpha-1}} \right).$$

Furthermore, for $\alpha = 1$, it holds that:

$$x - 1 \geqslant \log x.$$

Proof. Consider the auxiliary function $g_\alpha(x) = x - 1 - \frac{1}{\alpha-1}\left(1 - \frac{1}{x^{\alpha-1}}\right)$. We are going to prove that the minimum of $g_\alpha(x)$ is zero. Suppose $\alpha > 1$, then $g_\alpha(0) = \infty$ and $g_a(\infty) = \infty$. Thus, the minimum must lie in between and since function g_α is differentiable, we have:

$$\frac{\partial}{\partial x} g_\alpha(x) = 1 - x^{-\alpha} = 0 \quad \Longrightarrow \quad x = 1.$$

Thus, we have $g_\alpha(1) = 0$. Suppose now that $\alpha < 1$, we have $g_\alpha(0) = \frac{\alpha}{1-\alpha} > 0$ and $g_\alpha(\infty) = \infty$. Thus, again, the minimum must lie in between and with the same calculations as before, we conclude $g_\alpha(1) = 0$. The case $\alpha = 1$ is trivial. ■

Lemma A.2.2. Let $P \in \mathscr{P}(\mathcal{X})$ and let $\alpha \in (0, \infty)$. Let $\mathcal{Q} \subseteq \mathscr{P}(\mathcal{X})$ be an $(\alpha - 1)$-convex [van Erven and Harremoës, 2014, Definition 4] subset of distributions. Let $Q^* \in \mathcal{Q}$ be the α-moment projection:

$$Q^* = \arg\min_{Q \in \mathcal{Q}} \{D_\alpha(P\|Q)\}.$$

If Q^* exists, then for every $Q \in \mathcal{Q}$ if holds that:

$$D_\alpha(P\|Q) \geqslant D_\alpha(P\|Q^*) + D_\alpha(Q^*\|Q).$$

Proof. The proof of the result is inspired to [van Erven and Harremoës, 2014, Theorem 14]. Let $\lambda \in [0, 1]$ and let us define Q_λ as the $(1 - \alpha, (1 - \lambda, \lambda))$-mixture of Q^* and Q:

$$q_\lambda(x) = Z_\lambda^{-1} \left((1-\lambda)q^*(x)^{1-\alpha} + \lambda q(x)^{1-\alpha}\right)^{\frac{1}{1-\alpha}},$$

$$Z_\lambda = \int_{\mathcal{X}} \left((1-\lambda)q^*(x)^{1-\alpha} + \lambda q(x)^{1-\alpha}\right)^{\frac{1}{1-\alpha}} \mathrm{d}x.$$

Let us first observe that for $\lambda = 0$, we have $Q_0 = Q^*$ and $Z_0 = \int_{\mathcal{X}} q^*(x)\mathrm{d}x = 1$. Since $\mathcal{Q}$ is $(1 - \alpha)$-convex and Q^* is the minimizer over $\mathcal{Q}$, it holds that $\frac{\partial}{\partial\lambda} D_\alpha(P\|Q_\lambda)|_{\lambda=0} \geqslant 0$. First of all, we compute:

$$\int_{\mathcal{X}} p(x)^\alpha q_\lambda(x)^{1-\alpha}\mathrm{d}x = Z_\lambda^{\alpha-1} \int_{\mathcal{X}} \left[(1-\lambda)p(x)^\alpha q^*(x)^{1-\alpha} + \lambda p(x)^\alpha q(x)^{1-\alpha}\right] \mathrm{d}x$$

$$\frac{\partial}{\partial\lambda} Z_\lambda = \frac{1}{1-\alpha} \int_{\mathcal{X}} \left((1-\lambda)q^*(x)^{1-\alpha} + \lambda q(x)^{1-\alpha}\right)^{\frac{\alpha}{1-\alpha}} \left(q(x)^{1-\alpha} - q^*(x)^{1-\alpha}\right) \mathrm{d}x.$$

The latter, for $\lambda = 0$, becomes: $\frac{\partial}{\partial\lambda} Z_\lambda\Big|_{\lambda=0} = \frac{1}{1-\alpha}\left[\int_{\mathcal{X}} q^*(x)^\alpha q(x)^{1-\alpha} - 1\right]$. For calculation easiness, instead of directly operating on $D_\alpha(P\|Q_\lambda)$, we consider:

$$\frac{\partial}{\partial\lambda} \int_{\mathcal{X}} p(x)^\alpha q_\lambda(x)^{1-\alpha}\mathrm{d}x$$

$$= Z_\lambda^{\alpha-1} \int_{\mathcal{X}} \left[-p(x)^\alpha q^*(x)^{1-\alpha} + p(x)^\alpha q(x)^{1-\alpha}\right] \mathrm{d}x,$$

$$+ (\alpha-1) Z_\lambda^{\alpha-2} \frac{\partial}{\partial\lambda} Z_\lambda \int_{\mathcal{X}} \left[(1-\lambda)p(x)^\alpha q^*(x)^{1-\alpha} + \lambda p(x)^\alpha q(x)^{1-\alpha}\right] \mathrm{d}x.$$

We now evaluate it at $\lambda = 0$:

$$\begin{aligned}
&\frac{\partial}{\partial\lambda}\int_{\mathcal{X}} p(x)^{\alpha} q_{\lambda}(x)^{1-\alpha}\mathrm{d}x\Big|_{\lambda=0} \\
&\qquad = -\int_{\mathcal{X}} p(x)^{\alpha} q^{*}(x)^{1-\alpha}\mathrm{d}x + \int_{\mathcal{X}} p(x)^{\alpha} q(x)^{1-\alpha}\mathrm{d}x \\
&\qquad\quad - \int_{\mathcal{X}} p(x)^{\alpha} q^{*}(x)^{1-\alpha}\mathrm{d}x \left[\int_{\mathcal{X}} q^{*}(x)^{\alpha} q(x)^{1-\alpha}\mathrm{d}x - 1\right].
\end{aligned}$$

For $\alpha \geqslant 1$, we require $\frac{\partial}{\partial\lambda}\int_{\mathcal{X}} p(x)^{\alpha} q_{\lambda}(x)^{1-\alpha}\mathrm{d}x\Big|_{\lambda=0} \geqslant 0$, to obtain:

$$\int_{\mathcal{X}} p(x)^{\alpha} q(x)^{1-\alpha}\mathrm{d}x \geqslant \int_{\mathcal{X}} p(x)^{\alpha} q^{*}(x)^{1-\alpha}\mathrm{d}x \int_{\mathcal{X}} q^{*}(x)^{\alpha} q(x)^{1-\alpha}\mathrm{d}x.$$

By applying both sides the log function and dividing by $\frac{1}{\alpha-1} > 0$ we get the result. Symmetrically, for $\alpha < 1$, we require the converse

$$\frac{\partial}{\partial\lambda}\int_{\mathcal{X}} p(x)^{\alpha} q_{\lambda}(x)^{1-\alpha}\mathrm{d}x\Big|_{\lambda=0} \leqslant 0$$

.

Recalling that $\frac{1}{\alpha-1} < 0$, we obtain the desired result. ■

A.3 Optimizing Moments of f

Lemma A.3.1. Let $P \in \mathscr{P}(\mathcal{X})$ and $f \in \mathscr{B}(\mathcal{X}, [\underline{m}, \overline{m}]$. If $\alpha \in (1, \infty)$, it holds that:

$$\begin{aligned}
0 &\leqslant \mathop{\mathbb{E}}_{x\sim P}[f(x)^{\alpha}] - \left(\mathop{\mathbb{E}}_{x\sim P}[f(x)]\right)^{\alpha} \\
&\leqslant \frac{\underline{m}^{\alpha}(\overline{m} - \mathbb{E}_{x\sim P}[f(x)]) + \overline{m}^{\alpha}(\mathbb{E}_{x\sim P}[f(x)] - \underline{m}) - \mathbb{E}_{x\sim P}[f(x)]^{\alpha}(\overline{m} - \underline{m})}{\overline{m} - \underline{m}}.
\end{aligned}$$

In particular for $\alpha = 2$, we have:

$$0 \leqslant \mathop{\mathbb{E}}_{x\sim P}\left[f(x)^2\right] - \left(\mathop{\mathbb{E}}_{x\sim P}[f(x)]\right)^2 \leqslant \left(\overline{m} - \mathop{\mathbb{E}}_{x\sim P}[f(x)]\right)\left(\mathop{\mathbb{E}}_{x\sim P}[f(x)] - \underline{m}\right),$$

that is the Bhatia-Davis inequality for the variance.

Proof. We explicitly consider the optimization problem, for $\alpha \geqslant 1$ and hav-

ing denoted $\mu = \mathbb{E}_{x\sim P}[f(x)]$:

$$\max_{f:\mathcal{X}\to\mathbb{R}} \int_{\mathcal{X}} p(x) f(x)^{\alpha} \mathrm{d}x$$

$$\text{s.t.} \int_{\mathcal{X}} p(x) f(x) = \mu$$

$$\underline{m} \leqslant f(x) \leqslant \overline{m}.$$

Since $\alpha \geqslant 1$, the optimization problem corresponds to the maximization of a concave function subject to linear and box constraints. It is simple to prove that the optimal solution must assign extreme values to function f. Let $p \in [0, 1]$, the linear and box constraints enforce:

$$p\underline{m} + (1-p)\overline{m} = \mu \quad \Longrightarrow \quad p = \frac{\overline{m} - \mu}{\overline{m} - \underline{m}}.$$

From which, by substitution in the objective function, we have:

$$\int_{\mathcal{X}} p(x) f(x)^{\alpha} \mathrm{d}x = p\underline{m}^{\alpha} + (1-p)\overline{m}^{\alpha} = \frac{\underline{m}^{\alpha}(\overline{m} - \mu) + \overline{m}^{\alpha}(\mu - \underline{m})}{\overline{m} - \underline{m}}.$$

■

Thus, in general, optimizing moments of the function f, leads to different optimal policies compared to optimizing function f directly. However, from the above results, we see that this discrepancy reduces when the expectation $\mathbb{E}_{x\sim P}[f(x)]$ approaches the extreme value $\overline{m}$ (and also $\underline{m}$, but this is less interesting since we are maximizing). The value $\overline{m}$ can be indeed achieved if we have no restrictions on the distribution space.

A.4 Closed Form of the Integral for Gaussians

Finally, we derive a closed form for the integral involved in the computation of the bound of Theorem 4.4.1 in the case that all involved distributions are Gaussians and for $\alpha = 2$. Let us introduce the notation:

$$\mu = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_{\boldsymbol{\mu}}), \qquad \phi = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}, \boldsymbol{\Sigma}_{\boldsymbol{\phi}}), \qquad \nu = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\nu}}, \boldsymbol{\Sigma}_{\boldsymbol{\nu}}).$$

We have to compute the following integral:

$$\int_{\mathcal{X}} \frac{\mu^4(\mathbf{x})}{\phi(\mathbf{x})\nu(\mathbf{x})^2} \mathrm{d}\mathbf{x}.$$

Let us start elaborating on the integrand function, denoting for properly sized vector $\mathbf{x}$ and matrix $\mathbf{S}$, $\|\mathbf{m}\|_{\mathbf{S}} = \mathbf{x}^T\mathbf{S}\mathbf{x}$ and $|\mathbf{S}|$ the determinant of $\mathbf{S}$:

$$\frac{\mu^4(\mathbf{x})}{\phi(\mathbf{x})\nu(\mathbf{x})^2}$$

$$= \frac{\alpha|\boldsymbol{\Sigma_\mu}|^{-2}\exp\left(-2\|\mathbf{x}-\boldsymbol{\mu_\mu}\|^2_{\boldsymbol{\Sigma_\mu}^{-1}}\right)}{\alpha|\boldsymbol{\Sigma_\phi}|^{-1/2}\exp\left(-1/2\|\mathbf{x}-\boldsymbol{\mu_\phi}\|^2_{\boldsymbol{\Sigma_\phi}^{-1}}\right)\alpha|\boldsymbol{\Sigma_\nu}|^{-1}\exp\left(-\|\mathbf{x}-\boldsymbol{\mu_\nu}\|^2_{\boldsymbol{\Sigma_\nu}^{-1}}\right)}$$

$$= C\cdot\exp\left(-2\|\mathbf{x}-\boldsymbol{\mu_\mu}\|^2_{\boldsymbol{\Sigma_\mu}^{-1}} + 1/2\|\mathbf{x}-\boldsymbol{\mu_\phi}\|^2_{\boldsymbol{\Sigma_\phi}^{-1}} + \|\mathbf{x}-\boldsymbol{\mu_\nu}\|^2_{\boldsymbol{\Sigma_\nu}^{-1}}\right),$$

where the term C is defined as:

$$C = \frac{(2\pi)^{-k/2}|\boldsymbol{\Sigma_\mu}|^{-2}}{|\boldsymbol{\Sigma_\phi}|^{-1/2}|\boldsymbol{\Sigma_\nu}|^{-1}}$$

and $\alpha = (2\pi)^{-k/2}$. Now, dealing with the argument of the exponential:

$$\begin{aligned}
&-2\|\mathbf{x}-\boldsymbol{\mu_\mu}\|^2_{\boldsymbol{\Sigma_\mu}^{-1}} + 1/2\|\mathbf{x}-\boldsymbol{\mu_\phi}\|^2_{\boldsymbol{\Sigma_\phi}^{-1}} + \|\mathbf{x}-\boldsymbol{\mu_\nu}\|^2_{\boldsymbol{\Sigma_\nu}^{-1}} \\
&\quad= -\frac{1}{2}\mathbf{x}^T\underbrace{\left(4\boldsymbol{\Sigma_\mu}^{-1} - \boldsymbol{\Sigma_\phi}^{-1} - 2\boldsymbol{\Sigma_\nu}^{-1}\right)}_{M}\mathbf{x} \\
&\qquad+ \underbrace{\left(4\boldsymbol{\Sigma_\mu}^{-1}\boldsymbol{\mu_\mu} - \boldsymbol{\Sigma_\phi}^{-1}\boldsymbol{\mu_\phi} - 2\boldsymbol{\Sigma_\nu}^{-1}\boldsymbol{\mu_\nu}\right)^T}_{b^T}\mathbf{x} \\
&\qquad- \frac{1}{2}\underbrace{\left(4\boldsymbol{\mu_\mu}^T\boldsymbol{\Sigma_\mu}^{-1}\boldsymbol{\mu_\mu} - \boldsymbol{\mu_\phi}^T\boldsymbol{\Sigma_\phi}^{-1}\boldsymbol{\mu_\phi} - 2\boldsymbol{\mu_\nu}^T\boldsymbol{\Sigma_\nu}^{-1}\boldsymbol{\mu_\nu}\right)}_{c}.
\end{aligned}$$

We now proceed completing the square:

$$\mathbf{x}^T\mathbf{M}\mathbf{x} - 2\mathbf{b}^T\mathbf{x} = (\mathbf{x}-\mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\mathbf{x}-\mathbf{M}^{-1}\mathbf{b}) - \mathbf{b}^T\mathbf{M}^{-1}\mathbf{b}.$$

Thus, we have:

$$-\frac{1}{2}(\mathbf{x}^T\mathbf{M}\mathbf{x} - 2\mathbf{b}^T\mathbf{x} + \mathbf{c}) \\ = -\frac{1}{2}(\mathbf{x} - \mathbf{M}^{-1}\mathbf{b})^T\mathbf{M}(\mathbf{x} - \mathbf{M}^{-1}\mathbf{b}) + \frac{1}{2}\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b} - \frac{1}{2}\mathbf{c}.$$

Moreover, we observe that the following expression is the density of a k-variate normal distribution with mean $M^{-1}b$ and covariance matrix M^{-1}:

$$(2\pi)^{-k/2}|\mathbf{M}^{-1}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{M}^{-1}\mathbf{x})^T\mathbf{M}(\mathbf{x} - \mathbf{M}^{-1}\mathbf{b})\right)$$

Thus, its integral is 1. Therefore, coming to the initial expression:

$$\int_{\mathcal{X}} \frac{\mu^4(\mathbf{x})}{\phi(\mathbf{x})\nu(\mathbf{x})^2}\mathrm{d}\mathbf{x}$$

$$= \frac{(2\pi)^{-k/2}|\boldsymbol{\Sigma_\mu}|^{-2}}{|\boldsymbol{\Sigma_\phi}|^{-1/2}|\boldsymbol{\Sigma_\nu}|^{-1}}\left((2\pi)^{-k/2}|\mathbf{M}^{-1}|^{-1/2}\right)^{-1}\exp\left(\frac{1}{2}\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b} - \frac{1}{2}\mathbf{c}\right)$$

$$= \frac{|\boldsymbol{\Sigma_\phi}|^{1/2}|\boldsymbol{\Sigma_\nu}|}{|\boldsymbol{\Sigma_\mu}|^2|\mathbf{M}|^{1/2}}\exp\left(\frac{1}{2}\left(\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b} - \mathbf{c}\right)\right).$$

Appendix B

Experimental Details

In this appendix, we report the low-level experimental details and the configuration of hyperparameters which leads to the results in Chapter 5.

Infrastructure The experiments have been run on two machines:

- 2 x CPUs Intel(R) Xeon(R) CPU E7-8880 v4 @ 2.20GHz (22 cores, 44 thread, 55 MB cache) and 128 GB RAM;
- 4 x Intel(R) Xeon(R) CPU E5-4610 v2 @ 2.30GHz (8 cores, 16 thread, 16 MB cache) and 256 GB RAM.

Environments The environments are the rllab implementations [Duan et al., 2016], MIT license, `https://github.com/rll/rllab`. The Swimmer, Hopper and Half-Cheetah environments belong to the Mujoco suite [Todorov et al., 2012], MuJoCo Personal License, `http://www.mujoco.org/`.

Algorithms The TRPO implementation is taken from baselines [Dhariwal et al., 2017], MIT licence, `https://github.com/openai/baselines`. For POIS we use the original implementation [Metelli et al., 2018], MIT license, `https://github.com/T3p/baselines`.

Hyperparameters A set of 5 seeds, in bold, was used to test the performances during the tuning phase. Once the optimal hyperparameters were found, the experiments were extended to other 15 seeds. In the following, we report the values of the hyperparameters for PO^{2}PE, POIS and TRPO. The *shift return* refers to the need for making the return non-negative. The *variance initialization* hyperparameter refers to the logarithm of the standard deviation.

Table B.1: Hyperparameters used for Cartpole simulations.

	PO^{2}PE	POIS	TRPO
Max iterations	500	500	500
Number of episodes	100	100	100
Policy	linear	linear	linear
Policy initialization	zeros	zeros	zeros
Variance initialization	-1	-1	-1
Step size	constant	line search	D_{KL}
Delta	0.75	0.4	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	False	-	-
Max KL	-	-	0.01

Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87, 90
123, 222, 343, 404, 452, 542, 875, 943, 999

Table B.2: Hyperparameters used for Inverted Double Pendulum simulations.

	PO^{2}PE	POIS	TRPO
Max iterations	500	500	500
Number of episodes	100	100	100
Policy	linear	linear	linear
Policy initialization	zeros	zeros	zeros
Variance initialization	-1	-1	-1
Step size	constant	line search	D_{KL}
Delta	0.99	0.1	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	False	-	-
Max KL	-	-	0.001

Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87, 90
123, 222, 343, 404, 452, 542, 875, 943, 999

Table B.3: Hyperparameters used for Mountain Car simulations.

	PO^{2}PE	POIS	TRPO
Max iterations	500	500	500
Number of episodes	100	100	100
Policy	linear	linear	linear
Policy initialization	zeros	zeros	zeros
Variance initialization	-1	-1	-1
Step size	constant	line search	D_{KL}
Delta	0.9	0.9	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	True	-	-
Max KL	-	-	0.01

Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87, 90
123, 222, 343, 404, 452, 542, 875, 943, 999

Table B.4: Hyperparameters used for Swimmer simulations.

	PO^{2}PE	POIS	TRPO
Max iterations	500	500	500
Number of episodes	100	100	100
Policy	linear	linear	linear
Policy initialization	zeros	zeros	zeros
Variance initialization	-1	-1	-1
Step size	constant	line search	D_{KL}
Delta	0.99	0.8	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	True	-	-
Max KL	-	-	0.01

Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87, 90
123, 222, 343, 404, 452, 542, 875, 943, 999

Table B.5: Hyperparameters used for Hopper simulations.

	PO^{2}PE	POIS	TRPO
Max iterations	500	500	-
Number of episodes	100	100	-
Policy	linear	linear	-
Policy initialization	zeros	zeros	-
Variance initialization	-0.5	-0.5	-
Step size	constant	line search	-
Delta	0.99	0.99	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	False	-	-
Max KL	-	-	-
Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87			

Table B.6: Hyperparameters used for Half-Cheetah simulations.

	PO^{2}PE	POIS	TRPO
Max iterations	500	500	-
Number of episodes	100	100	-
Policy	linear	linear	-
Policy initialization	zeros	zeros	-
Variance initialization	-0.03	-0.03	-
Step size	constant	line search	-
Delta	0.7	0.99	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	True	-	-
Max KL	-	-	-
Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87			

Table B.7: Hyperparameters used for Cartpole simulations with $n = 50$.

	PO^{2}PE	POIS	TRPO
Max iterations	1000	1000	1000
Number of episodes	50	50	50
Policy	linear	linear	linear
Policy initialization	zeros	zeros	zeros
Variance initialization	-1	-1	-1
Step size	constant	line search	D_{KL}
Delta	0.75	0.4	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	False	-	-
Max KL	-	-	0.001
Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87			

Table B.8: Hyperparameters used for Cartpole simulations with $n = 11$.

	PO^{2}PE	POIS	TRPO
Max iterations	5000	5000	5000
Number of episodes	11	11	11
Policy	linear	linear	linear
Policy initialization	zeros	zeros	zeros
Variance initialization	-1	-1	-1
Step size	constant	line search	D_{KL}
Delta	0.75	0.4	-
Max offline iterations	10	10	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	False	-	-
Max KL	-	-	0.001
Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87			

Table B.9: Hyperparameters used for Inverted Double Pendulum simulations with power functions.

	PO^{2}PE	POIS	TRPO
Max iterations	500	-	-
Number of episodes	100	-	-
Policy	linear	-	-
Policy initialization	zeros	-	-
Variance initialization	-1	-	-
Step size	constant	-	-
Delta	0.99	-	-
Max offline iterations	10	-	-
Capacity	1	-	-
Inner iterations	1	-	-
Penalization	True	-	-
Shift return	False	-	-
Max KL	-	-	-
Seeds: **0**, **19**, **42**, **66**, **72**			
Power: 0.1, 0.5, 1, 2, 4			

Table B.10: Hyperparameters used for Inverted Double Pendulum simulations with PDIS estimator.

	PO^{2}PE-PDIS	PO^{2}PE	POIS	TRPO
Max iterations	500	500	-	-
Number of episodes	100	100	-	-
Policy	linear	linear	-	-
Policy initialization	zeros	zeros	-	-
Variance initialization	-1	-1	-	-
Step size	constant	constant	-	-
Delta	0.99	0.99	-	-
Max offline iterations	10	10	-	-
Capacity	1	-	-	-
Inner iterations	1	-	-	-
Penalization	True	-	-	-
Shift return	False	-	-	-
Bound	pdis-max	max-d2	-	-
IS method	pdis	is	-	-
Seeds: **0**, 3, 11, 16, **19**, **42**, **66**, **72**, 84, 87				

Bibliography

Ackley, D. (2012). *A connectionist machine for genetic hillclimbing*, volume 28. Springer Science & Business Media.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybern.*, 13(5):834–846.

Baxter, J. and Bartlett, P. L. (2000). Direct gradient-based reinforcement learning. In *IEEE International Symposium on Circuits and Systems, ISCAS 2000, Emerging Technologies for the 21st Century, Geneva, Switzerland, 28-31 May 2000, Proceedings*, pages 271–274. IEEE.

Baxter, J. and Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *J. Artif. Intell. Res.*, 15:319–350.

Bellman, R. (1957). *Dynamic Programming*. Dover Publications.

Bellman, R. and Dreyfus, S. (2015). *Applied Dynamic Programming*. Princeton University Press.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*, volume 3 of *Optimization and neural computation series*. Athena Scientific.

Bottou, L. (1998). On-line learning and stochastic approximations. In *In On-line Learning in Neural Networks*, pages 9–42. Cambridge University Press.

Boucheron, S., Lugosi, G., and Bousquet, O. (2003). Concentration inequalities. In Bousquet, O., von Luxburg, U., and Rätsch, G., editors, *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176 of *Lecture Notes in Computer Science*, pages 208–240. Springer.

Boucheron, S., Lugosi, G., and Massart, P. (2009). On concentration of self-bounding functions. *Electronic Journal of Probability*, 14(none):1884 – 1899.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym. *CoRR*, abs/1606.01540.

Cassandra, A., Kaelbling, L., and Kurien, J. (1996). 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 vol.2.

Chatzilygeroudis, K. I., Vassiliades, V., Stulp, F., Calinon, S., and Mouret, J. (2020). A survey on policy search algorithms for learning robot controllers in a handful of trials. *IEEE Trans. Robotics*, 36(2):328–347.

Ciosek, K. A. and Whiteson, S. (2017). OFFER: off-environment reinforcement learning. In Singh, S. P. and Markovitch, S., editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1819–1825. AAAI Press.

Conn, A. R., Gould, N. I. M., and Toint, P. L. (2000). *Trust Region Methods*. MOS-SIAM Series on Optimization. SIAM.

Cortes, C., Mansour, Y., and Mohri, M. (2010). Learning bounds for importance weighting. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 442–450. Curran Associates, Inc.

Coulom, R. (2002). *Reinforcement Learning Using Neural Networks, with Applications to Motor Control. (Apprentissage par renforcement utilisant des réseaux de neurones, avec des applications au contrôle moteur)*. PhD thesis, Grenoble Institute of Technology, France.

Cuadras, C. M. (2002). On the covariance between functions. *Journal of Multivariate Analysis*, 81(1):19–27.

Daniel, C., Neumann, G., and Peters, J. (2012). Hierarchical relative entropy policy search. In Lawrence, N. D. and Girolami, M. A., editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 273–281. JMLR.org.

Deisenroth, M. P., Neumann, G., and Peters, J. (2013). A survey on policy search for robotics. *Found. Trends Robotics*, 2(1-2):1–142.

Deisenroth, M. P. and Rasmussen, C. E. (2011). PILCO: A model-based and data-efficient approach to policy search. In Getoor, L. and Scheffer,

T., editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 465–472. Omnipress.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. (2017). Openai baselines.

D'Oro, P., Metelli, A. M., Tirinzoni, A., Papini, M., and Restelli, M. (2020). Gradient-aware model-based policy search. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 3801–3808. AAAI Press.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1329–1338. JMLR.org.

Ernst, D., Geurts, P., and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556.

Feyzabadi, S. and Carpin, S. (2014). Risk-aware path planning using hierarchical constrained markov decision processes. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 297–303.

Frank, J., Mannor, S., and Precup, D. (2008). Reinforcement learning in the presence of rare events. In Cohen, W. W., McCallum, A., and Roweis, S. T., editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 336–343. ACM.

Furuta, K., Okutani, T., and Sone, H. (1978). Computer control of a double inverted pendulum. *Computers & Electrical Engineering*, 5(1):67–84.

Ghosh, D., Machado, M. C., and Roux, N. L. (2020). An operator view of policy gradient methods. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing*

Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Gu, S., Holly, E., Lillicrap, T. P., and Levine, S. (2017). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 3389–3396. IEEE.

Hammersley, J. (2013). *Monte carlo methods*. Springer Science & Business Media.

Hanna, J. P. et al. (2019). *Data efficient reinforcement learning with off-policy and simulated data*. PhD thesis.

Hanna, J. P. and Stone, P. (2018). Towards a data efficient off-policy policy gradient. In *2018 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 26-28, 2018*. AAAI Press.

Hanna, J. P., Thomas, P. S., Stone, P., and Niekum, S. (2017). Data-efficient policy evaluation through behavior policy search. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1394–1403. PMLR.

He, H. Y. and Owen, A. B. (2014). Optimal mixture weights in multiple importance sampling.

Hesterberg, T. C. (1988). *Advances in importance sampling*. PhD thesis, Citeseer.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

Ionides, E. L. (2008). Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311.

Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *J. Mach. Learn. Res.*, 11:1563–1600.

Kahn, H. (1950). Random sampling (monte carlo) techniques in neutron attenuation problems. i. *Nucleonics (US) Ceased publication*, 6(See also NSA 3-990).

Kahn, H. and Marshall, A. W. (1953). Methods of reducing sample size in monte carlo computations. *Oper. Res.*, 1(5):263–278.

Kakade, S. M. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In Sammut, C. and Hoffmann, A. G., editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann.

Kober, J. and Peters, J. (2009). Policy search for motor primitives in robotics. In *Advances in neural information processing systems 21*, pages 849–856, Red Hook, NY, USA. Max-Planck-Gesellschaft, Curran.

Kohl, N. and Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, April 26 - May 1, 2004, New Orleans, LA, USA*, pages 2619–2624. IEEE.

Kuzborskij, I., Vernade, C., György, A., and Szepesvári, C. (2021). Confident off-policy evaluation and selection through self-normalized importance weighting. In Banerjee, A. and Fukumizu, K., editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 640–648. PMLR.

Lange, S., Gabel, T., and Riedmiller, M. A. (2012). Batch reinforcement learning. In Wiering, M. A. and van Otterlo, M., editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 45–73. Springer.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In Bengio, Y. and LeCun, Y., editors, *ICLR*.

Maurer, A. et al. (2003). A bound on the deviation probability for sums of non-negative random variables. *J. Inequalities in Pure and Applied Mathematics*, 4(1):15.

Metelli, A. M., Papini, M., Faccio, F., and Restelli, M. (2018). Policy optimization via importance sampling. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5447–5459.

Metelli, A. M., Papini, M., Montali, N., and Restelli, M. (2020). Importance sampling techniques for policy optimization. *J. Mach. Learn. Res.*, 21:141:1–141:75.

Meyn, S. P. and Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability*. Communications and Control Engineering Series. Springer.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533.

Moore, A. (1991). *Efficient Memory-based Learning for Robot Control*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Munos, R. (2005). Error bounds for approximate value iteration. In Veloso, M. M. and Kambhampati, S., editors, *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1006–1011. AAAI Press / The MIT Press.

Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. (2018). Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018, Brisbane, Australia, May 21-25, 2018*, pages 7559–7566. IEEE.

Neumann, G. (2011). Variational inference for policy search in changing situations. In Getoor, L. and Scheffer, T., editors, *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 817–824. Omnipress.

Owen, A. B. (2013). Monte carlo theory, methods and examples.

Papini, M., Metelli, A. M., Lupo, L., and Restelli, M. (2019). Optimistic policy optimization via multiple importance sampling. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4989–4999. PMLR.

Peters, J., Mülling, K., and Altun, Y. (2010). Relative entropy policy search. In Fox, M. and Poole, D., editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press.

Peters, J. and Schaal, S. (2006). Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2006, October 9-15, 2006, Beijing, China*, pages 2219–2225. IEEE.

Peters, J. and Schaal, S. (2008). Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697.

Pirotta, M., Restelli, M., and Bascetta, L. (2013a). Adaptive step-size for policy gradient methods. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 1394–1402.

Pirotta, M., Restelli, M., Pecorino, A., and Calandriello, D. (2013b). Safe policy iteration. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 307–315. JMLR.org.

Precup, D., Sutton, R. S., and Singh, S. P. (2000). Eligibility traces for off-policy policy evaluation. In Langley, P., editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 759–766. Morgan Kaufmann.

Puterman, M. L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley.

Puterman, M. L. and Shin, M. C. (1978). Modified Policy Iteration Algorithms for Discounted Markov Decision Problems. *Management Science*, 24(11):1127–1137.

Rényi, A. (1961). On measures of entropy and information. Technical report, Hungarian Academy of Sciences Budapest Hungary.

Rummery, G. A. and Niranjan, M. (1994). On-line Q-learning using connectionist systems. Technical Report TR 166, Cambridge University Engineering Department, Cambridge, England.

Rust, J. (1996). Numerical dynamic programming in economics. In Amman, H. M., Kendrick, D. A., and Rust, J., editors, *Handbook of Computational Economics*, volume 1 of *Handbook of Computational Economics*, chapter 14, pages 619–729. Elsevier.

Scherrer, B. (2014). Approximate policy iteration schemes: A comparison. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1314–1322. JMLR.org.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P. (2015). Trust region policy optimization. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 12(1-2):1–286.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In Solla, S. A., Leen, T. K., and Müller, K., editors, *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pages 1057–1063. The MIT Press.

Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Thomas, P. S., Theocharous, G., and Ghavamzadeh, M. (2015). High-confidence off-policy evaluation. In Bonet, B. and Koenig, S., editors,

Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, pages 3000–3006. AAAI Press.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012, Vilamoura, Algarve, Portugal, October 7-12, 2012*, pages 5026–5033. IEEE.

van Erven, T. and Harremoës, P. (2014). Rényi divergence and kullback-leibler divergence. *IEEE Trans. Inf. Theory*, 60(7):3797–3820.

Veach, E. and Guibas, L. J. (1995). Optimally combining sampling techniques for monte carlo rendering. In Mair, S. G. and Cook, R., editors, *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1995, Los Angeles, CA, USA, August 6-11, 1995*, pages 419–428. ACM.

Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P., and Ba, J. (2019). Benchmarking model-based reinforcement learning. *CoRR*, abs/1907.02057.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Glossary

BEE Bellman Expectation Equations. 11, 12

BEO Bellman Expectation Operator. 11, 12

BOO Bellman Optimality Operator. 11, 13

DP Dynamic Programming. 1, 4, 12, 13

EM Expectation Maximization. xiii, 16, 17

G(PO)MDP Gradient of a (Partially Observable) Markov Decision Process. xv, 22, 23

Inf.Th. Information-Theoretic Insights. xiii, 16, 17

IS Importance Sampling. 2, 3, 4, 25, 26, 27, 28, 30, 31, 33, 35, 39, 40, 41, 48, 50

KL Kullback-Leibler. xiii, 43, 44

MC Monte Carlo. xiii, 26

MDP Markov Decision Process. 1, 4, 7, 8, 9, 10, 11, 12, 13, 15, 39, 44, 63

Mf-PS Model-free Policy Search. 17, 18

MIS Multiple Importance Sampling. 28, 29, 33, 50

ML Machine Learning. 5

Off-VM Off-policy Variance Minimization. 2, 3, 30, 39, 40, 41, 48, 53, 63

Off-PL Off-policy Learning. 2, 3, 30, 39, 40, 41, 48, 53, 63

PDIS Per-Decision Importance Sampling. xiv, 31, 32, 53, 62, 64

PE Policy Evaluation. 10

PG Policy Gradient. xiii, 16, 17, 18, 20, 21, 33, 47, 51, 53

PGT Policy Gradient Theorem. 19, 21

PI Policy Iteration. 12, 13

PO Policy Optimization. 2, 3, 4, 11, 15, 25, 48, 53, 63, 64

PO^{2}PE Policy Optimization via Optimal Policy Evaluation. xiv, xv, 3, 4, 39, 48, 49, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 64

POIS Policy Optimization via Importance Sampling. 4, 25, 37, 53, 54, 56, 57, 58, 59, 60, 64

PPO Proximal Policy Optimization. 4, 25, 36

PS Policy Search. xiii, 2, 3, 15, 16, 17, 18, 31

REPS Relative Entropy Policy Search. 4, 25, 33

RL Reinforcement Learning. xii, 1, 2, 3, 4, 5, 6, 7, 12, 13, 14, 17, 25, 30, 31, 40, 53, 54

SARSA State Action Reward State Action. 13

TRPO Trust Region Policy Optimization. 4, 25, 35, 36, 53, 54, 55, 56, 57, 59, 60, 64

VI Value Iteration. 13