Reimagining Exploration: Theoretical Insights and Practical Advancements in Policy Gradient Methods

---

Adrien Bolland

Department of Electrical Engineering and Computer Science
Faculty of Applied Sciences

This dissertation has been submitted in partial fulfillment of the requirements for the Degree of Doctor of Engineering Sciences.

Advisor

Prof. Damien Ernst

February 2025

CONTENTS

ABSTRACT

In reinforcement learning, direct policy optimization, and specifically policy gradient methods, has proven effective for solving complex control problems. However, these methods are highly sensitive to the evolution of the policy's stochasticity during learning. It is essential to maintain sufficient exploration to avoid premature convergence toward a deterministic or low-entropy policy. This thesis studies this issue, with focus on policy parameterization choices and reward-shaping methods with intrinsic exploration bonuses.

First, we analyze the influence of policy stochasticity on the optimization process. We formulate direct policy optimization within the optimization-by-continuation framework, which involves optimizing a sequence of surrogate objectives called continuations. We show that optimizing the expected return of an affine Gaussian policy, which is sufficiently stochastic either through manually controlling the variance or through regularizing the entropy, corresponds to optimizing a continuation of the expected return of an underlying deterministic policy. This continuation corresponds to the expected return filtered to remove local extrema. Hereby, we argue that policy gradient algorithms enforcing exploration can be understood as methods for optimizing policies by continuation and that the policy variance should be a history-dependent function adapted to avoid local optima.

Next, we introduce a novel analysis of intrinsic bonuses through the lens of numerical optimization. We define two criteria for the learning objective and two for the stochastic gradient estimates, using them to evaluate the policy's quality after optimization. Our analysis highlights two key effects of exploration techniques: smoothing the learning objective to remove local optima while preserving the global maximum, and modifying gradient estimates to increase the likelihood of eventually finding an optimal policy. We empirically illustrate these effects, identifying limitations and suggesting directions for future work.

Finally, we propose a new intrinsic reward bonus for exploration as in maximum-entropy reinforcement learning methods. The intrinsic reward is defined as the relative entropy of the discounted distribution of future state-action pairs, or features of these pairs. We prove that an optimal exploration policy maximizing this reward also maximizes a lower bound on the state-action value function under certain assumptions. We further show that the visitation distribution defining the intrinsic rewards is the fixed point of a contraction operator and describe how existing algorithms can be adapted to learn this fixed point. A new off-policy maximum-entropy algorithm is finally introduced. It demonstrates effective exploration and efficient computation of high-quality control policies.

ACKNOWLEDGMENTS

LIST OF PUBLICATIONS

The core part of this thesis is based on the three following publications.

- Bolland, A., Louppe, G., & Ernst, D. (2022) Policy Gradient Algorithms Implicitly Optimize by Continuation. Transactions on Machine Learning Research.
- Bolland, A., Lambrechts, G., & Ernst, D. (2024). Behind the Myth of Exploration in Policy Gradients. arXiv preprint arXiv:2402.00162.
- Bolland, A., Lambrechts, G., & Ernst, D. (2024). Off-Policy Maximum Entropy RL with Future State and Action Visitation Measures. arXiv preprint arXiv:2412.06655.

In addition, I authored and co-authored the following publications during my thesis.

- Berger, M., Bolland, A., Miftari, B., Djelassi, H., & Ernst, D. (2021). Graph-Based Optimization Modeling Language: A Tutorial.
- Boukas, I., Ernst, D., Théate, T., Bolland, A., Huynen, A., Buchwald, M., Wynants, C. & Cornélusse, B. (2021). A deep reinforcement learning framework for continuous intraday market bidding. Machine Learning, 110, 2335-2387.
- Bolland, A., Boukas, I., Berger, M., & Ernst, D. (2022). Jointly Learning Environments and Control Policies with Projected Stochastic Gradient Ascent. Journal of Artificial Intelligence Research, 73, 117-171.
- Lambrechts, G., Bolland, A., & Ernst, D. (2022). Belief states of POMDPs and internal states of recurrent RL agents: an empirical analysis of their mutual information. European Workshop on Reinforcement Learning.
- Lambrechts, G., Bolland, A., & Ernst, D. (2022). Recurrent networks, hidden states and beliefs in partially observable environments. Transactions on Machine Learning Research.
- Lambrechts, G., Bolland, A., & Ernst, D. (2023). Informed POMDP: Leveraging additional information in model-based RL. Reinforcement Learning Conference.
- Cauz, M., Bolland, A., Miftari, B., Perret, L., Ballif, C., & Wyrsch, N. (2023). Reinforcement Learning for Joint Design and Control of Battery-PV Systems. Proceedings of ECOS 2023, 2.

- Théate, T., Wehenkel, A., Bolland, A., Louppe, G., & Ernst, D. (2023). Distributional reinforcement learning with unconstrained monotonic neural networks. Neurocomputing, 534, 199-219.
- Cauz, M., Bolland, A., Wyrsch, N., & Ballif, C. (2024). Reinforcement Learning for Efficient Design and Control Co-optimisation of Energy Systems. ICML 2024 AI for Science.
- Aittahar, S., Bolland, A., Derval, G., & Ernst, D. (2024). Optimal control of renewable energy communities subject to network peak fees with model predictive control and reinforcement learning algorithms. arXiv preprint arXiv:2401.16321.
- Pirlet, M., Bolland, A., Louppe, G., & Ernst, D. (2024). Cost Estimation in Unit Commitment Problems Using Simulation-Based Inference. NeurIPS 2024 Workshop on Data-driven and Differentiable Simulations, Surrogates, and Solvers.

1

INTRODUCTION

Living and conscious beings are characterized by their ability and obligation to make decisions in a world where time is constantly slipping away. Humans are no exception to this rule. In daily life, we make trivial decisions that influence the course of our day, as what time to wake up and what meal to eat. Other decisions are more significant, such as choosing a partner or buying a house. They shape our entire lives. Some decisions must also be made collectively as a society. For instance, determining which energy source will meet our needs or deciding which countries to engage with in peace or not. The decisions we make have often an irreversible impact on future events and may influence many subsequent decisions. Hence, we take the time before acting when making important decisions. We reflect on the information available to us. Based on our observations of the world, we try to understand how it evolves and predict the influence of our choices on it. We consider the alternatives, and finally, we act to satisfy our present and future interests. This process has been repeated by every conscious being, every human, and every society since time immemorial. It is known as sequential decision-making.

The study of good and bad decisions was first addressed philosophically, and this is still partly the case. Ethics, deontology, religion, and various moral frameworks aim to determine whether a decisions is good or bad in the eyes of society. Modern mathematics, on the contrary, is not concerned with morality but with rationality (Hansson, 2005). It has answered the question of what an optimal decision is and how to make it within an axiomatic system where good and bad are measured a priori by a utility function. A rational agent observes its environment and, based on its belief about the state of that environment, makes decisions that maximize its future expected utility. To fulfill this objective, the agent must consider all the future states it will encounter and all the future decisions it will make in order to evaluate the utility associated to future events. This concept of rational agents might seem theoretical and distant from human behavior, as we are biased in our decision-making. For computers, on the contrary, rationality guides the millions of decisions they make every day, in order to maximize a predefined utility function.

At birth, rationality is far out of our hands, our decisions are arbitrary, but through trial and error we get closer to some rationality, we learn from experience. Reinforcement learning formalizes and reproduces this process. Such methods have enabled machines to learn to act rationally, revolutionizing automatic decision-making since the beginning of this cen-

tury. In reinforcement learning, an agent, initially agnostic of the environment in which it evolves, interacts with this environment with the objective of increasing its utility by exploiting the information it gathers from these interactions (Sutton and Barto, 2018). In its simplest form, the environment is a Markov decision process. It sequentially presents states describing the environment to the agent, who must act accordingly and then receives a reward after taking the action. In reinforcement learning, the agent's behavior is described by a policy, which provides the probability that the agent executes an action in a state. The utility associated with executing the policy in the environment is called the return and corresponds to the sum of all rewards collected during the interaction, discounted to measure the relative decreasing importance of future events. As the agent interacts with the environment, it refines this policy to eventually act optimally, that is, to maximize the expected utility measured through the expected return.

The reinforcement learning paradigm described earlier poses a dilemma highlighted in the seminal work by March (1991). An efficient learning algorithm must act optimally as quickly as possible, and thus execute an optimal policy as quickly as possible. However, to improve the quality of the policy, it is necessary to obtain additional information by making decisions that are likely not optimal. This is known as the exploration-exploitation dilemma. The learning algorithm must balance exploration, which involves obtaining new information, and exploitation, which consists in acting in the best possible way given the collected information. This learning problem is usually formulated in terms of regret. The regret in a state measures the difference between the expected sum of future rewards from that state, called the state value function, under an optimal policy and under the current policy. The objective is to design algorithms that minimize the total regret during learning, i.e., the sum of the regret in each visited state. Designing minimum-regret algorithms is important and requires carefully balancing exploration and exploitation.

There are many different families of reinforcement learning algorithms, each aiming to obtain, remember and exploit information in different ways. Honoring the field is complicated due to its vastness. We focus on policy gradient algorithms (Andrychowicz et al., 2020; Duan et al., 2016), which were recently used to compute control policies in complex environments and are at the heart of this thesis. In essence, a parameterized density estimator is used to approximate the policy, and its parameters are optimized by stochastic gradient ascent to maximize the expected return. The density estimator is typically composed of a parameterized distribution over the actions, e.g., a categorical or Gaussian distribution, whose distribution parameters are provided by a function of the environment state. This function itself depends on parameters, which are optimized when learning the policy. Recently, that role has been fulfilled by neural networks, which we do not face the affront of redefining once again. In order to optimize the policy, most policy gradient algorithms involve estimating the gradient of the expected return with respect to the parameters. A particularly challenging task that must be completed without knowing the components of the Markov decision process, though interactions with the environment.

There are many policy gradient algorithms, each aiming to improve the stochastic ascent algorithm, e.g., by balancing bias and variance in the estimates, using additional function approximators to better remember information and increase efficiency, or employing higher-order methods to compute the ascent directions. In practice, we observe that the most complex methods are inefficient when the policy converges too quickly to become deterministic or low-entropy. Early practitioners already proposed methods to avoid this phenomenon. Some methods involve manually modifying the distribution of actions executed in the environment. This includes adding a constant disturbance to the actions sampled from the policy, thereby manually increasing the variety of actions executed in the environment during learning (Lillicrap et al., 2015). Alternatively, noise can be added to the policy parameters instead of the actions to achieve a similar effect (Salimans et al., 2017; Sehnke et al., 2010; Jiaxin Zhang et al., 2020). Finally, the policy parameterization influences the stochasticity of the policy and can be adapted to improve learning (Chou et al., 2017; Fujita and Maeda, 2018). Other methods ensure that the policy remains sufficiently stochastic by modifying the learning objective with reward shaping. Typically, additional intrinsic reward bonuses for some states and actions are provided to promote or hinder behaviors. These bonuses encourage actions that reduce the agent's uncertainty about its environment (Burda et al., 2018; D. Pathak et al., 2017; T. Zhang, Xu, et al., 2021) or that increase the variety of states and/or actions observed during learning (M. Bellemare et al., 2016; Islam et al., 2019; Lee et al., 2019; Williams and Peng, 1991). All these methods have been interpreted as implementations of the exploration-exploitation dilemma in policy gradients, which has given rise to a considerable amount of folklore justifying the performance of these so-called exploration methods.

Without going into too much detail, recent works have established convergence guarantees and convergence rates for policy gradient algorithms. Most interestingly, non-stationary convergence toward optimal policies has been shown for different algorithms, when there is a finite number of states and actions in the environment. However, it always assumes that each state is visited with nonzero probability by the policy, and at least one of the following three conditions must hold: the exact gradients are used, the number of samples to approximate the gradient increases with the number of iterations, or the expected return is extended with the entropy of the policy (Bhandari and Russo, 2020; Cen et al., 2022; Mei, Dai, et al., 2021; Mei, Xiao, Dai, et al., 2020; Mei, Xiao, Szepesvari, and Schuurmans, 2020; Junzi Zhang et al., 2021). When there is a continuous number of actions and states, results are sparse, but global convergence has also been shown under much stronger assumptions (Y. Liu et al., 2022; Yuan et al., 2022). Interestingly, it was also recently argued that heavy-tailed policy distributions should be favored to guarantee sufficient stochasticity during learning (Bedi, Chakraborty, et al., 2022; Bedi, Parayil, et al., 2021). These results highlight more formally the influence of policy stochasticity on algorithm convergence. However, this influence is still poorly understood, and few results study the effect of the previous practical exploration methods on the outcome of policy gradient optimization.

To summarize, in practice, some form of exploration is needed to achieve good results, and in theory, algorithms converge under assumptions that are related to the stochasticity of policies and are reminiscent of exploration arguments. However, a gap remains between theory and practice, particularly in properly defining and quantifying the influence of exploration on policy gradient methods. In general, the impact of policy stochasticity on algorithm performance is still poorly understood. Similarly, the role of intrinsic bonuses remains unclear. This thesis presents three original contributions that aim to address these questions about exploration and alleviate some of the associated folklore. Each contribution is presented in a dedicated chapter and originates from a scientific paper.

The contributions are presented in this manuscript as follows. Chapter 2 first provides general background on policy gradients, which is useful for problem understanding. In Chapter 3, the influence of the choice of the policy class and its stochasticity on the learning objective is studied. More specifically, the expected return of Gaussian policies is analyzed as a function of policy variance. We propose a new interpretation of Gaussian policy optimization in which the expected return of such a policy is a surrogate objective for the return of an underlying deterministic policy, where the variance acts as a hyperparameter to control the smoothness of the surrogate. This interpretation illustrates how previous entropy control techniques can be used to optimize deterministic policies more efficiently. The learning objective function is also studied in Chapter 4, assuming that it includes intrinsic exploration. The main contribution is to quantify the influence of intrinsic bonuses on the policy gradient algorithm. In short, intrinsic exploration affects both the learning objective function and the stochastic gradient estimate distribution, which is analyzed using four new criteria. If these criteria are met, intrinsic bonuses help in computing optimal policies, which appears to be the case in practice. Chapter 5 introduces a new intrinsic reward bonus along with an efficient off-policy policy gradient algorithm. The algorithm is tested and illustrated on several benchmarks. Finally, Chapter 6 concludes this thesis with a discussion of its limitations and important challenges for future works.

2

POLICY GRADIENT METHODS AND EXPLORATION

This chapter is dedicated to recalling the necessary theory to better situate the contributions of this thesis within the literature. First, a reminder of the Markov decision process and optimal policies is given in Section 2.1. Then, we detail policy gradient methods for computing such optimal policies from interactions with the environment in Section 2.2. We first reformulate the optimization problem for computing optimal parametric policies. Next, an ascent direction for solving this optimization problem through stochastic ascent steps is derived. The section ends with a discussion on estimating policy gradient ascent directions using Monte Carlo techniques. Actor-critic algorithms, which contrast with Monte Carlo methods for estimating ascent directions in policy gradient algorithms, are discussed in Section 2.3. In practice, policy gradient and actor-critic algorithms are inefficient without additional exploration. Typical exploration methods based on reward shaping are introduced in Section 2.4. Finally, the chapter concludes with Section 2.5, which provides an overview of the main results concerning the convergence of policy gradient algorithms, with particular emphasis on the resemblance between the assumptions required for guaranteeing convergence and the exploration required in practice.

2.1 MARKOV DECISION PROCESSES

We study problems in which an agent makes sequential decisions in a stochastic environment (Sutton and Barto, 2018). The environment is modeled with an infinite-time Markov decision process (MDP) composed of a state space $\mathcal{S}$, an action space $\mathcal{A}$, an initial state distribution p_0, a transition distribution p, a bounded reward function R, and a discount factor $\gamma \in [0, 1)$. When an agent interacts in this MDP, an initial state $s_0 \sim p_0(\cdot)$ is first sampled and observed, then, the agent provides at each time step t an action a_t leading to a new state $s_{t+1} \sim p(\cdot|s_t, a_t)$. Such a sequence of states and actions $h_t = (s_0, a_0, \dots, s_{t-1}, a_{t-1}, s_t) \in H$ is called a history and H is the set of all histories of any arbitrary length. In addition, after each action a_t is executed, a bounded reward $r_t = R(s_t, a_t) \in \mathbb{R}$ is observed.

A (stochastic) history-dependent policy $\eta \in \mathcal{E} = H \to \mathcal{P}(\mathcal{A})$ is a mapping from the set of histories H to the set of probability measures on the action space $\mathcal{P}(\mathcal{A})$. A (stochastic)

Markov policy $\pi \in \Pi = \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a mapping from the state space $\mathcal{S}$ to the set of probability measures on the action space $\mathcal{P}(\mathcal{A})$. Finally, deterministic policies $\mu \in M = \mathcal{S} \to \mathcal{A}$ are functions mapping an action $a = \mu(s) \in \mathcal{A}$ to each state $s \in \mathcal{S}$. We note that for each deterministic policy μ there exists an equivalent Markov policy, where the probability measure is a Dirac measure on the action $a = \mu(s)$ in each state s. In addition, for each Markov policy, there exists an equivalent history-dependent policy only accounting for the last state in the history. We therefore write by abuse of notation that $M \subsetneq \Pi \subsetneq \mathcal{E}$. For the sake of simplicity, probability distributions are only distinguished from probability densities in Chapter 3, where we believe it is necessary for clarity. In the following of this chapter, we only discuss the case of Markov policies, and we do not enter into unnecessarily theoretical considerations related to measure theory.

When an agent interacts with an MDP, the actions it executes are drawn from a policy $\pi \in \Pi$. The value functions measure the expected discounted sum of future rewards when such a policy is executed, starting from a state s_t with the state value function $V^\pi : \mathcal{S} \to \mathbb{R}$, and starting from a state s_t followed by an arbitrary action a_t with the state-action value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. These two functions have useful recursive analytical expressions

$$V^\pi(s_t) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim p(s_t, a_t)}} \left[R(s_t, a_t) + \gamma V(s_{t+1}) \right] \tag{2.1}$$

$$Q^\pi(s_t, a_t) = \mathop{\mathbb{E}}_{s_{t+1} \sim p(s_t, a_t)} \left[R(s_t, a_t) + \gamma V(s_{t+1}) \right] . \tag{2.2}$$

Similarly, the function $J : \Pi \to \mathbb{R}$ is defined as the function mapping to any policy $\pi \in \Pi$ the expected discounted sum of rewards gathered by following this policy in the MDP. Its value $J(\pi)$ is called the expected return of the policy

$$J(\pi) = \mathop{\mathbb{E}}_{\substack{s_0 \sim p_0(\cdot) \\ a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] . \tag{2.3}$$

The expected return of a policy can be rewritten cleverly using the state-occupancy (or state-visitation) measure

$$J(\pi) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\pi,\gamma}(\cdot) \\ a \sim \pi(\cdot|s)}} \left[R(s, a) \right] . \tag{2.4}$$

where $d^{\pi,\gamma}(s) = (1-\gamma) \sum_{\Delta=0}^{\infty} \gamma^\Delta p_\Delta^\pi(s)$ is a measure of states visited during histories, and p_Δ^π is the state probability after Δ time steps following the policy π.

In the control and reinforcement learning literature, there are several definitions of optimal policies. They are most notably defined as policies which in each state have the highest state value function or which have the highest expected return. Both definitions are equivalent under certain conditions that we do not discuss. We adopt the definition

most often used when studying policy gradient algorithms, i.e., an optimal policy π^* is one with maximum expected return

$$\pi^* \in \arg\max_{\pi \in \Pi} J(\pi)\,. \tag{2.5}$$

2.2 POLICY GRADIENT

Policy gradient algorithms are part of the family of direct policy search algorithms, which optimize a policy π_θ parameterized by $\theta \in \Theta$ to find an optimal parameter value

$$\theta^* \in \arg\max_{\theta \in \Theta} J(\pi_\theta)\,. \tag{2.6}$$

Policy gradient algorithms iteratively estimate an ascent direction $\hat{d}$ using histories from the MDP and update the parameter in that direction. Depending on the ascent direction and how it is estimated from histories, different algorithms emerge.

A slightly unconventional method to derive an ascent direction for policy gradient methods is to begin with policy iteration. We see later that it unifies two families of algorithms that were originally derived independently. Policy iteration is an algorithm that starts from an arbitrary policy and successively refines its quality through two repeated steps: policy evaluation and policy improvement. Given a policy π, policy evaluation consists of computing the state-action value function Q^π. Then, given Q^π, policy improvement consists of computing the new policy π' that maximizes this value function in each state. Formally, policy improvement has an amortized formulation, where the objective is to maximize

$$J_{\pi,\beta}(\pi') = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s \sim d^{\beta,\gamma}(\cdot) \\ a \sim \pi'(\cdot|s)}} \left[Q^\pi(s,a) \right]\,. \tag{2.7}$$

Policy iteration was originally developed for discrete state and action spaces and required knowledge of the transition distribution and the reward function. The policy and state-action value function have tabular representations, and both policy evaluation and policy improvement are solved in closed form. Under some additional assumptions, including preventing convergence to a deterministic policy and ensuring that $d^{\beta,\gamma}$ is nonzero for each state, the expected return satisfies $J(\pi') > J(\pi)$, and the repeated procedure eventually finds a near-optimal policy.

In order to derive an ascent direction for policy gradient algorithms, we first show that it is possible to improve the expected return of any policy π_θ with parameter θ by maximizing an approximation of equation (2.7). Let us compute $\pi_{\theta'}$ that achieves a better expected return than π_θ. As explained previously, such a policy could be computed by solving the policy improvement problem in equation (2.7). The latter is complex to solve directly, but it is sufficient to maximize a linear approximation of $J_{\pi_\theta,\beta}(\pi'_\theta)$ in a sufficiently small ball

around θ to find a parameter θ' for which the policy $\pi_{\theta'}$ has a higher expected return. The first-order Taylor expansion around θ provides this simplified linear objective function

$$J_{\pi_\theta,\beta}(\pi_{\theta'}) = J_{\pi_\theta,\beta}(\pi_\theta) + \left\langle \theta' - \theta, \nabla_{\theta'} J_{\pi_\theta,\beta}(\pi_{\theta'})\Big|_{\theta'=\theta} \right\rangle + O\left(\langle \theta' - \theta, \theta' - \theta \rangle^2\right) , \tag{2.8}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product. The maximum of the linear function in the ball around θ is known in closed form and is written as a gradient ascent step

$$\theta' = \theta + \alpha \nabla_{\theta'} J_{\pi_\theta,\beta}(\pi_{\theta'})\Big|_{\theta'=\theta} . \tag{2.9}$$

The parameter α is proportional to the size of the ball in which the optimization is restricted and any sufficiently small value ensures that $J(\pi_{\theta'}) > J(\pi_\theta)$. The direction $\nabla_{\theta'} J_{\pi_\theta,\beta}(\pi_{\theta'})|_{\theta'=\theta}$ is therefore a valid ascent direction for developing policy gradient algorithms, providing a procedure for approximating the gradient with MDP histories.

As explained, an ascent direction that improves the expected return of a policy is provided in equation (2.9) and can be used to solve the problem in equation (2.6). This ascent direction unifies two main results from the policy gradient literature: the (on-policy) policy gradient theorem from Sutton, McAllester, et al. (1999) and the off-policy actor-critic from Degris et al. (2012). Both provide ascent directions for optimizing policies in policy gradient algorithms, which can be approximated using histories.

ON-POLICY POLICY GRADIENT ALGORITHMS The policy gradient theorem (Sutton, McAllester, et al., 1999) sets the basis of so-called on-policy algorithms and suggests optimizing parameterized policies with stochastic gradient ascent following an estimate of the gradient of the expected return

$$\nabla_\theta J(\pi_\theta) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot) \\ a\sim\pi_\theta(\cdot|s)}} \left[Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a|s) \right] . \tag{2.10}$$

This gradient corresponds to the ascent direction $\nabla_{\theta'} J_{\pi_\theta,\beta}(\pi_{\theta'})|_{\theta'=\theta}$ derived in equation (2.9) when $\beta = \pi_\theta$. There is thus an equivalence between maximizing the policy iteration objective and directly maximizing the expected return.

Policy gradient algorithms have emerged from constructing Monte Carlo estimates of this ascent direction. Assuming the state-action value function is known, computing estimates is straightforward by averaging the product of the policy score and the state-action value function over sampled histories. There are many methods for estimating the state-action value function and the ascent direction solely from histories (Baxter and Bartlett, 2001; Glynn, 1990; Peters and Schaal, 2008; Williams, 1992). We nevertheless focus on the more efficient actor-critic approach in Section 2.3. Importantly, using the policy gradient theorem to build ascent estimates $\hat{d}$ (without importance sampling) requires sampling from the policy π_θ to estimate the expectation in equation (2.10). Such algorithms are called on-

policy and are usually sample inefficient, as they require sampling new histories at each parameter update.

OFF-POLICY POLICY GRADIENT ALGORITHMS The development of so-called off-policy algorithms was initiated with a generalized policy gradient theorem by Degris et al. (2012), which suggests updating the policy parameters by following the gradient of the expected state-action value function

$$\nabla_\theta J_\beta(\pi_\theta) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\beta,\gamma}(\cdot) \\ a\sim\beta(\cdot|s)}} \left[\frac{\pi_\theta(a|s)}{\beta(a|s)} Q^{\pi_\theta}(s,a) \nabla_\theta \log \pi_\theta(a|s) \right] . \tag{2.11}$$

Degris et al. (2012) provides an extensive discussion about this new objective and the corresponding gradient estimates. We do not go into detail, as this gradient equals $\nabla_{\theta'} J_{\pi_\theta,\beta}(\pi_{\theta'})|_{\theta'=\theta}$ for an arbitrary behavior policy β, which in practice can be a past or disturbed version of the policy π_θ. Again, algorithms relying on this gradient direction perform approximate policy iteration, as introduced above.

This new gradient direction also led to the development of policy gradient algorithms known as off-policy. Here, the histories required to estimate the gradient may be generated from any behavior policy β, allowing sample reuse and improving algorithm efficiency. However, estimating the state-action value function off-policy is significantly more complex than on-policy (Levine et al., 2020; Precup, 2000).

In conclusion, policy gradient algorithms are gradient ascent algorithms where the ascent steps can be interpreted as approximate policy improvement steps. This interpretation unifies on-policy and off-policy policy gradient methods. Furthermore, on the one hand, the ascent direction for deterministic policy updates is a particular case of the previous gradient expression (Lillicrap et al., 2015; Silver, Lever, et al., 2014). On the other hand, the development may also generalize to natural policy gradient (Kakade, 2001) using a scalar product in equation (2.8) that accounts for the policy geometry (Amari, 1997), or to higher-order methods using higher-order Taylor expansions. Finally, building estimates of the gradient is straightforward but requires estimating the state-action value function, which is inefficient in practice when using only sampled histories. Many other complex algorithms and ascent heuristics exist but are out of the scope of this discussion.

2.3 ACTOR-CRITIC ALGORITHMS

One often speaks of policy gradient algorithms when the ascent directions used for policy updates are estimated solely from histories generated in the environment. In Section 2.2, the gradients commonly used as directions are given, and they can easily be estimated from histories provided with the state-action value function. The latter can itself be estimated from histories, typically using the sum of future discounted rewards generated on-policy. In contrast to policy gradients, we refer to actor-critic algorithms when the eval-

uation of the state-action value function is based on an additional function approximator. The actor then refers to the policy, and the critic to this new function approximator, which evaluates the quality of a state and/or an action.

Probably the most straightforward actor-critic algorithms are those where the state-action value function is directly approximated by a function approximator. Again, there are several ways to learn the state-action value function and to combine it with the gradient estimate to update the policy parameters, each giving rise to a particular actor-critic algorithm. We detail an algorithm, which will be used in Chapter 5, and then briefly summarize alternative methods from the literature.

Let us consider the function $Q_\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ parameterized by $\phi \in \Phi$, which is optimized to approximate the value function Q^{π_θ} of the policy π_θ for all states and actions. The quality of the approximation is measured using the L^2-norm, and the optimization problem has an amortized formulation where the objective is to minimize

$$D(\phi) = \frac{1}{2} \mathop{\mathbb{E}}_{\substack{s\sim d^{\beta,\gamma}(\cdot) \\ a\sim\beta(\cdot|s)}} \left[\left(Q_\phi(s,a) - Q^{\pi_\theta}(s,a) \right)^2 \right] . \tag{2.12}$$

This objective function can be optimized by stochastic gradient descent, following an estimate of the (negated) gradient

$$\nabla_\phi D(\phi) = \mathop{\mathbb{E}}_{\substack{s\sim d^{\beta,\gamma}(\cdot) \\ a\sim\beta(\cdot|s)}} \left[\left(Q_\phi(s,a) - Q^{\pi_\theta}(s,a) \right) \nabla_\phi Q_\phi(s,a) \right] . \tag{2.13}$$

Similarly to the policy gradient updates, this critic gradient can be estimated from histories. The outer expectation is estimated using the sample mean over all states and actions encountered within a history sampled from an arbitrary policy β. The main difficulty in computing the sample mean is estimating the state-action value function Q^{π_θ} for each state and action encountered in the history. If the behavior policy is $\beta = \pi_\theta$, then an unbiased estimate is the discounted sum of future rewards in the sampled history $Q^{\pi_\theta}(s_t, a_t) \approx \sum_{t' \geq t} r_{t'}$. Learning the value function as such with Monte Carlo evaluation is inefficient as it requires to sample new histories as the policy is updated. The Monte Carlo method is on-policy.

In contrast to Monte Carlo methods, temporal difference (TD) learning exploits the recursive expression of the state-action value function, given in Section 2.1, to evaluate the gradient without relying on full histories. The first step is to express the gradient using this recursive expression,

$$\nabla_\phi D(\phi) = \mathop{\mathbb{E}}_{\substack{s\sim d^{\beta,\gamma}(\cdot) \\ a\sim\beta(\cdot|s) \\ s'\sim p(\cdot|s,a) \\ a'\sim\pi_\theta(\cdot|s')}} \left[\left(Q_\phi(s,a) - R(s,a) - \gamma Q^{\pi_\theta}(s',a') \right) \nabla_\phi Q_\phi(s,a) \right] . \tag{2.14}$$

Second, the gradient is evaluated by substituting the unknown $Q^{\pi_\theta}(s', a')$ with the function approximator $Q_\phi(s', a')$. The (biased) gradient simplifies to

$$\nabla_\phi D(\phi) = \mathop{\mathbb{E}}_{\substack{s\sim d^{\beta,\gamma}(\cdot)\\ a\sim\beta(\cdot|s)\\ s'\sim p(\cdot|s,a)\\ a'\sim\pi_\theta(\cdot|s')}} \left[\left(Q_\phi(s,a) - R(s,a) - \gamma Q_\phi(s',a')\right)\nabla_\phi Q_\phi(s,a)\right] . \tag{2.15}$$

This final gradient expression presents several advantages. First, it can be estimated using transitions instead of histories. There is thus no need to wait for the full history to be realized (with truncation) to update the critic and the policy in the algorithm. In practice, when the behavior policy $\beta = \pi_\theta$, the transition (s, a, s', a') is generated from the policy in the environment. Otherwise, gradient estimation requires sampling a new action using the optimized policy, $a' \sim \pi_\theta(\cdot|s')$, or adding importance weights. This leads to the second advantage of this method: it can be used off-policy, i.e., the critic is learned only through interactions with β. This learning algorithm is sometimes referred to as generalized SARSA. Finally, TD-learning can be generalized and hybridized with Monte Carlo methods, leading to efficient off-policy algorithms (Sutton and Barto, 2018).

Combining the actor gradient in equation (2.11) with the critic gradient in equation (2.15), and estimating both using transitions generated from a behavior policy β and stored in a buffer $\mathcal{D}$, leads to the end-to-end off-policy actor-critic method from Algorithm 2.1. This is a simplified version of the algorithm from Degris et al. (2012), where the critic is evaluated with an improved TD-learning strategy.

Algorithm 2.1 Minibatch Off-Policy Actor-Critic

Initialize environment
Initialize the actor π_θ and the critic Q_ϕ
Initialize learning rate α
for each iteration **do**
    Observe the environment state s
    Sample an action $a \sim \beta(\cdot|s)$
    Observe reward r and next state s'
    Store the transitions (s, a, r, s') in the buffer $\mathcal{D}$
    Sample a minibatch $\mathcal{M} \subsetneq \mathcal{D}$
    Update the critic parameter

$$\phi \leftarrow \phi - \alpha \sum_{(s,a,r,s')\in\mathcal{M}} \left(Q_\phi(s,a) - r - \gamma Q_\phi(s',a')\right)\nabla_\phi Q_\phi(s,a) \ , a' \sim \pi_\theta(\cdot|s')$$

    Update the actor parameter

$$\theta \leftarrow \theta + \alpha \frac{1}{1-\gamma} \sum_{(s,a,r,s')\in\mathcal{M}} \frac{\pi_\theta(a|s)}{\beta(a|s)} Q_\phi(s,a) \nabla_\theta \log \pi_\theta(a|s)$$

end for

Many other actor-critic algorithms exist. Commonly used algorithms are advantage actor-critic (Mnih, Badia, et al., 2016) and generalized advantage actor-critic (Schulman, Moritz,

et al., 2015). In short, they differ from the methodology presented so far, as they do not try to learn the state-action value function but instead focus on estimating the advantage $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$. This estimate is then used instead of the state-action value function in the policy gradient computation. This approach is justified by the fact that subtracting a baseline independent of the action (here V^{π_θ}) from the state-action value function keeps the policy gradient unchanged while improving the quality of its estimates (Greensmith et al., 2004; Weaver and Tao, 2013). In conclusion, being exhaustive is impossible and beyond the scope of this thesis. Sutton and Barto (2018) provides a broad overview of the different TD-learning approaches for approximating value functions. Furthermore, policy gradient and actor-critic algorithms are extensively discussed in the reviews by Duan et al. (2016) and Andrychowicz et al. (2020).

2.4 INTRINSIC REWARD BONUSES FOR EXPLORATION

This chapter has so far focused on policy optimization using policy gradient or actor-critic methods to maximize the expected return. In practice, naively maximizing the expected return of policies may prove inefficient or lead to highly suboptimal policies. As discussed in Chapter 1, theory and practice suggest keeping policies sufficiently stochastic during learning, which is usually interpreted as the exploration-exploitation dilemma and the intrinsic need for algorithms to explore. A common practical method for addressing this problem while keeping the previous algorithms applicable is reward shaping, particularly through intrinsic motivation. A surrogate learning objective L is then optimized, where additional reward bonuses are provided to encourage certain behaviors.

Formally, let us consider reward-shaping strategies where the expected discounted sum of rewards is extended by K additional reward terms, called intrinsic motivation terms. The learning objective function to maximize becomes

$$L(\theta) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot) \\ a\sim\pi_\theta(\cdot|s)}} \left[R(s,a) + \sum_{i=0}^{K-1} \lambda_i R_i^{int}(s,a) \right] = J(\pi_\theta) + J^{int}(\pi_\theta)\,, \tag{2.16}$$

where λ_i are non-negative weights for each intrinsic reward, and $J^{int}(\pi_\theta)$ is the expected intrinsic return of the policy. The parameter maximizing the learning objective is denoted by $\theta^\dagger$, which we distinguish from the optimal policy parameter θ^*.

Optimizing the objective function in equation (2.16) has the advantage of theoretically allowing the previously discussed algorithms to be applied by expanding the MDP rewards with the intrinsic rewards. However, there are two limitations to this approach. First, intrinsic motivation is impelled by the exploration-exploitation dilemma, which suggests encouraging agents to take actions that reveal new information about the environment. In practice, however, it is difficult to define an intrinsic reward function that effectively promotes such behaviors. The second limitation is that the intrinsic reward is often dependent

on the policy or an environment model, such that the gradient estimates computed with policy gradient algorithms are biased.

In practice, most intrinsic motivation terms can be classified into the following two groups.

UNCERTAINTY-BASED MOTIVATIONS. It is common to provide bonuses for performing actions that reduce the agent uncertainty about an internal environment model (Burda et al., 2018; D. Pathak et al., 2017; T. Zhang, Xu, et al., 2021). The intrinsic motivation terms are then proportional to the prediction errors of that model, which is typically learned.

ENTROPY-BASED MOTIVATIONS. It is also common to provide bonuses for visiting states and/or selecting actions that are less likely in histories. On the one hand, the most common bonuses implement a tradeoff between the expected return and the expected entropy of the policy (Haarnoja, Zhou, Hartikainen, et al., 2019; Williams and Peng, 1991). On the other hand, more recent bonuses implement a similar tradeoff between the expected return and the entropy of the state-visitation measure (M. Bellemare et al., 2016; Guo et al., 2021; Haarnoja, Zhou, Hartikainen, et al., 2019; Hazan, Kakade, et al., 2019; Islam et al., 2019; Lee et al., 2019; T. Zhang, Rashidinejad, et al., 2021).

Entropy-based motivation is at the heart of this thesis, and in particular the two following.

POLICY ENTROPY. An old idea involves adding the policy entropy to the expected return of the policy in the learning objective function to implement a trade-off between locally maximizing the expected return and keeping the policy entropy high (Mnih, Badia, et al., 2016; Williams and Peng, 1991). We then speak of entropy regularization. The learning objective function can be reformulated using intrinsic reward functions

$$L(\theta) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} [R(s,a)] + \frac{\lambda}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} [-\log \pi_\theta(a|s)] \tag{2.17}$$

$$= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} \left[R(s,a) + \lambda \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} [-\log \pi_\theta(a|s)] \right] \tag{2.18}$$

$$= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} [R(s,a) - \lambda \log \pi_\theta(a|s)] \ . \tag{2.19}$$

The optimization of such a learning objective has been studied theoretically in several works. Ahmed et al. (2019) illustrated that the learning objective function was smoother with entropy regularization. Regularization also provides substantial improvements in the robustness of the resulting policy (Brekelmans et al., 2022; Husain et al., 2021; Ziebart, 2010). This approach later became known as maximum entropy reinforcement learning (Toussaint, 2009; Ziebart et al., 2008) and is at the heart of several highly efficient algorithms (Haarnoja, Tang, et al., 2017; Haarnoja, Zhou, Abbeel, and Levine, 2018; Haarnoja, Zhou, Hartikainen, et al., 2019; Schulman, X. Chen, and Abbeel, 2017).

STATE-VISITATION ENTROPY. Hazan, Kakade, et al. (2019) were the first to intrinsically motivate agents to visit states uniformly in histories. The proposed learning objective is the sum of the expected return of the policy and the entropy of the discounted state-visitation measure, which can also be written using intrinsic rewards

$$L(\theta) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} [R(s,a)] + \frac{\lambda}{1-\gamma} \mathop{\mathbb{E}}_{s\sim d^{\pi_\theta,\gamma}(\cdot)} [-\log d^{\pi_\theta,\gamma}(s)] \tag{2.20}$$

$$= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} \left[R(s,a) + \lambda \mathop{\mathbb{E}}_{s\sim d^{\pi_\theta,\gamma}(\cdot)} [-\log d^{\pi_\theta,\gamma}(s)] \right] \tag{2.21}$$

$$= \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} [R(s,a) - \lambda \log d^{\pi_\theta,\gamma}(s)] \ . \tag{2.22}$$

Several recent works have developed algorithms to maximize the entropy of the discounted state-visitation measure or, alternatively, the stationary state-visitation measure, and are extensively discussed in Chapter 5.

Finally, it is important to note that in both cases, the intrinsic rewards depend on the policy parameters. The previous policy gradient algorithms neglect the partial derivatives with respect to these parameters and therefore produce biased gradient estimates of the learning objective function. In the two previous cases, the formulation in equation (2.18) is equivalent to that in equation (2.19), and similarly, the formulation in equation (2.21) is equivalent to that in equation (2.22). In both cases, the two equivalent formulations include the MDP reward and an additional intrinsic reward function. However, they do not yield the same gradient estimates, since the partial derivatives of the intrinsic reward functions with respect to θ are neglected. In equation (2.18) and equation (2.21), intrinsic rewards do not depend on states and actions, and the (biased) gradient estimates of the corresponding expected intrinsic return is therefore zero at each iteration if computed using simple Monte Carlo methods. As a result, they do not provide any additional exploration in practice. Equation (2.19) and equation (2.22) should be preferred, as they yield nonzero gradient estimates, and are also easier to estimate. They still remain policy-dependent, and the rewards are non-stationary during optimization, raising questions about convergence.

2.5 CONVERGENCE OF POLICY GRADIENT ALGORITHMS

Policy gradient and actor-critic algorithms have historically been introduced with little concern for their convergence. Early works were mostly limited to the construction of unbiased gradient estimates. This is straightforward for the gradients presented above when using sampled histories only or when using a critic that satisfies the compatibility property set out by Sutton, McAllester, et al. (1999). Policies computed with off-policy policy gradients were also studied in the seminal paper by Degris et al. (2012), which concluded that they correspond to stationary points.

More generally, policy gradients are stochastic gradient ascent methods and therefore converge to a stationary point if the gradient estimates are unbiased and the variance of these estimates is bounded (Bottou, 1998; Kakade and Langford, 2002). Depending on the structure of the objective function, this stationary point is either a local or a global optimum. Recent research has studied the global convergence of on-policy policy gradient algorithms and showed that they converge toward globally optimal solutions in some situations. Early work demonstrated that global convergence could be achieved for concave policy parameterization (Scherrer and Geist, 2014) and for linear quadratic regulators learned by policy gradient (Fazel et al., 2018).

In the general case of processes with discrete state and action spaces, global and non-asymptotic convergence was first shown for natural (tabular) and softmax policy parameterizations when the exact (on-policy) gradients are used in the ascent algorithm (Agarwal et al., 2020; Bhandari and Russo, 2019) and later for more general energy-based policies (Wang et al., 2019). These convergence rates were afterwards refined under different assumptions for vanilla and natural policy gradient algorithms (Bhandari and Russo, 2020; Cen et al., 2022; Mei, Dai, et al., 2021; Mei, Xiao, Dai, et al., 2020; Mei, Xiao, Szepesvari, and Schuurmans, 2020; Junzi Zhang et al., 2021). Again, for discrete state and action spaces, global convergence was also shown for more recent algorithms, including trust region policy optimization and proximal policy optimization (B. Liu et al., 2019; Shani et al., 2020; Zhao et al., 2019), and in the case of decision processes where the utility is a concave function of the state-action occupancy measure (Junyu Zhang et al., 2020). Interestingly, previous guarantees hold under some assumptions, always including that the initial state-action occupancy is nonzero everywhere, and that at least one of the following three conditions is respected: the exact gradients are used, the batch size increases with the number of iterations, or the expected return is extended with the entropy of the policy. These conditions relate to the need for sufficient stochasticity of the policy.

For Markov decision processes with continuous state and action spaces, results are weaker. Convergence toward stationary points was shown under some assumptions (Bhatt et al., 2019; Xiong et al., 2022; K. Zhang et al., 2020). Global convergence can also be achieved but, to the best of our knowledge, explicitly requires imposing some form of gradient domination (Y. Liu et al., 2022; Yuan et al., 2022), which is implied under weaker assumptions in discrete state and action spaces. It was also recently argued that heavy-tailed policy distributions should be favored to guarantee sufficient stochasticity in the policy (Bedi, Chakraborty, et al., 2022; Bedi, Parayil, et al., 2021).

Finally, the influence of intrinsic rewards on the quality of the final policy and on convergence has received little attention in the scientific literature. As mentioned above, some previous convergence results explicitly require the addition of entropy regularization in the gradient computation, but to the best of our knowledge, no generic result exists.

3

POLICY GRADIENT ALGORITHMS IMPLICITLY OPTIMIZE BY CONTINUATION

Prologue

This chapter is based on the following publication: *Bolland, A., Louppe, G., & Ernst, D. (2022) Policy Gradient Algorithms Implicitly Optimize by Continuation. Transactions on Machine Learning Research.*

Direct policy optimization in reinforcement learning is usually solved with policy gradient algorithms, which optimize policy parameters via stochastic gradient ascent. This paper provides a new theoretical interpretation and justification of these algorithms. First, we formulate direct policy optimization in the optimization by continuation framework. The latter is a framework for optimizing nonconvex functions where a sequence of surrogate objective functions, called continuations, are locally optimized. Second, we show that optimizing affine Gaussian policies and performing entropy regularization can be interpreted as implicitly optimizing deterministic policies by continuation. Based on these theoretical results, we argue that exploration in policy gradient algorithms consists in computing a continuation of the expected return of the policy at hand, and that the variance of policies should be history-dependent functions adapted to avoid local extrema rather than to maximize the expected return of the policy.

3.1 INTRODUCTION

Applications where one has to control an environment are numerous and solving these control problems efficiently is the preoccupation of many researchers and engineers. Reinforcement learning (RL) has emerged as a solution when the environments at hand have complex and stochastic dynamics (Sutton and Barto, 2018). Direct policy optimization and more particularly (on-policy) policy gradients are methods that have been successful in recent years. These methods, reviewed by Duan et al. (2016) and Andrychowicz et al. (2020), all consist in parameterizing a policy (most often with a neural network) and adapting the parameters with a local-search algorithm in order to maximize the expected sum of rewards received when the policy is executed, called the expected return of the policy. We distinguish two basic elements that determine the performance of these methods. As first element, we have the formalization of the optimization problem. It is defined through

two main choices: the (functional) parametrization of the policy and the learning objective function, which mostly relies on adding an entropy regularization term to the expected return. As second element, there is the choice of the local-search algorithm to solve the optimization problem – we focus on stochastic gradient ascent methods in this study.

The policy parameterization is the first formalization choice. In theory, there exists an optimal deterministic policy (Sutton and Barto, 2018), which can be optimized by deterministic policy gradient (Silver, Lever, et al., 2014) with a guarantee of converging towards a stationary solution (Xiong et al., 2022). However, this approach may give poor results in practice as it is subject to convergence towards local optima (Silver, Lever, et al., 2014). It is therefore usual to optimize stochastic policies where this problem is mitigated in practice (Andrychowicz et al., 2020; Duan et al., 2016). For discrete state and action spaces, theoretical guarantees of global convergence hold for softmax or direct policy parameterization (Agarwal et al., 2020; Bhandari and Russo, 2019; Junzi Zhang et al., 2021). In the general case of continuous spaces, these results no longer hold and only convergence towards stationarity can be ensured under strong hypotheses (Bedi, Parayil, et al., 2021; Bhatt et al., 2019; K. Zhang et al., 2020). Recently, convergence under milder assumptions was established assuming that the policy follows a heavy-tailed distribution, which guarantees a sufficiently spread distribution of actions (Bedi, Chakraborty, et al., 2022). Nevertheless, most of the empirical works have focused on (light-tailed) Gaussian policies (Andrychowicz et al., 2020; Duan et al., 2016) for which convergence is thus not ensured in the general case (Bedi, Chakraborty, et al., 2022). The importance of a sufficiently spread distribution in policy gradient had already been observed in early works and was loosely interpreted as exploration (Lillicrap et al., 2015; Mnih, Badia, et al., 2016). This concept originally introduced in bandit theory and value-based RL, where it consists in selecting a suboptimal action to execute in order to refine a statistical estimate (Simon, 1955; Sutton and Barto, 2018), is to our knowledge not well defined for direct policy optimization. Other empirical works also showed that relying on Beta distributions when the set of actions is bounded within an interval outperformed Gaussian policies (Chou et al., 2017; Fujita and Maeda, 2018). As a side note, another notable advantage of stochastic policies is the possibility to rely on information geometry and use efficient trust-region methods to speed up the local-search algorithms (Cen et al., 2022; Shani et al., 2020). In summary, no consensus has yet been reached on the exact policy parameterization that should be used in practice.

The second formalization choice is the learning objective and more particularly the choice of entropy regularization. Typically, a bonus enforcing the uniformity of the action distribution is added to the rewards in the objective function (Haarnoja, Zhou, Hartikainen, et al., 2019; Williams and Peng, 1991). Intuitively, it avoids converging too fast towards policies with small spread, which are subject to being locally optimal. More general entropy regularizations were applied for encouraging high-variance policies while keeping the distribution sparse (Nachum et al., 2016) or enforcing the uniformity of the state-visitation distribution in addition to the action distribution (Islam et al., 2019). Again, no consensus is reached about the best regularization to use in practice.

The importance of introducing sufficient stochasticity and regularizing entropy is commonly accepted in the community. Some preliminary research has been conducted to develop a theoretical foundation for this observation. Ahmed et al. (2019) proposed an empirical analysis of the impact of the entropy regularization term. They concluded that adding this term yields a smoothed objective function. A local-search algorithm will therefore be less prone to convergence to local optima. This problem was also studied by Husain et al. (2021). They proved that optimizing a policy by regularizing the entropy is equivalent to performing a robust optimization against changes in the reward function. This result was recently reinterpreted by Brekelmans et al. (2022) who deduced that the optimization is equivalent to a game where one player adapts the policy while an adversary adapts the reward. The research papers that have been reviewed concentrate solely on learning objectives in the context of entropy regularization, leaving unanswered the question of the relationship between a policy's expected return and the distribution of actions. This question is of paramount importance for understanding how the formalization of the direct policy optimization problem impacts the resulting control strategy.

In this chapter, we propose a new theoretical interpretation of the effects of the action distribution on the objective function. Our analysis is based on the theory of optimization by continuation (Allgower and Georg, 1980), which consists in locally optimizing a sequence of surrogate objective functions. The latter are called continuations and are often constructed by filtering the optimization variables in order to remove local optima. Our main contributions are twofold. First, we define a continuation for the expected return of policies and formulate direct policy optimization in the optimization by continuation framework. Second, based on this framework, we study different formulations, i.e., policy parameterization and entropy regularization, of direct policy optimization. Several conclusions are drawn from the analysis. First, we show that the continuation of the expected return of a deterministic policy is equal to the expected return of a Gaussian policy. Second, we show that the continuation of the expected return of a Gaussian policy equals the expected return of another Gaussian policy with scaled variance. We then derive from the previous results that optimizing Gaussian policies using policy gradient algorithms and performing regularization can be interpreted as optimizing deterministic policies by continuation. In this regard, exploration as it is usually understood in policy gradients, consists in computing the continuation of the expected return of the policy at hand. Finally, we show that for a more general continuation, the continuation of the expected return of a deterministic policy equals the expected return of a Gaussian policy where the variance is a function of the observed history of states and actions. These results provide a new interpretation for the variance of a policy: it can be seen as a parameter of the policy gradient algorithm instead of an element of the policy parameterization. Moreover, to fully exploit the power of continuations, the variance of a policy should be a history-dependent function iteratively adapted to avoid the local extrema of the expected return.

Optimization by continuation provides or aims to provide two main practical advantages to solve optimization problems. First, these methods smooth the objective function and al-

low the application of gradient-based optimization methods, as discussed in the literature on variational optimization (Staines and Barber, 2012) and applied by Murray and K.-M. Ng (2010) to discrete optimization problems. Second, it enables to compute the global optimum of the optimization problem. This global optimum can be reached for example assuming that the optimum of the first continuation is simple to compute, and that the path of local optima of each continuation leads to the global optimum of the problem (Allgower and Georg, 1980). Theoretical guarantees on convergence are still scarce in the literature. Several works focus on Gaussian continuations where the continuations are convolutions of the objective function by Gaussian kernels (Mobahi and Fisher III, 2015). It is particularly useful when the objective function is concave, ensuring smoothness of the surrogate optimization problems and providing convergence guarantees to the global optimal solution (Nesterov and Spokoiny, 2017). Interestingly, a recent work links these continuations to concave envelopes (Mobahi and Fisher, 2015). Another notable result holds for similar continuations relying on uniform kernels for building the surrogate problems where convergence to the global solution is guaranteed under certain assumptions on the objective function (Hazan, Levy, and Shalev-Shwartz, 2016). In the general case, finding the right sequence of continuations for achieving convergence to the global optimum remains heuristic and problem-dependent. The framework of optimization by continuation can nevertheless provide valuable insights, as will be further explored in this work.

Despite the lack of theoretical guarantees, optimization by continuation has found successful machine learning applications, including image alignment (Mobahi, Zitnick, and Ma, 2012), greedy layerwise training of neural networks (Bengio, 2009), and neural network training by iteratively increasing the non-linearity of the activation functions (H. N. Pathak and Paffenroth, 2019). To our knowledge, optimization by continuation has never yet been applied to direct policy optimization. However, optimizing a distribution over the policy parameters rather than directly optimizing the policy is a reinforcement learning technique that has been used to perform direct policy optimization (Salimans et al., 2017; Sehnke et al., 2010; Jiaxin Zhang et al., 2020). Among other things, it decreases the variance of the gradient estimates in some cases. If this distribution over policy parameters is a Gaussian, it is furthermore by definition equivalent to optimizing the policy by Gaussian continuation (Mobahi and Fisher III, 2015). Another method, called reinforcement learning with logistic reward-weighted regression (Peters and Schaal, 2007; Wierstra et al., 2008), consists in optimizing a surrogate objective of the expected return. The surrogate is the expected utility of the sum of rewards. Originally justified relying on the field of decision theory (Chernoff and Moses, 2012), it can equivalently be seen as an optimization by continuation method.

The chapter is organized as follows. In Section 3.2, the background of direct policy optimization is reminded. The framework for optimizing policies by continuation is developed in Section 3.3 and theoretical results relating the expected return of policies to their continuations are presented in Section 3.4. In Section 3.5, these results are used for elaborating

on the formulations of direct policy optimization. Finally, the results are summarized and further works discussed in Section 3.6.

3.2 PRELIMINARIES AND THEORETICAL BACKGROUND

Let us consider problems in which an agent interacts with a Markov decision process (MDP) as formalized in Chapter 2. The processes we consider here have continuous state and action spaces, when not specified otherwise. We therefore work mainly on policies in continuous probability spaces and represent them using their density function. Finally, there is a particular focus on history-dependent policies, and we thus briefly redefine the problem statement in this context.

Let $(\mathcal{S}, \mathcal{A}, p_0, p, R, \gamma)$ be an MDP and let $\eta_\theta \in \mathcal{E}$ be a policy parameterized by the real vector $\theta \in \mathbb{R}^{d_\Theta}$. The objective of the optimization problem is to find the optimal parameter $\theta^* \in \mathbb{R}^{d_\Theta}$ such that the expected return of the policy is maximized:

$$\theta^* = \underset{\theta \in \mathbb{R}^{d_\Theta}}{\operatorname{argmax}} \, J(\eta_\theta) \,. \tag{3.1}$$

In this chapter, we only consider on-policy policy gradient algorithms.

DETERMINISTIC POLICIES. It is possible to approximate a solution of the optimization problem equation (3.1) for a deterministic policy parameterized with a function approximator $\mu_\theta \in M$ by stochastic gradient ascent using the deterministic policy gradient theorem (Silver, Lever, et al., 2014). Nevertheless, optimizing deterministic policies with *inadequate exploration* (i.e., without sufficient additional policy randomization) usually results in locally optimal policies with poor performance (Silver, Lever, et al., 2014).

GAUSSIAN POLICIES. In direct policy optimization, most of the works focus on learning Markov policies where the actions follow a Gaussian distribution whose mean and covariance matrix are parameterized function approximators (Andrychowicz et al., 2020; Duan et al., 2016). More precisely, a parametrized Gaussian policy $\pi_\theta^{GP} \in \Pi$ is a policy where the actions follow a Gaussian distribution of mean $\mu_\theta(s)$ and covariance matrix $\Sigma_\theta(s)$ for each state s and parameter θ, it thus has the following density:

$$\pi_\theta^{GP}(a|s) = \mathcal{N}(a|\mu_\theta(s), \Sigma_\theta(s)) \,. \tag{3.2}$$

AFFINE POLICIES. A parameterized policy (deterministic or stochastic) is said to be affine, if the function approximators used to construct the functional form of the policy are affine functions of the parameter θ. Formally, each function approximator f_θ of a history-dependent policy has the following form $\forall h \in H$:

$$f_\theta(h) = a(h)^T \theta + b(h) \,, \tag{3.3}$$

where a and b are functions building features from the histories. Such policies are often considered in theoretical studies (Busoniu et al., 2017) and perform well on complex tasks in practice (Rajeswaran et al., 2017).

3.3 OPTIMIZING POLICIES BY CONTINUATION

In this section, we introduce the optimization by continuation methods and formulate direct policy optimization in this framework.

3.3.1 *Optimization by Continuation*

Optimization by continuation (Allgower and Georg, 1980) is a technique used to optimize nonconvex functions with the objective of avoiding local extrema. A sequence of optimization problems is solved iteratively using the optimum of the previous iteration. Each problem consists in optimizing a deformation of the original function and is typically solved by local search. Through the iterations, the function is less and less deformed. Such procedure is also sometimes referred to as graduated optimization (Blake and Zisserman, 1987) or optimization by homotopy (Watson and Haftka, 1989), as the homotopy of a function refers to its deformation in topology.

Formally, let $f : \mathcal{X} \to \mathbb{R}$ be the real-valued function to optimize. Let $g : \mathcal{Y} \to \mathbb{R}$ be another real-valued function used for building the deformation of f. Finally, let the conditional distribution function $p : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ be the mapping from an optimization variable $x \in \mathcal{X}$ to the set of probability measures $\mathcal{P}(\mathcal{Y})$, such that $p(y|x)$ is the associated density function for any random event $y \in \mathcal{Y}$ given $x \in \mathcal{X}$. The continuation of the function f under the distribution p and deformation function g is defined as the function $f^p : \mathcal{X} \to \mathbb{R}$ such that $\forall x \in \mathcal{X}$:

$$f^p(x) = \mathop{\mathbb{E}}_{y \sim p(\cdot|x)} \left[g(y) \right] . \tag{3.4}$$

For the optimization by continuation described hereafter, there must exist a conditional distribution p^* for which f^p equals f in the limit as p approaches p^*. A typical example is to choose the function g equal to f, and to use a Gaussian distribution with a constant diagonal covariance matrix for the distribution p. We then have so-called Gaussian continuations (Mobahi and Fisher III, 2015).

Finally, optimizing a function f by continuation involves iteratively locally optimizing its continuation for a sequence of conditional distributions approaching p^* with decreasing spread. Formally, let $p_0 \succ p_1 \succ \cdots \succ p_{I-1}$ be a sequence of conditional distributions (monotonically) approaching p^* with strictly decreasing covariance matrices[1]. Then, optimizing f by continuation consists in locally optimizing its continuation f^{p_i} with a local-search

1 In this work, we consider the L^2-norm of functions and the Loewner order over the set of covariance matrices (Siotani, 1967).

algorithm initialized at x_i^* for each iteration i. This general procedure is summarized in Algorithm 3.1. Particular instances of this algorithm are described by Hazan, Levy, and Shalev-Shwartz (2016) and Shao et al. (2019) for Gaussian continuations.

In practice, the optimization process can be approximated by performing a limited number of local-search iterations at each step of the optimization by continuation. In the following sections, we consider that each optimization of the continuation f^{p_i} is approximated with a single gradient ascent step and that the continuation distribution sequence $p_0 \succ p_1 \succ \cdots \succ p_{I-1}$ is constructed by iteratively reducing the variance of the distribution p_i. Note that if this variance reduction is sufficiently slow, and the stepsize is well chosen, a single gradient ascent step enables to accurately approximate x_i^*.

Algorithm 3.1 Optimization by Continuation

1: Provide a sequence of I functions $p_0 \succ p_1 \succ \cdots \succ p_{I-1}$
2: Provide an initial variable value $x_0^* \in \mathcal{X}$ for the local search
3: **for** $i = 0, 1, \ldots, I-1$ **do**
4: $\quad x_{i+1}^* \leftarrow$ Optimize the continuation f^{p_i} by local search initialized at x_i^*
5: **end for**
6: **return** x_I^*

3.3.2 *Continuation of the Expected Return of a Policy*

The direct policy optimization problem usually consists in maximizing a nonconvex function. Optimization by continuation is thus a good candidate for computing a solution. In this section, we introduce a novel continuation adapted to the expected return of policies.

The expected return of a policy depends on the probability of a sequence of actions through the product of the density $\eta_\theta(a_t|s_t)$ of each action a_t for a given parameter θ. We define the continuation of interest as the expectation of the expected return where each factor in the product of densities depends on a different parameter vector. This expectation is taken according to a distribution that disturbs these parameter vectors at each time step with a variance depending on the history. Formally, using the notations from Section 3.3.1, we optimize the function f that for all $x = \theta$ equals the expected return, $f(\theta) = J(\pi_\theta)$, over the set $\mathcal{X} = \mathbb{R}^{d_\Theta}$. Let the covariance function $\Lambda : H \to \mathbb{R}^{d_\Theta \times d_\Theta}$ be a function mapping a history $h_t \in H$ to a covariance matrix $\Lambda(h_t)$. Let the continuation distribution q be a distribution such that $q(\theta_t|\theta, \Lambda(h_t))$ is the density of θ_t distributed with mean θ and covariance matrix $\Lambda(h_t)$. Then, let $\mathcal{Y} = \left(\mathcal{S} \times \mathcal{A} \times \mathbb{R}^{d_\Theta}\right)^{\mathbb{N}}$ be the set of (infinite) sequences of states, actions and parameters and let p and g, the two functions defining the continuation, be as follows:

$$p(y|x) = p(s_0) \prod_{t=0}^{\infty} \eta_{\theta_t}(a_t|h_t) p_\theta(\theta_t|h_t) p(s_{t+1}|s_t, a_t) \tag{3.5}$$

$$g(y) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\,, \tag{3.6}$$

where $p_\theta(\theta_t|h_t) = q(\theta_t|\theta, \Lambda(h_t))$ such that the spread of p_θ depends on the function Λ. Taken together, the continuation $f^q_\Lambda = f^p$ of the expected return of the policy $\eta_\theta \in \mathcal{E}$ corresponding to the distribution q and covariance function Λ, is defined $\forall \theta \in \mathbb{R}^{d_\Theta}$ as:

$$f^q_\Lambda(\theta) = \mathop{\mathbb{E}}_{\substack{s_0 \sim p_0(\cdot) \\ \theta_t \sim q(\cdot|\theta, \Lambda(h_t)) \\ a_t \sim \eta_{\theta_t}(\cdot|h_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] . \tag{3.7}$$

Finally, the continuation equation (3.7) converges towards the expected return of η_θ in the limit as the covariance function Λ approaches zero, as required in Section 3.3.1.

This continuation is expected to be well-suited for removing local extrema of the expected return for three main reasons. First, marginalizing the variables of a function as in our continuation is expected to smooth this function and therefore remove local extrema – the particular case of Gaussian blurring has been widely studied in the literature (Mobahi and Fisher, 2015; Nesterov and Spokoiny, 2017). Second, we underline the interest of considering a continuation in which the disturbance of the policy parameters may vary based on the time step. Indeed, changing the parameter vector of the policy at different time steps (and changing the action distributions) may modify the objective function in significantly different ways. Third, we justify the factorization of the conditional distribution p_θ equation (3.5) by the causal effect of actions in the MDP. As the actions only influence the rewards to come, the past history is expected to provide a sufficient statistic for disturbing the parameters in order to remove local optima. We therefore chose parameter probabilities conditionally independent given the past history. This history-dependency is encoded through the covariance function Λ in equation (3.7).

Maximizing f^q_Λ to solve the optimization problem from Algorithm 3.1 is a complicated task. A common local-search algorithm used in machine learning is stochastic gradient ascent, which is known for performing well on several complex functions depending on many variables (Bottou, 2010). The gradient of f^q_Λ can be computed by Monte-Carlo sampling applying the reparameterization trick (Goodfellow et al., 2016) for simple continuation distributions or relying on the REINFORCE trick (Williams, 1992) in the more general case. Due to the complex time dependencies of the random events, these vanilla gradient estimates have practical limitations: the estimates may have large variance, the infinite horizon shall be truncated, and the direction provided is computed in the Euclidean space of parameters rather than in a space of distributions (Peters and Schaal, 2008). Finally, the evaluation of the continuation and its derivatives require one to sample parameter vectors, which may be computationally expensive for complex high-dimensional distributions. The study of different continuation distributions and the application of the optimization procedure from Algorithm 3.1 to practical problems is left for further works. In this work, we rather rely on the continuation to study existing direct policy optimization algorithms. To this end, we show in the next section that maximizing the continuation defined by equa-

tion (3.7) is equivalent to solving a direct policy optimization problem for another policy, called a mirror policy.

3.4 MIRROR POLICIES AND CONTINUATIONS

This section is dedicated to the interpretation of the continuation of the expected return of a policy. We show it equals the expected return of another policy, called a mirror policy. The existence and closed form of mirror policies is also discussed.

3.4.1 *Optimizing by Continuation with Mirror Policies*

Definition 1. *Let $(\mathcal{S}, \mathcal{A}, p_0, p, R, \gamma)$ be an MDP and let $\eta_\theta \in \mathcal{E}$ be a history-dependent policy parameterized with the vector $\theta \in \mathbb{R}^{d_\Theta}$. In addition, let f^q_Λ be the continuation of the expected return of the policy η_θ corresponding to a continuation distribution q and covariance function Λ as defined in equation (3.7). We call a mirror policy of the original policy η_θ, under the continuation distribution q and covariance function Λ, any history-dependent policy $\eta'_\theta \in \mathcal{E}$ such that $\forall \theta \in \mathbb{R}^{d_\Theta}$:*

$$f^q_\Lambda(\theta) = J(\eta'_\theta)\,. \tag{3.8}$$

Let us assume we are provided with the continuation f^q_Λ of the expected return of an original policy η_θ depending on the parameter θ that shall be optimized. In addition, let us assume we can compute a mirror policy η'_θ for the original policy η_θ. By Definition 1, the continuation of the original policy equals the expected return of the mirror policy for all θ. In addition, under smoothness assumptions, all their derivatives are equal too. Therefore, maximizing the continuation of an original policy by stochastic gradient ascent can be performed by maximizing the expected return of its mirror policy by policy gradient. Applying state-of-the-art policy gradient algorithms on the mirror policies for optimizing the continuations in Algorithm 3.1 may alleviate some of the shortcomings of the optimization procedure described earlier.

3.4.2 *Existence and Closed Form of Mirror Policies*

In this section, we first show that there always exists a mirror policy. In addition, several closed forms are provided depending on the original policy, the continuation distribution, and the covariance function.

Theorem 1. *For any original history-dependent policy $\eta_\theta \in \mathcal{E}$ parameterized with the vector $\theta \in \mathbb{R}^{d_\Theta}$ and for any continuation distribution q and covariance function Λ, there exists a mirror history-dependent policy $\eta'_\theta \in \mathcal{E}$ of the original policy η_θ that writes as:*

$$\eta'_\theta(a|h) = \mathop{\mathbb{E}}_{\theta' \sim q(\cdot|\theta, \Lambda(h))} \left[\eta_{\theta'}(a|h)\right]\,. \tag{3.9}$$

Theorem 1 guarantees the existence of mirror policies. Such a mirror policy is a function depending on the same parameters as its original policy but that has a different functional form and may therefore provide actions following a different distribution compared to the original policy.

Theorem 1 leads to two important corollary results. First, as demonstrated in Theorem 2 in Appendix 3.A, let η'' be a mirror policy of η' and let η' be a mirror policy of the original policy η of the form of equation (3.9). Then, there exists a continuation for which η'' is a mirror policy of the original policy η. It follows that the expected return of the mirror policy of another mirror policy is itself equal to a continuation of the original policy. Second, Theorem 1 also reveals that for a given original policy and continuation distribution, the variance of the mirror policy is defined through the continuation covariance function Λ. Furthermore, we remind that the variance of the continuation is an hyperparameter that shall be selected for each iteration of the optimization by continuation, see Section 3.3. This choice of hyperparameter is thus reflected as the choice of the variance of a mirror policy. The expert making this choice sees the effect of the disturbed parameters on the environment through the variance of the mirror policy. From a practical perspective, it is probably easier to quantify the effect on the local extrema depending on the variance of the mirror policy rather than depending on the variance of the continuation.

Property 3.4.1. *Let the original policy $\pi_\theta \in \Pi$ be a Markov policy and let the covariance function depend solely on the last state in the history. Then, there exists a mirror Markov policy $\pi'_\theta \in \Pi$.*

Property 3.4.1 is an intermediate result providing sufficient assumptions on the continuation for having mirror Markov policies. Note that for this type of continuation, the parameters of the policy are disturbed independently of the history followed by the agent.

Property 3.4.2. *Let the original policy $\pi_\theta^{GP} \in \Pi$ be a Gaussian policy as defined in equation (3.2) with affine function approximators. Let the covariance function depend solely on the last state in the history and let the distribution q be a Gaussian distribution. Then, there exists a mirror Markov policy $\pi'_\theta \in \Pi$ such that for all states $s \in \mathcal{S}$, it converges towards a Gaussian policy in the limit as the affine coefficients of the covariance matrix $\Sigma_\theta(s)$ approaches zero ($\|\nabla_\theta \Sigma_\theta(s)\| \to 0$):*

$$\pi'_\theta(a|s) \to \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(s))\,, \tag{3.10}$$

where $\Sigma'_\theta(s) = C_\theta(s) + \Sigma_\theta(s)$ and $C_\theta(s) = \nabla_\theta \mu_\theta(s)^T \, \Lambda(s) \, \nabla_\theta \mu_\theta(s)$.

Under the assumptions of Property 3.4.2, a mirror policy can be approached by a policy that only differs from the original one by having a variance which is increased by the term $C_\theta(s)$ proportional to the variance of the continuation. In particular, when the variance of the original policy π_θ^{GP} is solely dependent on the state, then $\|\nabla_\theta \Sigma_\theta(s)\| = 0$ and $\pi'_\theta(a|s) = \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(s))$. In this case, for any θ, the covariance matrix of this mirror policy is additionally bounded from below such that $\Sigma'_\theta(s) \succeq C_\theta(s)$.

Property 3.4.3. *Let the original policy $\mu_\theta \in M$ be an affine deterministic policy. Let the covariance function depend solely on the last state in the history and let the distribution q be a Gaussian distribution. Then, the Markov policy $\pi_\theta^{GP'} \in \Pi$ is a mirror policy:*

$$\pi_\theta^{GP'}(a|s) = \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(s)) \,, \tag{3.11}$$

where $\Sigma'_\theta(s) = \nabla_\theta \mu_\theta(s)^T \, \Lambda(s) \, \nabla_\theta \mu_\theta(s)$.

Therefore, under some assumptions, disturbing a deterministic policy and optimizing it afterwards can be interpreted as optimizing the continuation of the expected return of this policy.

Property 3.4.4. *Let the original policy $\mu_\theta \in M$ be an affine deterministic policy. Let the distribution q be a Gaussian distribution. Then, the policy $\eta'_\theta \in \mathcal{E}$ is a mirror policy:*

$$\eta'_\theta(a|h) = \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(h)) \,, \tag{3.12}$$

where $\Sigma'_\theta(h) = \nabla_\theta \mu_\theta(s)^T \, \Lambda(h) \, \nabla_\theta \mu_\theta(s)$.

Property 3.4.4 extends Property 3.4.3 to more general continuation distributions. This extension is used later to justify the interest of optimizing history-dependent policies in order to optimize an underlying deterministic policy by continuation. The theorem and properties are shown in Appendix 3.A.

3.5 IMPLICIT OPTIMIZATION BY CONTINUATION

In this section, two formulations, i.e., a parameterized policy and a learning objective each, used by several policy gradient algorithms are analyzed relying on original and mirror policies. In Section 3.5.1, we show that optimizing each formulation by local search corresponds to optimizing a continuation. The optimized policy is thus the mirror policy of an unknown original policy. We show the existence of the corresponding continuation and original policy and discuss their closed form. This analysis provides a novel interpretation of the state-of-the-art algorithms for direct policy optimization. We discuss the role of stochastic policies in light of this interpretation in Section 3.5.2.

3.5.1 *Gaussian Policies and Regularization*

The policy gradient literature has mainly focused on optimizing two problem formulations by local search – typically with stochastic gradient ascent and (approximate) trust-region methods. First, the vast majority of works focuses on optimizing the expected return of Gaussian policies (Andrychowicz et al., 2020; Duan et al., 2016). Second, in many formulations this objective function is extended by adding a bonus to the entropy of the optimized policy (Haarnoja, Zhou, Hartikainen, et al., 2019; Williams and Peng, 1991). We show that when optimizing a policy according to these formulations, there exists an (unknown) de-

terministic original policy and a continuation under which the optimized policy is a mirror policy. Provided with the local-search algorithm from the policy gradient method, we conclude that optimizing both formulations is equivalent to implicitly optimizing a deterministic policy by continuation.

First, we remind that under Property 3.4.3, for any affine deterministic policy μ_θ, there exists an affine Gaussian mirror policy $\pi_\theta^{GP'}$ as defined by equation (3.11). In Property 3.5.1, the converse of Property 3.4.3 is stated, which answers to the question: *under which conditions a Gaussian policy is the mirror policy of an (unknown) deterministic policy.* For this converse statement to be true, the transformation between covariance functions in Property 3.4.3 must be surjective, which is guaranteed if $d_\mathcal{A} \leq d_\Theta$ and $\nabla_\theta \mu_\theta(s)$ is full rank. The first assumption is always met in practice and the second is met when no action is a deterministic function of the others.

Property 3.5.1. *Let $\pi_\theta^{GP'}$ be an affine Gaussian policy with mean function μ_θ, and with covariance function $\Sigma'_\theta = \Sigma'$ constant with respect to the parameters of the policy (i.e., a function depending solely on the state). If $d_\mathcal{A} \leq d_\Theta$ and if $\nabla_\theta \mu_\theta(s)$ is full rank, then, there exists a continuation, with covariance Λ proportional to Σ', for which $\pi_\theta^{GP'}$ is a mirror policy of the original policy μ_θ.*

Entropy regularization ensures that the variance of the policy remains sufficiently large during the optimization process.[2] Similar objectives are pursued with maximum entropy reinforcement learning (Haarnoja, Zhou, Hartikainen, et al., 2019) or with (approximate) trust-region methods where the trust-region constraint is dualized (Schulman, Levine, et al., 2015; Schulman, Wolski, et al., 2017). Let us consider an affine Gaussian original policy π_θ^{GP} with constant covariance $\Sigma_\theta = \Sigma$. Under Property 3.4.2, there exists another affine Gaussian policy $\pi_\theta^{GP'}$ that is a mirror policy of π_θ^{GP}. This mirror policy has the same mean function and a covariance function bounded from below by $C_\theta = C$. Property 3.5.2 provides the converse and answers to the question: *under which conditions a Gaussian policy with sufficiently large covariance is the mirror policy of an (unknown and Gaussian) policy.* Similar to the previous property, this is guaranteed when $d_\mathcal{A} \leq d_\Theta$ and $\nabla_\theta \mu_\theta(s)$ is full rank.

Property 3.5.2. *Let $\pi_\theta^{GP'}$ be an affine Gaussian policy with mean function μ_θ, and with covariance function $\Sigma'_\theta = \Sigma' \succeq C$ constant with respect to the parameters of the policy (i.e., a function depending solely on the state) and bounded from bellow by C. If $d_\mathcal{A} \leq d_\Theta$ and if $\nabla_\theta \mu_\theta(s)$ is full rank, then, there exists a continuation, with covariance Λ proportional to C, for which $\pi_\theta^{GP'}$ is a mirror policy of an original Gaussian policy π_θ^{GP} with the same mean function μ_θ and with constant covariance function $\Sigma \preceq \Sigma'$.*

The two previous properties indicate that a Gaussian policy is guaranteed to be a mirror policy of another policy, Gaussian or deterministic, under some assumptions. If we furthermore guarantee that the continuation covariance decreases during the optimization, policy

2 Formally, for two matrices A and B, we have that $A \succeq B \Rightarrow |A| \geq |B|$ (Siotani, 1967). As the entropy of a Gaussian policy is a concave function of the determinant of the covariance matrix, a bounded covariance matrix implies a bounded entropy. The entropy-regularization learning objective can therefore be interpreted as the Lagrangian relaxation of the latter entropy-bounded optimization problem.

gradient algorithms optimizing affine Gaussian policies can be interpreted as algorithms optimizing an original policy by continuation.

Let us consider two cases, each corresponding to a problem formulation, where we optimize by policy gradient an affine Gaussian policy $\pi_\theta^{GP'}$ with covariance function constant with respect to the parameters of the policy. First, we consider the case where its covariance matrix decreases during the optimization through a manual scheduling. In this context, under property 3.5.1, there exists an original deterministic policy and the covariance of the continuation decreases through the optimization, such that the policy gradient algorithm optimizes this policy by continuation. Second, we consider the case where the entropy is regularized with a decreasing regularization term (e.g., by scheduling the Lagrange multiplier). Then, as entropy regularization can be seen as a constraint on the covariance of the policy, under property 3.5.2, there exists an original Gaussian policy and the covariance of the continuation decreases through the optimization, such that the policy gradient algorithm optimizes this stochastic policy by continuation. Finally, as stated previously and shown in Theorem 2 in Appendix 3.A, optimizing the expected return of the mirror policy of another mirror policy is equivalent to optimizing a continuation of the original policy. Therefore, policy gradient algorithms that optimize affine Gaussian policies with both discounted covariance and decreasing regularization by local search can also be interpreted as algorithms optimizing the mean function (i.e., a deterministic policy) of this policy by continuation.

We now illustrate how policy gradient algorithms implicitly optimize by continuation. We take as example an environment in which a car moves in a valley and must reach its lowest point (positioned in x_{target}) to maximize the expected sum of rewards gathered by the agent, see Appendix 3.B. We assume we want to find the best K-controller, i.e., a deterministic policy $\mu_\theta(x) = \theta \times (x - x_{target})$, where x is the position of the car. Directly optimizing such a policy is in practice subject to converging to a local extremum, as explain hereafter. We thus consider the Gaussian policy $\pi_\theta^{GP}(a|x) = \mathcal{N}(a|\mu_\theta(x), \sigma')$, where $\mu_\theta(x)$ and σ' are the mean and variance of the policy, respectively. This policy is a mirror policy of the deterministic policy μ_θ under a continuation of variance $\lambda = \sigma'/(x - x_{target})^2$, see Property 3.4.3. As can be seen in Figure 3.1, for each value of σ', the expected return of the mirror policy equals the smoothed expected return of the original deterministic policy μ_θ. Consequently, optimizing by policy gradient the Gaussian policy is equivalent to optimizing the deterministic policy by continuation. For a well-chosen sequence of σ', with a fixed scheduling or with adequate entropy regularization, the successive solutions found by local search will escape the basin of attraction of the suboptimal parameter for any initial parameter of the local search – whereas optimizing the deterministic policy directly would provide suboptimal solutions.

In this section, we have established an equivalence between the optimization of some policies by policy gradient and the optimization of an underlying policy by continuation. It opens up new questions about the hypothesis space of the (mirror) policy to consider in practice in order to exploit the properties of continuations at best. These considerations are

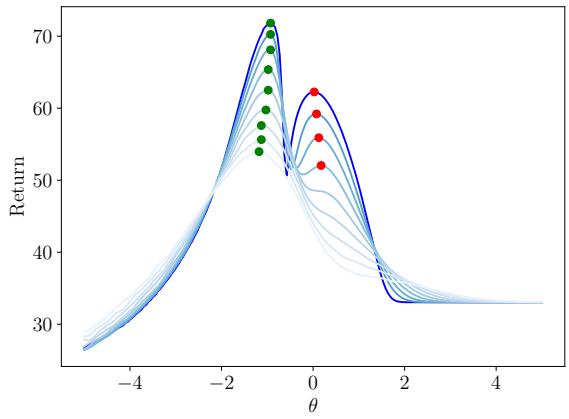


Figure 3.1: Illustration of the expected return of the policies $\mathcal{N}(a|\mu_\theta(x), \sigma')$, where $\mu_\theta(x) = \theta \times (x - x_{target})$, for different σ' values. The darker the curve, the smaller σ', and the darkest one is the expected return of the deterministic policy μ_θ. The green dots represent the global maxima and the red dots the local maxima. For some sufficiently large value for σ', the expected return of the policy has a single extremum. For a well chosen schedule of decreasing σ', a local-search algorithm will track the sequence of global extrema and converge towards the optimal deterministic policy.

made in the next section. We finally recall that a central assumption in the previous results is the affinity of policies. Seemingly restrictive, such policies allow to optimally control complex environments in practice (Rajeswaran et al., 2017) and give first-order results for non-affine policies.

3.5.2 *Continuations for Interpreting Stochastic Policies*

In practice, we know that optimizing stochastic policies tends to converge to a final policy with low variance and better performance than if we had directly optimized a deterministic policy. Practitioners often justify this observation by the need to explore through a stochastic policy. Nevertheless, to our knowledge, this concept inherited from bandit theory is not well defined for direct policy optimization. The previous analysis establishes an equivalence between optimizing stochastic policies with policy gradient algorithms and optimizing deterministic policies by continuation. Furthermore, as explained in Section 3.3.2, the continuation equation (3.7) consists in smoothing the expected return of this deterministic policy through the continuation distribution. Local optima tend to be removed when the variance of the continuation is sufficiently large. Optimizing stochastic policies and regularizing the entropy, as in most state-of-the-art policy gradient algorithms, is therefore expected to avoid local extrema before converging towards policies with small variance. We thus provide a theoretical motivation for the performance reached by algorithms applying exploration as understood in direct policy optimization.

The relationships between optimization by continuation and policy gradient in Section 3.5.1 have been established relying on Property 3.4.2 and Property 3.4.3. They assume continuations where the covariance matrix depends only on the current state and not on the whole observed history. In the general case, Property 3.4.4 allows one to extend these re-

sults by performing an analysis similar to Section 3.5.1. To be more specific, let us assume an affine Gaussian policy $\pi_\theta^{GP'}$, where the mean μ_θ is a function of the state and where the covariance $\Sigma_\theta = \Sigma$ is a function of the history and is constant with respect to θ. Under this assumption, if $d_\mathcal{A} \leq d_\Theta$ and $\nabla_\theta \mu_\theta(s)$ is full rank, the expected return of the policy $\pi_\theta^{GP'}$ is equal to a (unknown) continuation of the mean function μ_θ (i.e., a deterministic policy). Furthermore, optimizing the Gaussian policy by policy gradient while discounting the covariance can be interpreted as optimizing the deterministic policy μ_θ by continuation. In practice, this result suggests to optimize history-dependent policies by policy gradient to take advantage of the most general regularization of the objective function through implicit continuation. A similar observation was recently made by Mutti, De Santi, and Restelli (2022) who argued that history-dependent policies are required when more complex regularizations are involved. In parallel, Patil et al. (2022) also discussed the potential advantage of using history-depend parameterized policies for approximating correctly optimal policies using little representation power.

Finally, a last point has been left open in the previous discussions, namely the update of the covariance matrix of the mirror policies. The latter is defined through the covariance of the continuation. Therefore, the covariance must decrease through the optimization and must be chosen to avoid local optima. One direction to investigate in order to select a variance that removes local extrema is to update the parameters of the policy by following a combination of two directions: the functional gradient of the optimized policy's expected return with respect to the policy mean and the functional gradient of another measure (to be defined) with respect to the policy variance. An example of heuristic measure for smoothness might be the entropy of the actions and/or states encountered in histories. This strategy obviously does not follow the classical approach when optimizing stochastic policies where the covariance is adapted by the policy gradient algorithm to locally maximize the expected return and the exact procedure for updating the variance will require future studies. However the empirical inefficiency of this classical approach has already been highlighted in previous works that improved the performance of policy gradient algorithms by exploring alternative learning objective functions (Houthooft et al., 2018; Papini et al., 2020).

3.6 CONCLUSION

In this work, we have studied the problem formulation, i.e., policy parameterization and reward-shaping strategy, when solving direct policy optimization problems. More particularly, we established connections between formulations of state-of-the-art policy gradient algorithms and the optimization by continuation framework (Allgower and Georg, 1980). We have shown that algorithms optimizing stochastic policies and regularizing the entropy inherit the properties of optimization by continuation and are thus less subject to converging towards local optima. In addition, the role of the variance of the policies is reinterpreted in this framework: it is a parameter of the optimization procedure to adapt in order to avoid local extrema. Additionally, to inherit the properties of generic continu-

ations, it may be beneficial to consider variances that are functions of the history of states and actions observed at each time step.

Our study leaves several questions open. Firstly, our results rely on several assumptions that may not hold in practice. Specifically, it is unclear how our findings can be generalized to non-affine policies and alternative to Gaussian policies. Nonetheless, our results can be extended in cases where we can obtain an analytic expression for the mirror policy outlined in Theorem 1. While finding such an expression may be challenging in general, we can easily extend our conclusions to non-affine policies by considering the first-order approximation. Additionally, our study is focused on Gaussian policies, which are commonly used in continuous state-action spaces. However, for discrete action spaces, a natural choice of policy is a Bernoulli distribution over the actions (or a categorical distribution for more than one action). If the state space is also discrete, this distribution may be parameterize by a table providing the success probability of the Bernoulli distribution for each state. In the case of a Beta continuation distribution, a mirror policy can be derived where actions follow a Beta-binomial distribution in each state, a result known in Bayesian inference as the Beta distribution is a conjugate distribution of the binomial distribution (Bishop and Nasrabadi, 2006). An analysis of this mirror policy would allow us to draw conclusions equivalent to those of the continuous case studied in this chapter. Secondly, the study focused on entropy regularization of the policy only. Recent works have underlined the benefits of other regularization strategies that enforce the spread of other distributions as the state visiting frequency or the marginal state probability (Guo et al., 2021; Hazan, Kakade, et al., 2019; Mutti, De Santi, and Restelli, 2022). Future research is also needed to better understand the effect of these regularizations on the optimization procedure.

Finally, we give a new interpretation for the variance of policies that suggests it shall be updated to avoid local extrema rather than to maximize the expected return locally. A first strategy for updating the variance is proposed in Section 3.5.2, which opens the door to further research and new algorithm development.

APPENDIX

3.A THEORETICAL RESULTS ON MIRROR POLICIES

THEOREM 1. *For any original history-dependent policy $\eta_\theta \in \mathcal{E}$ parameterized with the vector $\theta \in \mathbb{R}^{d_\Theta}$ and for any continuation distribution q and covariance function Λ, there exists a mirror history-dependent policy $\eta'_\theta \in \mathcal{E}$ of the original policy η_θ that writes as:*

$$\eta'_\theta(a|h) = \mathop{\mathbb{E}}_{\theta' \sim q(\cdot|\theta,\Lambda(h))} \left[\eta_{\theta'}(a|h)\right] . \tag{3.13}$$

PROOF. The continuation f^q_Λ is defined in equation (3.7) as the expectation of the function (3.6) over the distribution (3.7) such that:

$$f^q_\Lambda(\theta) = \mathop{\mathbb{E}}_{\substack{s_0 \sim p_0(\cdot) \\ \theta_t \sim q(\cdot|\theta,\Lambda(h_t)) \\ a_t \sim \eta_{\theta_t}(\cdot|h_t) \\ s_{t+1} \sim p(\cdot|s_t,a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)\right] \tag{3.14}$$

$$= \int \left(p(s_0) \prod_{t=0}^{\infty} \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta,\Lambda(h_t)) p(s_{t+1}|s_t,a_t)\right) \left(\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)\right) ds_0 da_0 d\theta_0 \dots . \tag{3.15}$$

For the sake of simplifying the notations, let $h = (s_0, a_0, s_1, a_1, \dots) \in H$ be a history and let $R(h)$ be the discounted sum of rewards computed from this history. Let us reorder the terms of the integral and change the order of integration such that:

$$f^q_\Lambda(\theta) = \int \left(p(s_0) \prod_{t=0}^{\infty} \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta,\Lambda(h_t)) p(s_{t+1}|s_t,a_t)\right) R(h)\, dh d\theta_0 \dots \tag{3.16}$$

$$= \int \left(\prod_{t=0}^{\infty} \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta,\Lambda(h_t))\right) \left(p(s_0) \prod_{t=0}^{\infty} p(s_{t+1}|s_t,a_t)\right) R(h)\, dh d\theta_0 \dots \tag{3.17}$$

$$= \int \left(\int \prod_{t=0}^{\infty} \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta,\Lambda(h_t)) d\theta_0 \dots\right) \left(p(s_0) \prod_{t=0}^{\infty} p(s_{t+1}|s_t,a_t)\right) R(h)\, dh . \tag{3.18}$$

In the inner integral over the parameters, each term of the product depends solely on the parameter at a single time step such that the integral of the product simplifies to the product of the integrals as follows:

$$f^q_\Lambda(\theta) = \int \left(\prod_{t=0}^{\infty} \int \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta, \Lambda(h_t))\, d\theta_t\right) \left(p_0(s_0) \prod_{t=0}^{\infty} p(s_{t+1}|s_t, a_t)\right) R(h)\, dh \tag{3.19}$$

$$= \int \left(\prod_{t=0}^{\infty} \eta'_\theta(a_t|h_t)\right) \left(p_0(s_0) \prod_{t=0}^{\infty} p(s_{t+1}|s_t, a_t)\right) R(h)\, dh\,. \tag{3.20}$$

By definition, the latter equation is equal to the expected return $J(\eta'_\theta)$ of the policy η'_θ for any parameter vector θ. Therefore, η'_θ is a mirror policy of the original policy η_θ under the process q and covariance matrix Λ.

□

THEOREM 2. *Let q be a continuation distribution and let Λ be a covariance function as defined in Section 3.3.2. In addition, let η_θ, η'_θ and η''_θ be three parameterized history-dependent policies such that:*

$$\eta'_\theta(a|h) = \int \eta_{\theta'}(a|h) q(\theta'|\theta, \Lambda(h))\, d\theta' \tag{3.21}$$

$$\eta''_\theta(a|h) = \int \eta'_{\theta''}(a|h) q(\theta''|\theta, \Lambda(h))\, d\theta''\,. \tag{3.22}$$

Then, η'_θ is a mirror policy of the original policy η_θ and η''_θ is a mirror policy of the original policy η'_θ under continuation distribution q and covariance function Λ. In addition, there exists a continuation for which η''_θ is a mirror policy of the original policy η_θ.

PROOF. First, η'_θ is a mirror policy of the original policy η_θ and η''_θ is a mirror policy of the original policy η'_θ under continuation distribution q and covariance function Λ, see Theorem 1. Then, let us substitute equation (3.21) in equation (3.22):

$$\eta''_\theta(a|h) = \int \eta'_{\theta''}(a|h) q(\theta''|\theta, \Lambda(h))\, d\theta'' \tag{3.23}$$

$$= \int \left(\int \eta_{\theta'}(a|h) q(\theta'|\theta'', \Lambda(h))\, d\theta'\right) q(\theta''|\theta, \Lambda(h))\, d\theta'' \tag{3.24}$$

$$= \int \eta_{\theta'}(a|h) \left(\int q(\theta'|\theta'', \Lambda(h)) q(\theta''|\theta, \Lambda(h))\, d\theta''\right) d\theta'\,. \tag{3.25}$$

We thus have that:

$$\eta''_\theta(a|h) = \int \eta_{\theta'}(a|h) p_\theta(\theta'|h)\, d\theta' \tag{3.26}$$

$$p_\theta(\theta'|h) = \int q(\theta'|\theta'', \Lambda(h)) q(\theta''|\theta, \Lambda(h))\, d\theta''\,. \tag{3.27}$$

The distribution p_θ is a continuation distribution with a spread depending on the history h through the covariance function Λ. By Theorem 1, η''_θ is a mirror policy of the original policy η_θ.

□

PROPERTY 3.4.1. *Let the original policy $\pi_\theta \in \Pi$ be a Markov policy and let the covariance function depend solely on the last state in the history. Then, there exists a mirror Markov policy $\pi'_\theta \in \Pi$.*

PROOF. By hypotheses, the covariance matrix only depends on the last state s_t of the history h_t, therefore:

$$q(\theta_t|\theta, \Lambda(h_t)) = q(\theta_t|\theta, \Lambda(s_t))\,. \tag{3.28}$$

In addition, the original policy π_θ is a Markov policy, therefore :

$$\eta_\theta(a_t|h_t) = \pi_\theta(a_t|s_t)\,. \tag{3.29}$$

The closed form of the mirror policy, provided by equation (3.13), can thus be simplified as:

$$\eta'_\theta(a_t|h_t) = \int \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta, \Lambda(h_t))\, d\theta_t \tag{3.30}$$

$$= \int \pi_{\theta_t}(a_t|s_t) q(\theta_t|\theta, \Lambda(s_t))\, d\theta_t\,. \tag{3.31}$$

The previous equation is independent of h_t knowing s_t, there thus exists a Markov mirror policy $\pi'_\theta \in \Pi$ respecting Theorem 1 such that:

$$\eta'_\theta(a_t|h_t) = \pi'_\theta(a_t|s_t)\,. \tag{3.32}$$

□

PROPERTY 3.4.2. *Let the original policy $\pi_\theta^{GP} \in \Pi$ be a Gaussian policy as defined in equation (3.2) with affine function approximators. Let the covariance function depend solely on the last state in the history and let the distribution q be a Gaussian distribution. Then, there exists a mirror Markov policy $\pi'_\theta \in \Pi$ such that for all states $s \in \mathcal{S}$, it converges towards a Gaussian policy in the limit as the affine coefficients of the covariance matrix $\Sigma_\theta(s)$ approaches zero ($\|\nabla_\theta \Sigma_\theta(s)\| \to 0$):*

$$\pi'_\theta(a|s) \to \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(s))\,, \tag{3.33}$$

where $\Sigma'_\theta(s) = C_\theta(s) + \Sigma_\theta(s)$ and $C_\theta(s) = \nabla_\theta \mu_\theta(s)^T\, \Lambda(s)\, \nabla_\theta \mu_\theta(s)$.

Proof. First, the existence of a Markov mirror policy results from Property 3.4.1 and is provided by equation (3.32):

$$\pi'_\theta(a_t|s_t) = \int \pi_{\theta_t}(a_t|s_t) q(\theta_t|\theta, \Lambda(s_t))\, d\theta_t\,. \tag{3.34}$$

In addition, π_{θ_t} and q are Gaussian distributions by hypotheses:

$$\pi_{\theta_t}(a_t|s_t) = \mathcal{N}(a_t|\mu_{\theta_t}(s_t), \Sigma_{\theta_t}(s)) \tag{3.35}$$
$$q(\theta_t|\theta, \Lambda(s_t)) = \mathcal{N}(\theta_t|\theta, \Lambda(s_t))\,, \tag{3.36}$$

where $\mu_{\theta_t}(s_t)$ and $\Sigma_{\theta_t}(s_t)$ are affine functions of θ_t. Therefore, these functions can be written as follows:

$$\mu_{\theta_t}(s_t) = (\nabla_{\theta_t}\mu_{\theta_t}(s_t))\,\theta_t + \mu'(s_t) \tag{3.37}$$
$$\Sigma_{\theta_t}(s_t) = (\nabla_{\theta_t}\Sigma_{\theta_t}(s_t))\,\theta_t + \Sigma'(s_t)\,. \tag{3.38}$$

For any state s_t, in the limit as affine coefficients of the covariance approaches zero, the covariance is such that:

$$\lim_{\|\nabla_{\theta_t}\Sigma_{\theta_t}(s_t)\|\to 0} \Sigma_{\theta_t}(s_t) = \Sigma'(s_t)\,. \tag{3.39}$$

In this limit, equation (3.34) consists in marginalizing a conditional linear Gaussian transition model with a Gaussian prior and is such that (Bishop and Nasrabadi, 2006):

$$\lim_{\|\nabla_\theta\Sigma_\theta(s_t)\|\to 0} \pi'_\theta(a_t|s_t) = \mathcal{N}\left(a_t|\,(\nabla_\theta\mu_\theta(s_t))\,\theta + \mu'(s_t), (\nabla_\theta\mu_\theta(s_t))^T\,\Lambda(s_t)\,(\nabla_\theta\mu_\theta(s_t)) + \Sigma'(s_t)\right) \tag{3.40}$$
$$= \mathcal{N}\left(a_t|\mu_\theta(s_t), (\nabla_\theta\mu_\theta(s_t))^T\,\Lambda(s_t)\,(\nabla_\theta\mu_\theta(s_t)) + \Sigma_\theta(s_t)\right)\,. \tag{3.41}$$

□

PROPERTY 3.4.3. *Let the original policy $\mu_\theta \in M$ be an affine deterministic policy. Let the covariance function depend solely on the last state in the history and let the distribution q be a Gaussian distribution. Then, the Markov policy $\pi_\theta^{GP\prime} \in \Pi$ is a mirror policy:*

$$\pi_\theta^{GP\prime}(a|s) = \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(s))\,, \tag{3.42}$$

where $\Sigma'_\theta(s) = \nabla_\theta\mu_\theta(s)^T\,\Lambda(s)\,\nabla_\theta\mu_\theta(s)$.

Proof. The statement results from the particularization of Property 3.4.2 to the case of deterministic policies. Let $\pi_\theta^{GP} \in \Pi$ be an affine Gaussian policy with constant covariance matrix for any state $\Sigma_\theta(s_t) = C$. In that case, we have by Property 3.4.2 that $\pi'_\theta \in \Pi$ is a mirror policy as follows:

$$\pi'_\theta(a_t|s_t) = \mathcal{N}\left(a_t|\mu_\theta(s_t), (\nabla_\theta\mu_\theta(s_t))^T\,\Lambda(s_t)\,(\nabla_\theta\mu_\theta(s_t)) + \Sigma_\theta(s_t)\right) \tag{3.43}$$
$$= \mathcal{N}\left(a_t|\mu_\theta(s_t), (\nabla_\theta\mu_\theta(s_t))^T\,\Lambda(s_t)\,(\nabla_\theta\mu_\theta(s_t)) + C\right)\,. \tag{3.44}$$

Taking the limit of π_θ^{GP} as the constant covariance matrix C approaches zero, we get that the original policy from Property 3.4.2 converges to the one of Property 3.4.3, namely the

deterministic policy μ_θ. This implies that the policy $\pi_\theta^{GP\prime} \in \Pi$ provided by the limit of the mirror policy from Property 3.4.2, see equation (3.44), is a mirror policy of the original policy μ_θ from Property 3.4.3:

$$\pi_\theta^{GP\prime}(a_t|s_t) = \lim_{C\to 0} \pi'_\theta(a_t|s_t) = \mathcal{N}\left(a_t|\mu_\theta(s_t), \left(\nabla_\theta \mu_\theta(s_t)\right)^T \Lambda(s_t) \left(\nabla_\theta \mu_\theta(s_t)\right)\right) . \tag{3.45}$$

□

PROPERTY 3.4.4. *Let the original policy $\mu_\theta \in M$ be an affine deterministic policy. Let the distribution q be a Gaussian distribution. Then, the policy $\eta'_\theta \in \mathcal{E}$ is a mirror policy:*

$$\eta'_\theta(a|h) = \mathcal{N}(a|\mu_\theta(s), \Sigma'_\theta(h)) , \tag{3.46}$$

where $\Sigma'_\theta(h) = \nabla_\theta \mu_\theta(s)^T \Lambda(h) \nabla_\theta \mu_\theta(s)$.

Proof. The policy μ_{θ_t} is an affine function of the parameter vector θ_t and can thus be written as follows:

$$\mu_{\theta_t}(s_t) = \left(\nabla_{\theta_t} \mu_{\theta_t}(s_t)\right) \theta_t + \mu'(s_t) . \tag{3.47}$$

In addition, the samples drawn from the process q are distributed according to a Gaussian distribution:

$$q(\theta_t|\theta, \Lambda(h_t)) = \mathcal{N}(\theta_t|\theta, \Lambda(h_t)) . \tag{3.48}$$

The closed form of the density of the mirror policy, provided by equation (3.13), is thus simplified as:

$$\eta'_\theta(a_t|h_t) = \int \eta_{\theta_t}(a_t|h_t) q(\theta_t|\theta, \Lambda(h_t))\, d\theta_t \tag{3.49}$$

$$= \int \eta_{\theta_t}(a_t|h_t) \mathcal{N}(\theta_t|\theta, \Lambda(h_t))\, d\theta_t , \tag{3.50}$$

where η_{θ_t} is the policy where each action respecting equation (3.47) has a probability one. The policy is a degenerated Gaussian distribution (Rao, 1973), it provides a dirac measure to each state, and its (generalized) density function may be approached as follows:

$$\eta_{\theta_t}(a_t|h_t) = \lim_{\|\Sigma\|\to 0} \mathcal{N}(a_t|\left(\nabla_{\theta_t} \mu_{\theta_t}(s_t)\right) \theta_t + \mu'(s_t), \Sigma) . \tag{3.51}$$

By substitution, we therefore get that the mirror policy η'_θ writes as follow:

$$\eta'_\theta(a_t|h_t) = \int \eta_{\theta_t}(a_t|h_t) \mathcal{N}(\theta_t|\theta, \Lambda(h_t))\, d\theta_t \tag{3.52}$$

$$= \int \lim_{\|\Sigma\|\to 0} \mathcal{N}\left(a_t|\left(\nabla_{\theta_t} \mu_{\theta_t}(s_t)\right) \theta_t + \mu'(s_t), \Sigma\right) \mathcal{N}\left(\theta_t|\theta, \Lambda(h_t)\right)\, d\theta_t . \tag{3.53}$$

The product of the Gaussian prior over parameters and the linear Gaussian transition model of the actions provides a joint Gaussian distribution of actions and parameters (Bishop and Nasrabadi, 2006), which is degenerated but has a density for the (marginal) Gaussian distribution of actions (Rao, 1973). The density of the mirror policy η'_θ can thus be computed taking the limit of the marginalization:

$$\eta'_\theta(a_t|h_t) = \lim_{\|\Sigma\|\to 0} \int \mathcal{N}\left(a_t|\left(\nabla_{\theta_t}\mu_{\theta_t}(s_t)\right)\theta_t + \mu'(s_t), \Sigma\right)\mathcal{N}\left(\theta_t|\theta, \Lambda(h_t)\right)\, d\theta_t \tag{3.54}$$

$$= \lim_{\|\Sigma\|\to 0} \mathcal{N}\left(a_t|\left(\nabla_\theta\mu_\theta(s_t)\right)\theta + \mu'(s_t), \left(\nabla_\theta\mu_\theta(s_t)\right)^T \Lambda(h_t)\left(\nabla_\theta\mu_\theta(s_t)\right) + \Sigma\right) \tag{3.55}$$

$$= \lim_{\|\Sigma\|\to 0} \mathcal{N}\left(a_t|\mu_\theta(s_t), \left(\nabla_\theta\mu_\theta(s_t)\right)^T \Lambda(h_t)\left(\nabla_\theta\mu_\theta(s_t)\right) + \Sigma\right) \tag{3.56}$$

$$= \mathcal{N}\left(a_t|\mu_\theta(s_t), \left(\nabla_\theta\mu_\theta(s_t)\right)^T \Lambda(h_t)\left(\nabla_\theta\mu_\theta(s_t)\right)\right)\,. \tag{3.57}$$

We note that this result can be obtained without working on degenerated Gaussian distributions. The policy is an affine function of the parameters, which follow a Gaussian distribution, the marginal distribution of actions is thus also a Gaussian distribution of the form of equation (3.57). This distribution is furthermore the one of a mirror policy, see Theorem 1.

□

3.B DESCRIPTION OF THE CAR ENVIRONMENT

In this section, we formalize the reinforcement learning environment that models the movement of a car in a valley with two floors separated by a peak, as depicted in Figure 3.B.1. The car always starts at the topmost floor and receives rewards proportionally to its depth in the valley. An optimal agent drives the car from the initial position to the lowest floor in the valley by passing the peak. In the following, we describe each element composing the environment.

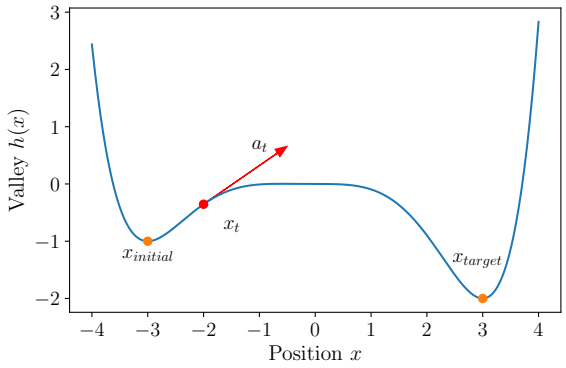


Figure 3.B.1: Valley in which the car moves.

STATE SPACE. The state $s_t \in \mathbb{R}^2$ of the environment is composed of two scalar values, namely the position $x_t \in \mathbb{R}$ of the pointmass representing the car and its tangent speed $v_t \in \mathbb{R}$.

ACTION SPACE. At each time step, the agent controls the force applied on the car through the actions $a_t \in \mathbb{R}$ it executes.

INITIAL POSITION. The car always starts at the topmost floor $x_{initial} = -3$ in the valley at rest. The initial state distribution thus provides a probability one to the state

$$x_0 = x_{initial} \tag{3.58}$$
$$v_0 = 0\,. \tag{3.59}$$

TRANSITION DISTRIBUTION. The continuous motion of the car in the valley is derived for Newton's formula. The valley's analytical description is provided by the function h, the car's mass is denoted by $m = 0.5$, gravitational acceleration by $g = 9.81$, and damping factor by $e = 0.65$. The position x and speed v of the car follow the subsequent continuous-time dynamics as a function of the force a:

$$\dot{x} = v \tag{3.60}$$
$$\dot{v} = \frac{a}{m(1+h'(x)^2)} - \frac{gh'(x)}{1+h'(x)^2} + \frac{v^2 h'(x)h''(x)}{1+h'(x)^2} - ev^2\,. \tag{3.61}$$

The position and force are furthermore bounded to intervals as part of the dynamics such that

$$x \in [x_m, x_M] = [-4, 5] \tag{3.62}$$
$$a \in [a_m, a_M] = [-10, 10]\,. \tag{3.63}$$

Clamped force values are therefore used in equation (3.61). Similarly, the position is clamped in equation (3.60).

In discrete time, the state s_{t+1} is computed through Euler integration of the continuous-time dynamics, considering an initial position given by the current state s_t. The force a remains constant during a discretization time $\Delta = 0.1$ and is equal to the action a_t, with an additive noise drawn from $\mathcal{N}(\cdot|0, 1)$ and clamped before integration.

REWARD FUNCTION. The rewards correspond to the depth of the valley at the current position. The reward function thus solely depends on the position

$$R(s_t, a_t) = -h(x_t)\,. \tag{3.64}$$

DISCOUNT FACTOR. The discount factor equals $\gamma = 0.99$ and the horizon is curtailed to $T = 100$ in each numerical computation.

4

BEHIND THE MYTH OF EXPLORATION IN POLICY GRADIENTS

Prologue

This chapter is based on the following publication: *Bolland, A., Lambrechts, G., & Ernst, D. (2024). Behind the Myth of Exploration in Policy Gradients. arXiv preprint arXiv:2402.00162.*

Policy gradient algorithms are effective reinforcement learning methods for solving control problems. To compute near-optimal policies, it is essential in practice to include exploration terms in the learning objective. Although the effectiveness of these terms is usually justified by an intrinsic need to explore environments, we propose a novel analysis with the lens of numerical optimization. Two criteria are introduced on the learning objective and two others on its stochastic gradient estimates, and are afterwards used to discuss the quality of the policy after optimization. The analysis sheds the light on two separate effects of exploration techniques. First, they make it possible to smooth the learning objective and to eliminate local optima while preserving the global maximum. Second, they modify the gradient estimates, increasing the probability that the stochastic parameter updates eventually provide an optimal policy. These effects are illustrated empirically on exploration strategies based on entropy bonuses, highlighting their limitations and opening avenues for future works in the design and analysis of such strategies.

4.1 INTRODUCTION

Many practical problems require making sequential decisions in environments. Reinforcement learning (RL) is a framework for solving such decision-making problems that has been successful on complex tasks, including playing games (Mnih, Kavukcuoglu, et al., 2015; Silver, Schrittwieser, et al., 2017), operating power systems (Aittahar et al., 2024), controlling robots (Kalashnikov et al., 2018), or interacting with electricity markets (Boukas et al., 2021).

Reinforcement learning algorithms interact with an environment to gather information about this environment, which in turn enables to compute and follow an optimal policy. This creates a trade-off between exploration and exploitation. In short, in order to even-

tually compute a good policy, it is necessary to obtain additional information about the environment by taking actions that are likely not optimal. In algorithms where the trade-off is explicit, exploration is well-understood and has been the subject of many works (Azar et al., 2017; Dann et al., 2017; Neu and Pike-Burke, 2020). In policy gradient algorithms, one can most often not distinguish exploration from exploitation. Nevertheless, a main theoretical requirement to converge towards globally (or even locally) optimal solutions is that policies remain sufficiently stochastic during the learning procedure (Agarwal et al., 2020; Bedi, Chakraborty, et al., 2022; Bhandari and Russo, 2019; Bhatt et al., 2019; Junzi Zhang et al., 2021). Interestingly, neither softmax nor Gaussian policies guarantee enough stochasticity for ensuring (fast) convergence (Bedi, Chakraborty, et al., 2022; Mei, Dai, et al., 2021; Mei, Xiao, Dai, et al., 2020). This requirement of stochasticity in policy gradient algorithms is often abusively called exploration and understood as the need to infinitely sample all states and actions.

Practitioners have tried to meet the theoretical requirement of sufficient randomness of policies in policy gradient via reward-shaping strategies, whereby a learning objective that promotes or hinders behaviors by providing reward bonuses for some states and actions is optimized as a surrogate to the expected return of the policy. These bonuses typically promote actions that reduce the uncertainty of the agent about its environment (Burda et al., 2018; D. Pathak et al., 2017; T. Zhang, Xu, et al., 2021), or that maximize the entropy of states and/or actions (M. Bellemare et al., 2016; Guo et al., 2021; Haarnoja, Zhou, Hartikainen, et al., 2019; Hazan, Kakade, et al., 2019; Islam et al., 2019; Lee et al., 2019; Williams and Peng, 1991; T. Zhang, Rashidinejad, et al., 2021). Optimizing a surrogate objective is particularly effective for solving tasks with complex dynamics and reward functions, or with sparse rewards.

The differences between theory and practical implementations of exploration has led to common folklore seeking to provide intuition for the efficiency of policy gradient methods. This work is part of the research line that studies the maximization of practical surrogate learning objective functions from a mathematical optimization perspective. Close to our work, studies of the learning objective with entropy regularization (an exploration-based reward shaping technique where the entropy of the policy is added in the learning objective) were conducted. It includes the study by Ahmed et al. (2019) concluding that it helps to provide smooth learning objective functions. The same exploration strategy was reinterpreted as a robust optimization method by Husain et al. (2021) and equivalently as a two-player game by Brekelmans et al. (2022). Bolland, Louppe, and Ernst (2023) furthermore argued that optimizing an entropy regularized objective is equivalent to optimizing the expected return of another policy with larger variance. More general studies on the learning dynamics have focused on the influence of baselines in policy gradient (Chung et al., 2021), and reward-shaping strategies that do not modify the learning objective, called potential based (Forbes et al., 2024; Harutyunyan et al., 2015; A. Y. Ng et al., 1999; Wiewiora et al., 2003). All these studies are too restrictive and the literature lacks unified explanations and interpretations about exploration in policy gradients.

Before delving into our contributions, we recall that the convergence of stochastic ascent methods is driven by the objective function and how the ascent directions are estimated. First, the objective function shall be (pseudo) concave to find its global maximum (Bottou, 1998). Second, the convergence rate is influenced by the distribution of the stochastic ascent estimates (Ajalloeian and Stich, 2020; J. Chen and Luss, 2018). In this chapter, we rigorously study policy gradient methods with exploration-based reward shaping through the lens of these two optimization theory aspects. To that end, we first introduce two new criteria that relate the expected return of a policy to the learning objective with exploration bonuses, and their respective optima. Second, we introduce two additional criteria on the distribution of the gradient estimates of the learning objective and their likelihood of providing directions in which the learning objective and the expected return increase. Importantly, these criteria are general to any reward-shaping strategy, and highlight the importance of reward shaping that modify the optimal control behavior, in opposition to the literature on potential-based reward shaping. The influence of some exploration bonuses are illustrated and discussed in the light of these four criteria. In practice, finding good exploration strategies is problem specific and we thus introduce a general framework for the study and interpretation of exploration in policy gradient methods instead of trying to find the best exploration method for a given task.

The chapter is organized as follows. In Section 4.2, we provide the background about policy gradients and exploration. Section 4.3 focuses on the effect of exploration on the learning objective while Section 4.4 is dedicated to the effect on the gradient estimates used in the policy gradient algorithms.

4.2 PRELIMINARIES AND THEORETICAL BACKGROUND

Let us consider problems in which an agent interacts with a Markov decision process (MDP) as formalized in Chapter 2. We remind that policy gradient algorithms (locally) optimize a parameterized policy π_θ to find the optimal parameter θ^* for which the expected return of the policy $J(\pi_{\theta^*})$ is maximized. In practice, it is common to maximize the surrogate learning objective L where the expected discounted sum of rewards is extended by K additional reward terms R_i^{int}, called intrinsic motivation terms

$$L(\theta) = \frac{1}{1-\gamma} \mathop{\mathbb{E}}_{\substack{s\sim d^{\pi_\theta,\gamma}(\cdot)\\ a\sim\pi_\theta(\cdot|s)}} \left[R(s,a) + \sum_{i=0}^{K-1} \lambda_i R_i^{int}(s,a) \right] = J(\pi_\theta) + J^{int}(\pi_\theta)\,, \tag{4.1}$$

where λ_i are non-negative weights for each intrinsic reward and where $J^{int}(\pi_\theta)$ is the expected intrinsic return of the policy. The parameter maximizing the learning objective is denoted by $\theta^\dagger$, which we distinguish from the optimal policy parameter θ^*.

In this chapter, we focus on two of these bonuses

$$R^s(s,a) = -\log d^{\pi_\theta,\gamma}(\phi(s)) \tag{4.2}$$

$$R^a(s,a) = -\log \pi_\theta(a|s)\,, \tag{4.3}$$

where $\phi(s)$ is a feature built from the state s. The corresponding expected intrinsic returns are maximized for policies that visit uniformly every feature, and for policies with uniformly distributed actions in each state, respectively. Note that these rewards require to estimate the distribution over the states and/or actions. Furthermore, they implicitly depend on the policy parameter θ. The second reward is usually referred to as entropy regularization.

In this chapter, we consider on-policy policy gradient algorithms, which were among others reviewed by Duan et al., 2016 and Andrychowicz et al., 2020. These algorithms optimize differentiable parameterized policies with gradient-based local optimization. They iteratively approximate an ascent direction $\hat{d}$ relying on histories sampled from the policy in the MDP and update the parameters in the ascent direction, or in a combination of the previous ascent directions (Hinton et al., 2012; Kingma and Ba, 2014). For the sake of simplicity and without loss of generality, we consider that the ascent direction $\hat{d}$ is composed of the sum of an estimate of the gradient of the expected return $\hat{g} \approx \nabla_\theta J(\pi_\theta)$ and an estimate of the gradient of the expected intrinsic return $\hat{i} \approx \nabla_\theta J^{int}(\pi_\theta)$. In practice, the first is usually unbiased while the second is computed neglecting some partial derivatives of θ and is thus biased, typically neglecting the influence of the policy on the intrinsic reward.

4.3 STUDY OF THE LEARNING OBJECTIVE

In this section, we study the influence of the exploration terms on the learning objective defined in equation (4.1). We define two criteria under which the learning objective can be globally optimized by ascent methods, and such that the solution is close to an optimal policy. We then graphically illustrate how exploration modifies the learning objective to remove local extrema.

4.3.1 *Policy Gradient Learning Objective*

Policy gradient algorithms using exploration maximize the learning objective function L, as defined in equation (4.1). We introduce two criteria related to this learning objective for studying the performance of the policy gradient algorithm. First, we say that a learning objective L is ϵ-coherent when its global maximum is in an ϵ-neighborhood of the expected return of an optimal policy. Second, we call learning objectives that have a unique maximum and no other stationary point pseudoconcave.

Coherence criterion. *A learning objective L is ϵ-coherent if, and only if,*

$$J(\pi_{\theta^*}) - J(\pi_{\theta^\dagger}) \leq \epsilon \,, \tag{4.4}$$

where $\theta^* \in argmax_\theta J(\pi_\theta)$ and where $\theta^\dagger \in argmax_\theta L(\theta)$.

Pseudoconcavity criterion. *A learning objective L is pseudoconcave if, and only if,*

$$\exists! \, \theta^\dagger : \nabla L(\theta^\dagger) = 0 \wedge L(\theta^\dagger) = \max_\theta L(\theta) \,. \tag{4.5}$$

If the pseudoconcavity criterion is respected, there is a single optimum, and it is thus possible to globally optimize the learning objective function by (stochastic) gradient ascent (Bottou, 2010)[1]. If the learning objective is furthermore ϵ-coherent, the latter solution is also a near-optimal policy, where ϵ is the bound on the suboptimality of its expected return.

Let us finally remind a theorem from A. Y. Ng et al. (1999).

Consistency Theorem. *The learning objective L is ϵ-coherent, with $\epsilon = 0$, in any MDP with state space $\mathcal{S}$, action space $\mathcal{A}$ and factor γ, if, and only if, $J(\pi_\theta) = L(\theta)$ for all θ. The intrinsic rewards are furthermore potential based.*

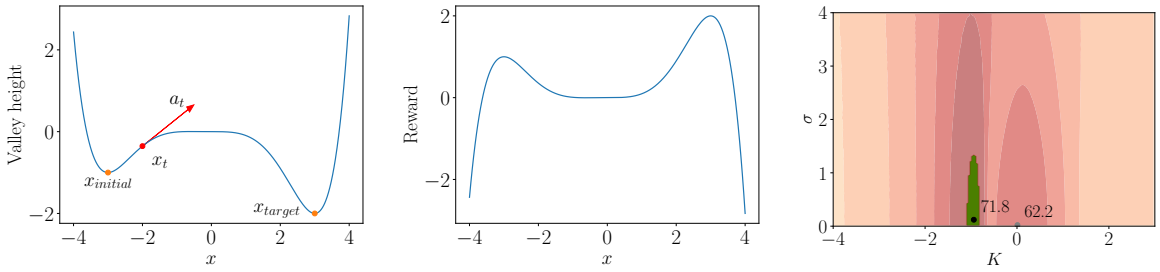
This theorem states that there is no MDP-agnostic exploration method that guarantees consistency with ϵ equal to zero and that modifies the objective function. This type of exploration is only possible with potential-based reward shaping (A. Y. Ng et al., 1999). In conclusion, if the expected return is not pseudoconcave, there is a trade-off between the two criteria, which can not be resolved by potential-based exploration.

4.3.2 *Illustration of the Effect of Exploration on the Learning Objective*

Exploration is of paramount importance in environments with complex dynamics and reward functions, where many locally optimal policies may exist (Lee et al., 2019; H. Liu and Abbeel, 2021; T. Zhang, Rashidinejad, et al., 2021). In the following, we first define such an environment and a policy parameterization that will serve as an example to illustrate the effect of exploration on the optimization process. For the sake of the analysis, we then represent the learning objectives associated to different exploration strategies, and depict their global and local optima. Learning objectives with a single optimum respect the pseudoconcavity criterion. In addition, we represent the neighborhood Ω of the optimal policy parameters, such that any learning objective with its global maximum within this region is coherent for a given ϵ. In light of the coherence and the pseudoconcavity criteria, we finally elaborate on the policy parameter computed by stochastic gradient ascent algorithms.

1 For the sake of keeping discussions simple, the definition of pseudoconcavity is simplified (Mangasarian, 1975), and additional assumptions on the stochastic gradient estimates are neglected.

We consider the environment illustrated in Figure 4.3.1a where a car moves in a valley (Bolland, Louppe, and Ernst, 2023). We denote by x and v the position and speed of the car, both composing its state $s = (x, v)$. The valley contains two separate low points, positioned in $x_{initial} = -3$ and $x_{target} = 3$, separated by a peak. The car starts at rest $v_0 = 0$ at the highest low point $x_0 = x_{initial}$ and receives rewards proportional to the depth of the valley at its current position. The reward function is provided in Figure 4.3.1b. We consider a policy $\pi_{K,\sigma}(a|s) = \mathcal{N}(a|\mu_K(s), \sigma)$, namely a normally disturbed proportional controller with $\mu_K(s) = K \times (x - x_{target})$, parameterized by the vector $\theta = (K, \sigma)$. Figure 4.3.1c illustrates the contour map of the expected return of the policy as a function of the parameters K and σ. The optimal parameters are represented by black dots and correspond to policies that drive the car to pass the peak and reach the lowest valley point in x_{target}. The green area represents the set of parameters $\Omega = \{\theta' | \max_\theta J(\pi_\theta) - J(\pi_{\theta'}) \leq \epsilon\}$ for $\epsilon = 1$.



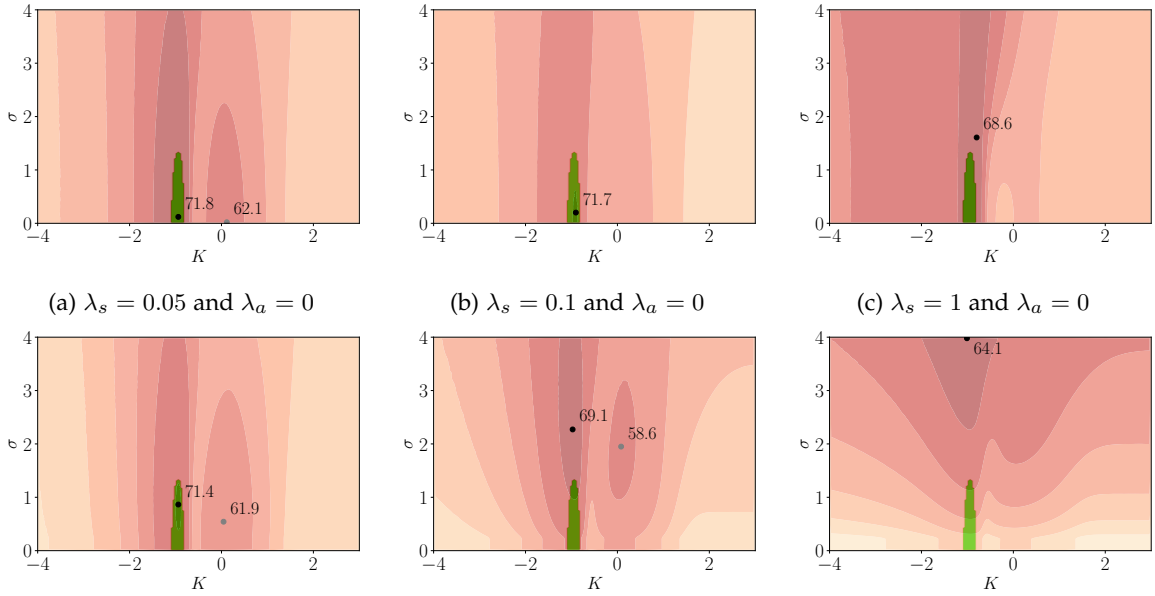
(a) Hill environment. (b) Reward function R. (c) Return $J(\pi_{K,\sigma})$.

Figure 4.3.1: Illustration of the *hill environment* in Figure 4.3.1a and its reward function in Figure 4.3.1b. In Figure 4.3.1c, the expected return of the policy $\pi_{K,\sigma}$ with the global and local maximum represented in black and grey, together with their respective expected return values.

Figure 4.3.2 illustrates the learning objective with the intrinsic rewards $R^s(s, a) = -\log d^{\pi_{K,\sigma},\gamma}(\phi(s))$, from equation (4.2), and $R^a(s, a) = -\log \pi_{K,\sigma}(a|s)$, from equation (4.3), for different values of the corresponding weights λ_s and λ_a. Here, the feature is the position in the valley $\phi(s) = x$. First, we observe that for weights approaching zero, the parameter $\theta^\dagger$ maximizing the learning objective, represented by a black dot, corresponds to a policy with a high expected return. More precisely, it is in the green set Ω such that ϵ-coherence is guaranteed for the value $\epsilon = 1$. Larger weights require larger values of ϵ for guaranteeing the ϵ-coherence criterion. Nevertheless, when increasing the weights, we observe that the learning objective eventually becomes pseudoconcave. There appears to be a trade-off between the two criteria. In Figure 4.3.2b, we observe that in this environment, there is a learning objective that respects the pseudoconcavity criterion and the ϵ-coherence criterion for $\epsilon = 1$. Indeed, there is a single global maximum in Figure 4.3.2b represented by a black dot that is furthermore part of the set Ω.

Shaping the reward function with an exploration strategy based on the state-visitation entropy appears to be a good solution for optimizing the policy. However, a notable drawback is that the reward depends on the policy and its (gradient) computation requires to estimate a complex probability measure. In this example, the intrinsic reward function itself was estimated by Monte-Carlo sampling for every parameter, which would not scale

for complex problems and requires approximations and costly evaluation strategies (Islam et al., 2019). In Appendix 4.A we present an alternative problem-dependent intrinsic reward, independent of the policy parameters and thus simple to compute efficiently, that still respects the pseudoconcavity and ϵ-coherence criteria, and in Appendix 4.B we extend the study to more complex environments from the MiniGrid library (Chevalier-Boisvert et al., 2023) where the policy is a deep neural network and the state-visitation probability is approximated.



(a) $\lambda_s = 0.05$ and $\lambda_a = 0$ (b) $\lambda_s = 0.1$ and $\lambda_a = 0$ (c) $\lambda_s = 1$ and $\lambda_a = 0$

(d) $\lambda_s = 0$ and $\lambda_a = 0.01$ (e) $\lambda_s = 0$ and $\lambda_a = 0.1$ (f) $\lambda_s = 0$ and $\lambda_a = 0.5$

Figure 4.3.2: Contour map of (scaled) learning objective functions for different values of λ_s and λ_a. The darker the map, the larger the learning objective value. The green area represents the set $\Omega = \{\theta' | \max_\theta J(\pi_\theta) - J(\pi_{\theta'}) \leq \epsilon = 1\}$, such that when the parameter maximizing the learning objective is part of Ω, then the learning objective function is ϵ-coherent with $\epsilon = 1$. The black dot is the parameter $\theta^\dagger$ globally maximizing the learning objective and the grey dot is the local (non-global) maximum of the learning objective if it exists. Both are labeled with the expected return values of the corresponding policies.

The observations suggest that well-chosen exploration strategies can lead to learning objective functions that satisfy the two criteria defined in the previous section, thereby guaranteeing that policies suboptimal by at most ϵ can be computed by local optimization. When designing exploration strategies, it is essential to keep in mind that we modify the learning objective for the algorithms to converge to optimal policy parameters, which can be achieved when both criteria are respected. While strategies such as enforcing entropy can be effective in some environments, they are only heuristic strategies and not to be relied upon exclusively. Furthermore, as illustrated, both criteria may be subject to a trade-off. In more complex environments, an efficient exploration strategy may require to balance both criteria, e.g., through a schedule on the learning objective weights.

4.4 STUDY OF THE ASCENT DIRECTION DISTRIBUTION

Optimizing pseudoconcave functions with stochastic ascent methods are guaranteed to converge (at a certain rate) under assumptions on the distribution of the gradient estimates at hand (Ajalloeian and Stich, 2020; Bottou, 2010; J. Chen and Luss, 2018). In this section, we study the influence of the exploration terms on this distribution in the context of policy gradients. More precisely, we study the probability of improving the learning objective and the expected return with stochastic ascent steps. Intuitively, they shall be sufficiently large for the algorithm to be efficient. We formalize this intuition and illustrate how exploration strategies can increase these probabilities, leading to more efficient algorithms.

4.4.1 *Policy Gradient Estimated Ascent Direction*

In general, gradient ascent algorithms update parameters in a direction $\hat{d}$ in order to locally improve an objective function f. The quality of these algorithms can therefore be studied (for a small step size $\alpha \to 0$) through the random variable representing the quantity by which the objective increases for each θ

$$X = f(\theta + \alpha\hat{d}) - f(\theta) = \alpha \left\langle \hat{d}, \nabla_\theta f(\theta) \right\rangle , \tag{4.6}$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product. This variable depends on the random event $\hat{d}$ estimated by Monte-Carlo simulations in practice.

The (asymptotic) convergence of a gradient ascent algorithm is usually studied by bounding the expectation of X. Such bounds depend, among others, on the expectation of $\hat{d}$, which equals $\nabla_\theta f(\theta)$ when unbiased, and depend on the variance of $\hat{d}$, which deteriorates the expected convergence rate. Finding rates depending on the algorithm is an active field of research. In parallel, a slightly more general problem is to quantify if the expectation of X is driven by rare events. Intuitively, an algorithm with unbiased gradient estimates has positive expected improvements $\mathbb{E}[X] > 0$, and should theoretically converge, but may be inefficient in practice if positive events $X > 0$ rarely occur. We illustrate in the next section that this phenomenon makes reinforcement learning in sparse-reward environments particularly hard. To the best of our knowledge, no existing result fits to the study of policy gradients. We therefore introduce two new criteria on the probability of improvement $P(X > 0)$, which we empirically validate afterwards.

First, we define an exploration strategy as δ-efficient if, and only if, following the ascent direction $\hat{d} \approx \nabla_\theta L(\theta)$ has a probability at least δ to increasing the learning objective $L(\theta)$ almost everywhere. Second, an exploration strategy is δ-attractive if, and only if, there exists a neighborhood of $\theta^\dagger$ containing the parameter θ^{int} maximizing the expected intrinsic return J^{int}, where the probability of increasing the expected return by following $\hat{d}$ is almost everywhere at least equal to δ. Note that each probability measure and random variable is a function of θ, which we do not explicitly write for the sake of keeping notations simple.

Efficiency criterion. *An exploration strategy is δ-efficient if, and only if,*

$$\forall^{\infty}\theta : \mathbb{P}(D > 0) \geq \delta\,, \tag{4.7}$$

where $D = \langle \hat{d}, \nabla_\theta L(\theta) \rangle$.

Attraction criterion. *An exploration strategy is δ-attractive if, and only if,*

$$\exists B(\theta^\dagger) : \theta^{int} \in B(\theta^\dagger)\,, \tag{4.8}$$

such that

$$\forall^{\infty}\theta \in B(\theta^\dagger) : \mathbb{P}(G > 0) \geq \delta\,, \tag{4.9}$$

where $\theta^{int} = argmax_\theta J^{int}(\pi_\theta)$, $B(\theta^\dagger)$ *is a ball centered in* $\theta^\dagger$, *and* $G = \langle \hat{d}, \nabla_\theta J(\pi_\theta) \rangle$.

The efficiency criterion quantifies how often a stochastic gradient ascent step improves the learning objective. The larger, the better the learning objective and its stochastic ascent direction approximations. The rationale behind the attraction criterion is that in many exploration strategies, the intrinsic reward is dense, and it is then presumably easy to optimize the expected intrinsic return in the sense that $\mathbb{P}(\langle \hat{i}, \nabla_\theta J^{int}(\pi_\theta) \rangle > 0)$ is large. It implies that it is easy to locally improve the learning objective by (solely) increasing the value of the intrinsic motivation terms. It furthermore implies that policy gradient algorithms may be subject to converging towards θ^{int} rather than $\theta^\dagger$ when $\mathbb{P}(\langle \hat{d}, \nabla_\theta J(\pi_\theta) \rangle > 0)$ is small. If the criterion is respected for large δ, the latter is less likely to happen as policy gradients will eventually tend to improve the expected return of the policy if the parameter approaches θ^{int} and enters the ball $B(\theta^\dagger)$; eventually converging towards $\theta^\dagger$.

These two new criteria on $\hat{d}$ are independent of the previous ones on L, which only captured the quality of the deterministic learning objective functions. In the particular cases where the learning objectives L are ϵ-coherent, for $\epsilon = 0$, and pseudoconcave, e.g., with potential-based intrinsic rewards, only the distribution of estimates $\hat{d}$ can explain why some algorithms succeed and others fail. Finally, the value of δ in the new criteria we introduce can be related to the variance of the estimate $\hat{d}$ under some assumptions, e.g., with Cantelli's concentration inequalities.

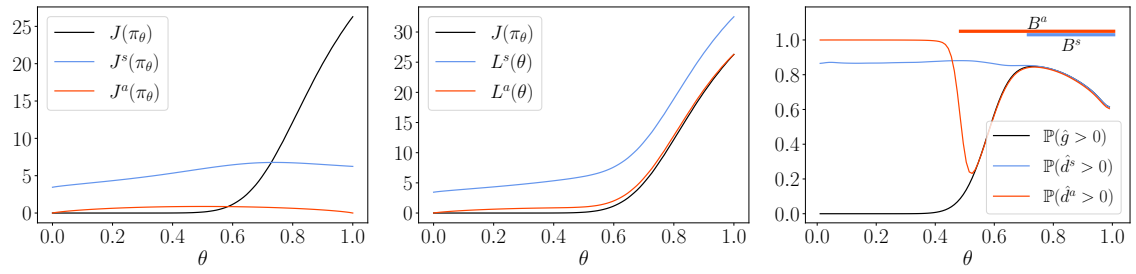
4.4.2 *Illustration of the Effect of Exploration on the Estimated Ascent Direction*

Exploration is usually promoted and tested for problems where the reward function is sparse, typically in maze-environments (Guo et al., 2021; Islam et al., 2019; H. Liu and Abbeel, 2021). In this section, we first introduce a new maze-environment with sparse rewards where we illustrate the influence of exploration on the gradient estimates of the learning objective. To this end, we present two learning objective functions and elaborate on the influence of exploration on the performance of policy gradient algorithms in the light of the efficiency and attraction criteria.

Let us consider a maze-environment consisting of a horizontal corridor composed of $S \in \mathbb{N}$ tiles. The state of the environment is the index of the tile $s \in \{1, \dots, S\}$, and the actions consist in going left $a = -1$ or right $a = +1$. When an action is taken, the agent stays idle with probability $p = 0.7$, and moves with probability $1 - p = 0.3$ in the direction indicated by the action, then $s' = \min(S, \max(1, s + a))$. The agent starts in state $s = 1$ and the target state $s = S = 15$ is absorbing. Zero rewards are observed except when the agent reaches the target state where a reward $r = 100$ is observed. A discount factor of $\gamma = 0.99$ is considered. Finally, we study the policy going with probability θ to the right and probability $1 - \theta$ to the left, and with density

$$\pi_\theta(a|s) = \begin{cases} \theta & \text{if } a = 1 \\ 1 - \theta & \text{if } a = -1 \,. \end{cases} \tag{4.10}$$

The expected return $J(\pi_\theta)$ is represented in black in Figure 4.4.1a as a function of θ along with two expected intrinsic returns, $J^a(\pi_\theta)$ in orange and $J^s(\pi_\theta)$ in blue. The intrinsic reward $R^a(s, a) = -\log \pi_\theta(a|s)$, from equation (4.3), and the intrinsic reward $R^s(s, a) = -\log d^{\pi_\theta, \gamma}(s)$, from equation (4.2), are used respectively. In Figure 4.4.1b, we illustrate the expected return of the policy without exploration $J(\pi_\theta)$, along with two learning objective functions, $L^a(\theta)$ and $L^s(\theta)$, using as exploration strategies the intrinsic returns $J^a(\pi_\theta)$ and $J^s(\pi_\theta)$. We observe that the expected return is a pseudoconcave function with respect to θ and the optimal parameter is $\theta^* = 1$. In addition, the two learning objectives respect the ϵ-coherence criterion for $\epsilon = 0$, implying that $\theta^* = \theta^\dagger$, and respect the pseudoconcavity criterion. It is important to note that with regard to the discussion from Section 4.3, there is no interest in optimizing the learning objectives rather than directly optimizing the expected return, as the latter is already pseudoconcave. In the following we illustrate how choosing a correct exploration strategy still deeply influences the policy gradient algorithms when it comes to building gradient estimates.



(a) Return functions. (b) Learning objective functions. (c) Improvement probabilities.

Figure 4.4.1: Figure 4.4.1a represents the expected return of the policy along with two expected intrinsic return functions. In Figure 4.4.1b the expected return is also represented together with two learning objective functions, corresponding to the two expected intrinsic returns. Figure 4.4.1c illustrates the probability (estimated by Monte-Carlo) of positive stochastic gradient (derivative) estimates $J(\pi_\theta)$, $L^a(\theta)$, and $L^s(\theta)$. At the top of the figure, the intervals $B^a = [\theta^{int,a}, \theta^{\dagger,a}]$ and $B^s = [\theta^{int,s}, \theta^{\dagger,s}]$ are represented. These intervals represent the smallest balls containing the parameters maximizing the expected intrinsic return and the learning objective, for both exploration strategies.

Let us compute the estimate $\hat{g}$ and $\hat{d}$ relying on REINFORCE (Williams, 1992) by sampling 8 histories of length $T = 100$. In this particular environment, $\mathbb{P}(D > 0)$ equals $\mathbb{P}(G > 0)$, and equal the probability that the derivative is positive. We represent in Figure 4.4.1c this probability for the expected return and for both learning objectives. First, we see that the learning objectives are more efficient than the expected return, meaning they are δ-efficient for larger values of δ. Depending on the parameter value, the objective $L^a(\theta)$ or $L^s(\theta)$ is best in that regard. Second, concerning the attraction criterion, we represent at the top of Figure 4.4.1c the intervals $B^a = [\theta^{int,a}, \theta^{\dagger,a}]$ and $B^s = [\theta^{int,s}, \theta^{\dagger,s}]$. They correspond to the smallest balls containing the maximizers of the expected intrinsic return and of the learning objective. Let the minima of the orange and blue curves over these intervals be denoted by δ^a and δ^s. By definition of the attraction criterion, it is thus respected for any values of δ at most equal to δ^a and δ^s, for $L^a(\theta)$ and $L^s(\theta)$, respectively. All these observations can eventually be explained as the computation of $\hat{g}$ is always zero when the target is not sampled in the histories, which is highly likely for policies with small values of θ. Policy gradient algorithms relying on intrinsic exploration would compute optimal policies efficiently where naive optimization without exploration would fail or be sample inefficient.

We have empirically shown that a well-chosen exploration strategy in policy gradients may not only remove local extrema from the objective function, but may also increase the probability that stochastic ascent steps improve the objective function. Under the previous assumptions, this probability measures the efficiency of algorithms. Furthermore, among different learning objectives respecting the coherence and pseudoconcavity criteria, it is best to choose one that has high values for δ in both the efficiency and attraction criteria. In Appendix 4.A we use these criteria to study other reward-shaping strategies, and in Appendix 4.B we extend the study to more complex environments from the MiniGrid library (Chevalier-Boisvert et al., 2023) where the policy is a deep neural network.

The problem discussed in this section strongly relates to overfitting or generalization in reinforcement learning. In situations where the same state and action pairs are repeatedly sampled with high probability, the policy may appear optimal by neglecting the rewards observed in state and action pairs sampled with low probability. The gradient estimates will then be zero with high probability, and the gradient updates will not lead to policy improvements. In the previous example, gradient estimates computed from policies with a small parameter value θ wrongly indicate that a stationary point has been reached as they equal zero with high probability. We quantify this effect with a novel definition of local optimality. We define as locally optimal policies over a space with probability Δ the policies that maximize the reward on expectation over a set of states and actions observed

in a history with probability at least Δ. Formally, a policy π is locally optimal over a space with probability Δ if, and only if,

$$\exists\, \mathcal{E} \in \left\{ \mathcal{X} \middle| \int_{\mathcal{X}} d^{\pi,\gamma}(s)\pi(a|s)\, dads \geq \Delta \right\} : \\ \pi \in \underset{\pi'}{\operatorname{argmax}} \int_{\mathcal{E}} d^{\pi',\gamma}(s)\pi'(a|s)R(a,s)\, dads\,. \tag{4.11}$$

In the typical case of environments with sparse rewards, many policies observe with high probability state and action pairs with zero rewards and are locally optimal for large probabilities Δ. Typically, in the previous example, the joint set $\{1,\dots,S-2\}\times\{-1,1\}$ is a set of state and action pairs $\mathcal{E}$ that respects the definition equation (4.11) for large values Δ when θ is small. As we have shown, exploration mitigates the convergence of policy gradient algorithms towards these locally optimal policies. Note that assuming a non-zero reward is uniformly distributed over the state and action space, exploration policies with uniform probabilities over visited states and actions are the best choice for sampling non-zero rewards with high probability. It can thus also be considered as the best choice of exploration to reduce the probability that the stochastic gradient ascent steps do not increase the objective value. Such initial policy may be learned from the framework developed by Lee et al. (2019).

4.5 CONCLUSION

In conclusion, this research takes a step towards dispelling misunderstandings about exploration through the study of its effects on the performance of policy gradient algorithms. More particularly, we distinguished two effects exploration has on the optimization. First, it modifies the learning objective in order to remove local extrema. Second, it modifies the gradient estimates and increases the likelihood that the update steps lead to improved expected returns. These two phenomena were studied through four criteria that we introduced and illustrated.

These ideas apply to other direct policy optimization algorithms. Indeed, the four criteria do not assume any structure on the learning objective and can thus be straightforwardly applied to any objective function optimized by a direct policy search algorithm. In particular, for off-policy policy gradient, we may simply consider that the off-policy objective is itself a surrogate or that the gradients of the expected return are biased estimates based on past histories. Ideas introduced in this work also apply to other reinforcement learning techniques. Typically, for value-based RL with sparse-reward environments, convergence towards a value function outputting zero is expected with high probability. This is mostly due to the low probability of sampling non-zero rewards by Monte-Carlo. The discussions from Section 4.4 then apply, and a similar analysis can be performed.

Our framework opens the door for further theoretical analysis, and the potential development of new criteria. We believe that deriving practical conditions on the exploration

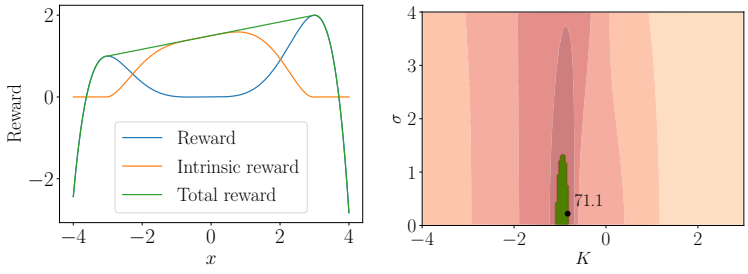
strategies, and the scheduling of the expected intrinsic return, for guaranteeing fast convergence should be the focus of attention. It could be achieved by bounding the policy improvement on expectation, which is nevertheless usually a hard task without strong assumptions. We furthermore believe that we provide a new lens on exploration necessary for interpreting and developing exploration strategies, in the sense of optimizing surrogate learning objective functions.

APPENDIX

4.A REWARD SHAPING AND EXPLORATION STRATEGIES

As discussed in the manuscript, exploration strategies are reward-shaping strategies where the intrinsic reward bonuses are, among others, dependent on the policy parameters. This dependency makes the shaping strategies adaptive but makes the computation of gradients and the study of the learning objectives more complex. In this section, we study handcrafted reward-shaping strategies that have pseudoconcave and dense reward functions in the hill and maze environments. We then illustrate that the same criteria can be used to study these expert-knowledge based shaped rewards.

For the hill environment from Section 4.3, we illustrate in Figure 4.A.1a an intrinsic reward bonus making the sum of rewards in equation (4.1) concave. The corresponding learning objective has a unique maximum, which is part of the set $\Omega = \{\theta' | \max_\theta J(\pi_\theta) - J(\pi_{\theta'}) \leq \epsilon\}$ with $\epsilon = 1$ and $\theta = (K, \sigma)$. It can be seen in Figure 4.A.1b where the global maximum in black is within the set Ω in green. Both, the ϵ-coherence and the pseudoconcavity criteria are thus respected for $\epsilon = 1$. Here, the intrinsic reward function is a simple function independent of the policy π_θ. Finding such an intrinsic reward may be complex for other environments but the example underlines that exploration and reward shaping are mostly equivalent and that designing reward functions that are concave may help converging towards optimal policies.

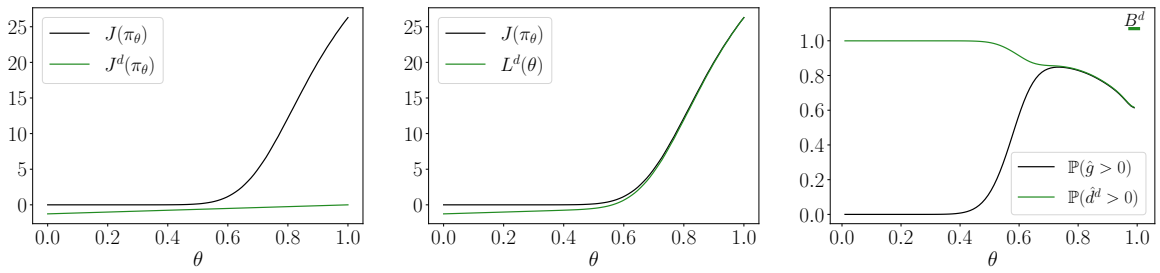


(a) Alternative reward function. (b) Learning objective.

Figure 4.A.1: In Figure 4.A.1a, an alternative intrinsic reward function ensuring that the sum of rewards is a pseudoconcave function. In Figure 4.A.1b, the contour function of the learning objective.

For the maze environment, the expected return $J(\pi_\theta)$ is represented in black in Figure 4.A.2a together with the expected intrinsic return $J^d(\pi_\theta)$ in green. The latter is the expected return of the dense handcrafted reward function $R^d(s, a) = (a - 1)/2$ penalizing actions moving away from the target. In Figure 4.A.2b, the corresponding learning objec-

tive function is shown. In the same experimental setting as in Section 4.4, we observe that the objective function is δ-efficient for higher values of δ compared to the already-discussed learning objectives. Furthermore, the attraction criterion is respected for any value of δ as the unique global maxima of the learning objective, expected intrinsic return, and expected return are all equals.



(a) Return. (b) Learning objectives. (c) Improvement probabilities.

Figure 4.A.2: In Figure 4.A.2a the expected return of the maze environment is represented together with the expected intrinsic return of a dense handcrafted reward function. Figure 4.A.2b represents the corresponding learning objective and Figure 4.A.2c the probability that the REINFORCE estimates are positive.

4.B MINIGRID EXPERIMENTS

In this section, we introduce complex environments and parameterize policies with neural networks. In this context, it is impractical to naively compute and represent the objective functions and probability distributions for the different criteria. Therefore, we only evaluate the criteria along parameter trajectories, and extend the previous experimental setting.

We consider seven environments from the MiniGrid suite of environments (Chevalier-Boisvert et al., 2023), among others, designed for evaluating exploration strategies. In these environments, an agent moves in a maze and aims to reach a target position. To do so, the agent may choose actions that consist of turning left, turning right, moving forward, or staying idle. We consider two reward settings: the dense setting and the sparse setting. In the first, rewards of -1 are received for every non-idle move, and a reward of 1000 is received upon reaching the target position. In the second setting, zero rewards are received everywhere, except upon reaching the target position, where a bonus of 1000 is provided. In the dense setting, due to the action penalization incurred when moving, a policy outputting the idle action with probability one is locally optimal and has a expected return equal to zero. This is not (necessarily) the case in the second setting. We consider a discount factor of $\gamma = 0.98$ and optimize a fully connected neural network taking as input the position pair and the orientation of the agent, and outputting a categorical distribution over actions. The network is composed of three hidden layers of 64 neurons with ReLU activation functions.

In the dense reward setting, we optimize policies by maximizing three learning objective functions: $J(\pi_\theta)$, $L^a(\theta)$, and $L^s(\theta)$, respectively with $\lambda_a = 0.5$ and $\lambda_s = 0.25$. For the

last objective, the state-visitation density estimator is a ten-component Gaussian mixture model maximizing the likelihood of the sampled batch. The optimization is performed using the Adam update rule (Kingma and Ba, 2014), with REINFORCE ascent directions computed over 32 histories of constant length $T = 100$, and with learning rate (step size) equal to 0.0005. The length T of the histories is chosen such that the realization value T from a geometric distribution with success probability parameter $1 - \gamma$ has at least a cumulative probability of 0.85. In this setting, we illustrate the quasiconcavity criterion and the ϵ-coherence criterion. In Figure 4.B.1, we provide the evolution of the expected return of the policies when optimizing the three objectives $J(\pi_\theta)$, $L^a(\theta)$, and $L^s(\theta)$ for the different environments. For the `MiniGrid-Empty-8x8-v0` and the `MiniGrid-FourRooms-v0` environments, optimizing the expected return results in high-performance policies that do not stay idle. The other objectives also manage to find high-performing policies, but with a lower expected return. This phenomenon (assuming that the global optimum of each objective is found) illustrates the ϵ-coherence criterion, where this ϵ value is the bound on the best policy that can be found when optimizing the learning objective. For the other environments, the policies resulting from the optimization of the expected return fall into local optima, namely ones where the policy chooses the idling action with probability one. When optimizing the learning objectives with exploration bonuses, the resulting policies no longer fall into the previous local optima. This result suggests that, along these parameter trajectories, the expected return $J(\pi_\theta)$ has a local optimum (or saddle point), in opposition to the learning objective functions $L^a(\theta)$ and $L^s(\theta)$. The latter illustrates the validity of the pseudoconcavity criterion in that region of the parameter space. For the learning objective $L^s(\theta)$, the ϵ-coherence criterion is respected for a small value of ϵ, and the resulting policy manages to reach the target position. For the learning objective $L^a(\theta)$, the resulting policy does not reach the target position. We nevertheless hypothesize that it is not a real local optimum of the learning objective but a local optimum in the sense of equation (4.11).

In the previous experiments with the dense setting, the local optima exist due to the negative rewards associated to idle-actions. If we consider the sparse setting, we could then assume that directly optimizing the expected return is sufficient to find high-performing policies. It is not always the case and it can be justified using the efficiency and attraction criteria. We use the same parameters as in the previous set of experiments and we provide the evolution of the expected return of the policies when optimizing the three objectives $J(\pi_\theta)$, $L^a(\theta)$, and $L^s(\theta)$, for the different environments in Figure 4.B.2. On the one hand, for most environments, whatever the learning objective, the resulting policy has a high expected return. Note that the ϵ-coherence can again be illustrated where the policies resulting from the optimization of the expected return perform better than the others. On the other hand, for the `MiniGrid-Empty-16x16-v0` and the `MiniGrid-SimpleCrossingS11N5-v0` environments, a high performing policy can only be found when optimizing the learning objective $L^s(\theta)$. We illustrate that these results can be justified by the efficiency and attraction criteria in Figure 4.B.3. For each parameters obtained during the stochastic ascent steps on the expected return, we first estimate the probability of improving both objective functions

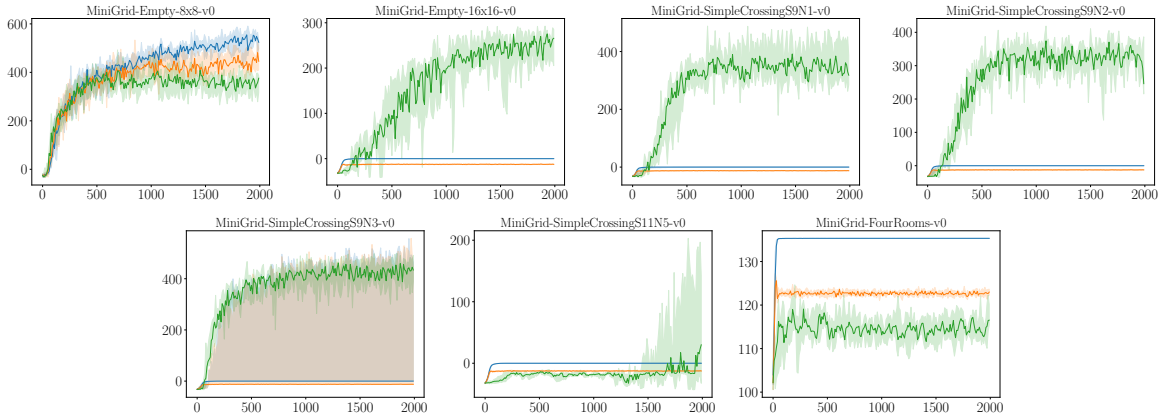


Figure 4.B.1: Evolution of the expected return of policies during optimization in the dense minigrid environments. In blue, the expected return $J(\pi_\theta)$ is optimized; in orange, the learning objective $L^a(\theta)$ is optimized; and in green, the learning objective $L^s(\theta)$ is optimized by performing Adam steps in REINFORCE directions. Note that the median, worst, and best cases over five runs are represented for the different curves. For the environments `MiniGrid-Empty-8x8-v0` and `MiniGrid-FourRooms-v0`, optimizing each objective results in a policy that does not stay idle. These policies are initialized outside of the basin of attraction of the local optimum of the expected return. The coherence criteria can be observed as optimizing the learning objective with intrinsic exploration bonuses results in suboptimal policies. For the other environments, optimizing the expected return directly leads to policies that always choose to stay idle, and are thus locally optimal. Optimizing the learning objective with exploration allows us to escape from these local optima, illustrating the quasiconcavity criterion. It can be noted that when optimizing the objective $L^a(\theta)$, the ϵ-coherence criterion is respected for a large value of ϵ, making the resulting policy worse than when optimizing the expected return directly. On the contrary, the objective $L^s(\theta)$ appears to have ϵ-coherence for a reasonable value of ϵ.

by stochastic gradient ascent. These probabilities are used to compare learning objectives in terms of the efficiency criterion; the larger, the better. Then, in order to illustrate the attraction criterion, we estimate the probability of improving the expected return and the learning objective, both by gradient ascent steps on the learning objective, for each parameters obtained during the stochastic ascent steps on the objective $L^a(\theta)$ and $L^s(\theta)$. As far as $L^a(\theta)$ is concerned, all probabilities remain small as result that the optimization procedure converges fast towards a stationary point where the target goal is observed with negligible probability. The efficiency and attraction criteria are respected for negligible probabilities δ, which is also a justification for the failure of converging towards good policies in the dense setting. For $L^s(\theta)$, on the contrary, and in both environments, the probability of improving the learning objective remains large for each parameter encountered when optimizing the expected return. The efficiency of the learning objective is much higher than that of the expected return in that part of the parameter space. Furthermore, the probability of improving the expected return when optimizing the learning objective, is small at the beginning and increases after some iterations. This indicates that once the policy has a sufficiently large expected intrinsic return, the attraction criterion is respected for a high value δ.

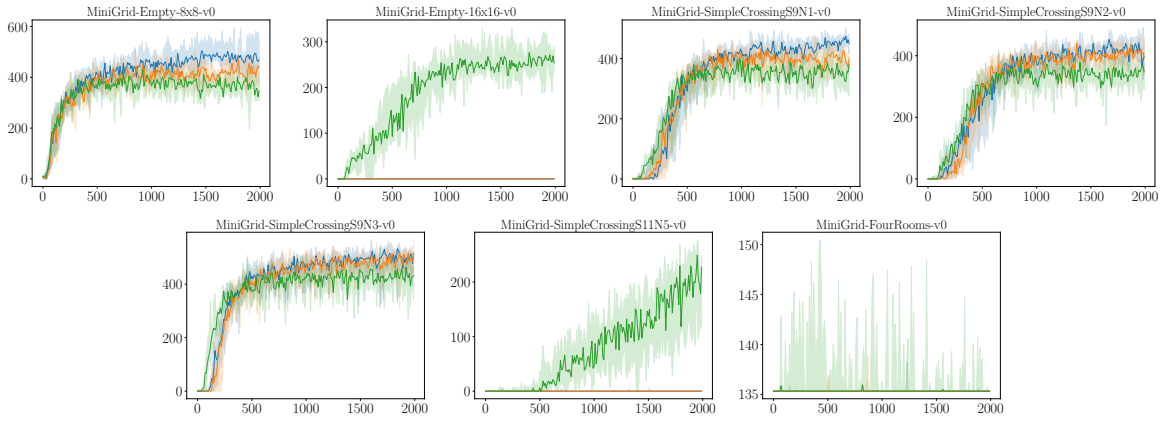


Figure 4.B.2: Evolution of the expected return of policies during optimization in the sparse minigrid environments. In blue, the expected return $J(\pi_\theta)$ is optimized; in orange, the learning objective $L^a(\theta)$ is optimized; and in green, the learning objective $L^s(\theta)$ is optimized performing Adam steps in REINFORCE directions. Note that the median, worst, and best cases over five runs are represented for the different curves. In most environments, a high-performing policy can be found by optimizing the expected return $J(\pi_\theta)$. This results from its quasiconcavity. The ϵ-coherence criterion can also be observed. However, for the `MiniGrid-Empty-16x16-v0` and the `MiniGrid-SimpleCrossingS11N5-v0` environments, a high-performing policy can only be found when optimizing the learning objective $L^s(\theta)$.

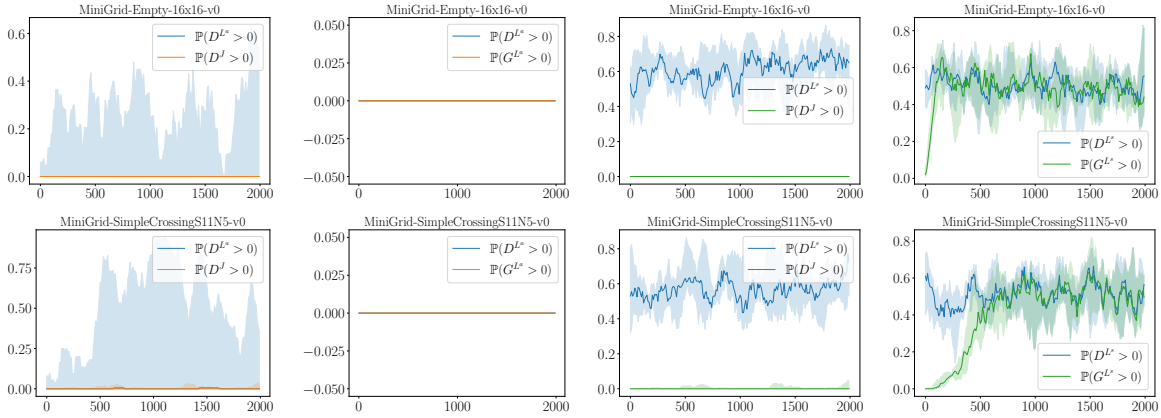


Figure 4.B.3: For the `MiniGrid-Empty-16x16-v0` and the `MiniGrid-SimpleCrossingS11N5-v0` environments, and for both learning objective functions $L^a(\theta)$ and $L^s(\theta)$, we first represent the estimated probability of improving the expected return $J(\pi_\theta)$ and the corresponding learning objective when following their REINFORCE gradient estimate. This value is estimated at each run of the optimization of the policy with learning objective $J(\pi_\theta)$. Second, we represent the estimated probability of improving the corresponding learning objective and the expected return $J(\pi_\theta)$ when following the REINFORCE gradient estimate of the learning objectives. These values are estimated at each run of the optimization of the policy with the learning objectives. The probabilities were estimated based on the frequencies of improving the objective functions by more than 0.2 when following 5 Adam ascent steps using REINFORCE update directions.

5

OFF-POLICY MAXIMUM ENTROPY RL WITH FUTURE STATE AND ACTION VISITATION MEASURES

Prologue

This chapter is based on the following publication: *Bolland, A., Lambrechts, G., & Ernst, D. (2024). Off-Policy Maximum Entropy RL with Future State and Action Visitation Measures. arXiv preprint arXiv:2412.06655.*

We introduce a new maximum entropy reinforcement learning framework based on the distribution of states and actions visited by a policy. More precisely, an intrinsic reward function is added to the reward function of the Markov decision process that shall be controlled. For each state and action, this intrinsic reward is the relative entropy of the discounted distribution of states and actions (or features from these states and actions) visited during the next time steps. We first prove that an optimal exploration policy, which maximizes the expected discounted sum of intrinsic rewards, is also a policy that maximizes a lower bound on the state-action value function of the decision process under some assumptions. We also prove that the visitation distribution used in the intrinsic reward definition is the fixed point of a contraction operator. Following, we describe how to adapt existing algorithms to learn this fixed point and compute the intrinsic rewards to enhance exploration. A new practical off-policy maximum entropy reinforcement learning algorithm is finally introduced. Empirically, exploration policies have good state-action space coverage, and high-performing control policies are computed efficiently.

5.1 INTRODUCTION

Many challenging tasks where an agent makes sequential decisions have been solved with reinforcement learning (RL). Examples range from playing games (Mnih, Kavukcuoglu, et al., 2015; Silver, Schrittwieser, et al., 2017), or controlling robots (Haarnoja, Pong, et al., 2018; Kalashnikov et al., 2018), to managing the energy systems and markets (Aittahar et al., 2024; Boukas et al., 2021). In practice, many RL algorithms are applied in combination with an exploration strategy to achieve high-performance control. Assuming the agent

takes actions in a Markov decision process (MDP), these exploration strategies usually consist in providing intrinsic reward bonuses to the agent for achieving certain behaviors. Typically, the bonus enforces taking actions that reduce the uncertainty about the environment (Burda et al., 2018; D. Pathak et al., 2017; T. Zhang, Xu, et al., 2021), or actions that enhances the variety of states and actions in trajectories (M. Bellemare et al., 2016; Guo et al., 2021; Haarnoja, Zhou, Hartikainen, et al., 2019; Lee et al., 2019; Williams and Peng, 1991). In many of the latter methods, the intrinsic reward function is the entropy of some distribution over the state-action space. Optimizing jointly the reward function of the MDP and the intrinsic reward function, in order to eventually obtain a high-performing policy, is called Maximum Entropy RL (MaxEntRL) and was shown effective in many problems.

The reward of the MDP was already extended with the entropy of the policy in early algorithms (Williams and Peng, 1991) and was only later called MaxEntRL (Toussaint, 2009; Ziebart et al., 2008). This particular reward regularization provides substantial improvements in robustness of the resulting policy (Brekelmans et al., 2022; Husain et al., 2021; Ziebart, 2010) and provides a learning objective function with good smoothness and concavity properties (Ahmed et al., 2019; Bolland, Louppe, and Ernst, 2023). Several commonly used algorithms can be named, like soft Q-learning (Haarnoja, Tang, et al., 2017; Schulman, X. Chen, and Abbeel, 2017) and soft actor-critic (Haarnoja, Zhou, Abbeel, and Levine, 2018; Haarnoja, Zhou, Hartikainen, et al., 2019). This MaxEntRL framework nevertheless only rewards the randomness of actions and neglects the influences of the policy on the visited states, which, in practice, may lead to inefficient exploration.

In order to enhance exploration, Hazan, Kakade, et al. (2019) were first to propose to intrinsically motivate agents to have a uniform discounted visitation measure over states. Several works have afterwards been developed to maximize the entropy of the discounted state visitation measure and the stationary state visitation measure. For discrete state and action spaces, optimal exploration policies, which maximize the entropy of these visitation measures, can be computed to near optimality with off-policy tabular model-based RL algorithms (Hazan, Kakade, et al., 2019; Mutti and Restelli, 2020; Tiapkin et al., 2023). For continuous state and action spaces, alternative methods rely on k nearest neighbors to estimate the density of the visitation measure of states (or features built from the states) and compute the intrinsic rewards, which can afterwards be optimized with any RL algorithm (H. Liu and Abbeel, 2021; Mutti, Pratissoli, and Restelli, 2021; Seo et al., 2021; Yarats et al., 2021). These methods require sampling new trajectories at each iteration, they are on-policy, and estimating the intrinsic reward function is computationally expensive. Some other methods rely on parametric density estimators to reduce the computational complexity and share information across learning steps (Guo et al., 2021; Islam et al., 2019; Lee et al., 2019; C. Zhang et al., 2021). The additional function approximator is typically learned on-policy by maximum likelihood estimation based on batches of truncated trajectories. Worth noticing methods have adapted this MaxEntRL framework to maximize entropy of states visited in single trajectories instead of on expectation over trajectories (Jain et al., 2024; Mutti, De Santi, and Restelli, 2022). When large and/or continuous state

and action spaces are involved, relying on parametric function approximators is likely the best choice. Nevertheless, existing algorithms are on-policy. They require sampling new trajectories from the environment at (nearly) every update of the policy, and can not be applied using a buffer of arbitrary transitions, in batch-mode RL, or in continuing tasks. Furthermore, learning the discounted visitation measure is more desirable than learning the stationary one, but may be challenging in practice due to the exponentially decreasing influence of the time step at which states are visited (Islam et al., 2019).

The main contribution of this chapter is to introduce a MaxEntRL framework relying on a new intrinsic reward function, for exploring effectively the state and action spaces, that also alleviates the previous limitations. In this new MaxEntRL framework, for each state and action, the intrinsic reward function is the relative entropy of the discounted distribution of states and actions (or features from these states and actions) visited during the next time steps. We first prove that a policy maximizing the expected discounted sum of these rewards is also one that maximizes a lower bound on the state-action value function of the MDP under some assumptions. In addition, we prove that the visitation distribution used in the intrinsic reward function definition is the fixed point of a contraction operator. Existing RL algorithms can integrate an additional learning step to approximate this fixed point off-policy, using N-step state-action transitions and bootstrapping the operator. It is then possible to approximate the intrinsic reward function and learn a policy maximizing the extended rewards with the adapted algorithm. We illustrate this methodology on off-policy actor-critic (Degris et al., 2012). The resulting MaxEntRL algorithm is off-policy, computes efficiently exploration policies with uniform discounted state visitation and high-performing control policies.

The visitation measure of future states and actions, which we use to extend the reward function in this article, has a well-established history in the development of RL algorithms. It was popularized by Janner et al. (2020), who learned the distribution of future states as a generalization of the successor features (Barreto et al., 2017). He demonstrated that this distribution allows to express the state-action value function by separating the influence of the dynamics and the reward function, and that it could be learned off-policy exploiting its recursive expression. Several algorithms have been proposed to learn this distribution, either by maximum likelihood estimation (Janner et al., 2020), by contrastive learning (Mazoure, Eysenbach, et al., 2023), or using diffusion models (Mazoure, Talbott, et al., 2023). These distributions of future states and actions have found applications in goal-based RL (Eysenbach, Salakhutdinov, and Levine, 2020; Eysenbach, T. Zhang, et al., 2022), in offline pre-training with expert examples (Mazoure, Bruce, et al., 2023), in model-based RL (Ma et al., 2023), or in planning (Eysenbach, Myers, et al., 2023). We are the first to integrate them into a MaxEntRL framework for enhancing exploration through learning.

The chapter is organized as follows. In Section 5.2, the problem of computing optimal policies is reminded and a general MaxEntRL framework is formulated. In Section 5.3, we introduce MaxEntRL with conditional state-action visitation probability and show how

policies can be computed in this framework. Finally, in Section 5.4 we present experimental results and conclude in Section 5.5.

5.2 PRELIMINARIES AND THEORETICAL BACKGROUND

Let us consider problems in which an agent interacts with a Markov decision process (MDP) as formalized in Chapter 2. In maximum entropy reinforcement learning (MaxEntRL) an optimal policy π^* is approximated by maximizing a surrogate objective function $L(\pi)$, where the reward function from the MDP is extended by an intrinsic reward function. The latter is the (relative) entropy of a conditional distribution. A general definition of the MaxEntRL objective function is

$$L(\pi) = \mathop{\mathbb{E}}_{\substack{s_0 \sim p_0(\cdot) \\ a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t) + \lambda R^{int}(s_t, a_t) \right) \right] , \tag{5.1}$$

where R^{int} is the intrinsic reward function. As discussed in Section 5.1, different MaxEntRL frameworks exist, each defining as intrinsic reward the entropy of some particular distribution. We propose a generic formulation for the intrinsic reward, which to the best of our knowledge encompasses most existing frameworks from the literature. Given a feature space $\mathcal{Z}$, a conditional distribution $q^\pi : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{Z})$, depending on the policy π, and a relative measure $q^* \in \mathcal{P}(\mathcal{Z})$, the intrinsic reward function is

$$\begin{aligned} R^{int}(s, a) &= -KL_z \left[q^\pi(z|s, a) \| q^*(z) \right] \\ &= \mathop{\mathbb{E}}_{z \sim q^\pi(\cdot|s,a)} \left[\log q^*(z) - \log q^\pi(z|s, a) \right] . \end{aligned} \tag{5.2}$$

Importantly, the intrinsic reward function is (implicitly) dependent on the policy π through the distribution q^π. We define an optimal exploration policy as a policy that maximizes the expected sum of discounted intrinsic rewards only. In practice, as soon as it is possible to generate samples from the distribution $q^\pi(z|s, a)$ and estimate their probabilities, the intrinsic reward equation (5.2) can be estimated by Monte Carlo, and used in any existing RL algorithm to extend the MDP reward function. Note that a policy maximizing $L(\pi)$ is generally not optimal, due to the potential gap between the optimum of the expected return $J(\pi)$ and the optimum of the learning objective $L(\pi)$. This subject is inherent to exploration with intrinsic rewards (Bolland, Lambrechts, and Ernst, 2024).

Many of the existing MaxEntRL algorithms rely on the entropy of the policy for exploring the action space (Haarnoja, Zhou, Abbeel, and Levine, 2018; Toussaint, 2009). The feature space is then the actions space $\mathcal{Z} = \mathcal{A}$, and the conditional distribution is the policy $q^\pi(z|s, a) = \pi(z|s)$, for all a. Other algorithms focus on exploring the state space (Guo et al., 2021; Islam et al., 2019; Lee et al., 2019). The feature space is the state space $\mathcal{Z} = \mathcal{S}$. The conditional distribution $q^\pi(z|s, a)$ is either the marginal probability of states in trajectories of T time steps, or the discounted state visitation measure, for all s and a. In

the literature, the relative measure $q^*(z)$ is usually a uniform distribution, and the relative entropy is computed as the differential entropy, i.e., by neglecting $\log q^*(z)$ in equation (5.2). In continuous spaces, the latter is ill-defined and other measures are sometimes used.

5.3 MAXENTRL WITH VISITATION DISTRIBUTIONS

In this section we introduce a new MaxEntRL framework, i.e., a new intrinsic reward function, followed by a method for approximating the intrinsic reward. The latter approximation requires learning an additional distribution, which can be integrated into existing RL algorithms together with an estimation step for computing the intrinsic reward.

5.3.1 *Definition of the MaxEntRL Framework*

In the following, we introduce a new MaxEntRL framework based on the conditional state-action visitation probability measure $d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)$ and the conditional state visitation probability measure $d^{\pi,\gamma}(\bar{s}|s,a)$

$$d^{\pi,\gamma}(\bar{s},\bar{a}|s,a) = (1-\gamma)\pi(\bar{a}|\bar{s})\sum_{\Delta=1}^{\infty}\gamma^{\Delta}p_{\Delta}^{\pi}(\bar{s}|s,a) \tag{5.3}$$

$$d^{\pi,\gamma}(\bar{s}|s,a) = (1-\gamma)\sum_{\Delta=1}^{\infty}\gamma^{\Delta}p_{\Delta}^{\pi}(\bar{s}|s,a)\,, \tag{5.4}$$

where $p_{\Delta}^{\pi}(\bar{s}|s,a)$ is the probability of reaching $\bar{s}$ in Δ time steps, starting from state s and executing action a before following the policy π. The distribution from equation (5.3) can be factorized as a function of the distribution from equation (5.4) such that $d^{\pi,\gamma}(\bar{s},\bar{a}|s,a) = \pi(\bar{a}|\bar{s})d^{\pi,\gamma}(\bar{s}|s,a)$. The conditional state (respectively, state-action) visitation probability distribution measures the states (respectively, states and actions) that are visited on expectation over infinite trajectories starting from a state and an action. Both distributions generalize the (marginal discounted) state visitation probability measure (Manne, 1960).

In our MaxEntRL framework, the feature space $\mathcal{Z}$ and a feature distribution $h : \mathcal{S}\times\mathcal{A}\to\mathcal{P}(\mathcal{Z})$ are assumed provided. The intrinsic reward is computed according to equation (5.2), for any relative measure q^*, with conditional distribution

$$q^{\pi}(z|s,a) = \int h(z|\bar{s},\bar{a})d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)\,d\bar{s}\,d\bar{a}\,. \tag{5.5}$$

Optimal exploration policies are here intrinsically motivated to take actions so that the discounted visitation measure of future features is distributed according to q^* in each state and for each action. It allows to select features that must be visited during trajectories according to prior knowledge about the problem if any. Alternatively, it allows to only explore lower dimensional feature spaces, or to explore sufficient statistics from the state-action pairs. Furthermore, samples can easily be generated from the distribution equation

(5.5) by sampling a state $\bar{s} \sim d^{\pi,\gamma}(\cdot|s,a)$, an action $\bar{a} \sim \pi(\cdot|\bar{s})$, and finally sampling a feature $z \sim h(\cdot|\bar{s},\bar{a})$. Similarly, the probability of this sample can be estimated solving the integral numerically using samples $\bar{s} \sim d^{\pi,\gamma}(\cdot|s,a)$ and $\bar{a} \sim \pi(\cdot|\bar{s})$.

Let us finally relate MaxEntRL with this new intrinsic reward function to the maximization of a lower bound on the state-action value function of the MDP (computed without intrinsic reward bonuses). We rely on Theorem 3, proved in Appendix 5.A, and extending the previous results from Kakade and Langford, 2002.

Theorem 3. *Let the reward function $R(s,a)$ be non-negative, and let π be a policy with state-action value function $Q^\pi(s,a)$, then,*

$$Q^\pi(s,a) \geq Q^{\pi^*}(s,a) \exp\left(- \| \log \frac{d^{\pi,\gamma}(\cdot,\cdot|s,a)}{d^{\pi^*,\gamma}(\cdot,\cdot|s,a)} \|_\infty \right), \tag{5.6}$$

where $\|f\|_\infty = \sup_x |f(x)|$ is the L_∞-norm of f.

Let us consider the MaxEntRL framework with the intrinsic reward function equation (5.5) where $\mathcal{Z} = \mathcal{S} \times \mathcal{A}$ and $h(z|s,a)$ is a dirac distribution centered in $z = (s,a)$, and with the relative measure $q^*(s,a)$. Let us apply the triangle inequality to the bound in Theorem 3. For any policy π, we get the bound

$$Q^\pi(s,a) \geq Q^{\pi^*}(s,a) \exp\left(- \| \log \frac{d^{\pi,\gamma}(\cdot,\cdot|s,a)}{q^*(\cdot,\cdot)} \|_\infty - \| \log \frac{d^{\pi^*,\gamma}(\cdot,\cdot|s,a)}{q^*(\cdot,\cdot)} \|_\infty \right). \tag{5.7}$$

The bound on the state-action value function of any policy π in equation (5.7) is an exponentially decreasing function of the two error terms $\| \log d^{\pi,\gamma}(\cdot,\cdot|s,a) - \log q^*(\cdot,\cdot)||_\infty$ and $\| \log d^{\pi^*,\gamma}(\cdot,\cdot|s,a) - \log q^*(\cdot,\cdot)||_\infty$. The first can be minimized as a function of π while the second is independent of the policy, and can thus not be reduced. Let us assume that an optimal exploration policy has zero expected discounted sum of intrinsic rewards, and that the target measure and the visitation measures are smooth. Then, an optimal exploration policy maximizes the bound in equation (5.7). Finding such a policy in the MaxEntRL framework we introduce can be seen as a practical algorithm to compute a policy that maximizes the lower bound equation (5.7), given a predefined target. The quality of the optimal exploration policy then only depends on the choice of this target, which can be selected with minimal or no prior knowledge about $d^{\pi^*,\gamma}$ or may even be deliberately different from it.

5.3.2 *Learning Conditional Visitation Models*

As explained in Section 5.2, an intrinsic reward can be computed by sampling from the conditional distribution $q^\pi(z|s,a)$ and evaluating the probability of these samples. Furthermore, in the new MaxEntRL framework introduced in Section 5.3.1, the conditional distribution $q^\pi(z|s,a)$ can be computed based on samples of the conditional state visitation distribution $d^{\pi,\gamma}(\bar{s}|s,a)$. In this section, we explain how to learn the latter distribution.

In order to estimate the conditional state visitation distribution, we first recall that this distribution is a fixed point of the operator $\mathcal{T}^\pi$ defined by Janner et al. (2020)

$$\mathcal{T}^\pi d^{\pi,\gamma}(\bar{s}|s,a) = (1-\gamma)p(\bar{s}|s,a) + \gamma \mathop{\mathbb{E}}_{\substack{s'\sim p(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}} \left[d^{\pi,\gamma}(\bar{s}|s',a')\right] . \tag{5.8}$$

Theorem 4, proved in Appendix 5.A, states that the operator $\mathcal{T}^\pi$ is a contraction mapping, which furthermore implies the uniqueness of its fixed point. Assuming the result of the operator could be computed (or estimated), the fixed point could theoretically also be computed by successive application of this operator. The computation of that fixed point would then allow computing the conditional state-action visitation distribution, and the intrinsic reward function.

Theorem 4. *The operator $\mathcal{T}^\pi$ is γ-contractive in $\bar{L}_n$-norm, where $\bar{L}_n(f)^n = \sup_y \int |f(x|y)|^n \, dx$.*

In practice, computing the result of the operator $\mathcal{T}^\pi$ (and $(\mathcal{T}^\pi)^N$ after N applications) may be intractable when large state and action spaces are at hand or when these spaces are continuous. It furthermore requires having a model of the MDP. An alternative approach is to rely on a function approximator d_ψ to approximate the fixed point. Furthermore, similarly to TD-learning methods (Sutton and Barto, 2018), Theorem 4 suggests to optimize the parameters of this model d_ψ to minimize the residual of the operator, measured with an $\bar{L}_n$-norm for which the operator is γ-contractive. With this metric, estimating the residual requires estimating the transition function (Janner et al., 2020), and cannot be trivially minimized by stochastic gradient descent using transitions from the environment. We therefore propose to solve as surrogate a minimum cross-entropy problem, in which stochastic gradient descent can be applied afterwards. For any policy π, the distribution is approximated with a function approximator d_ψ with parameter ψ optimized to solve

$$\arg\min_\psi \mathop{\mathbb{E}}_{\substack{s,a\sim g(\cdot,\cdot)\\ \bar{s}\sim(\mathcal{T}^\pi)^N d_\psi(\cdot|s,a)}} \left[-\log d_\psi(\bar{s}|s,a)\right] , \tag{5.9}$$

where g is an arbitrary distribution over the state and action space, and where N is any positive integer. This optimization problem is related to minimizing the KL-divergence instead of an $\bar{L}_n$-norm (Bishop and Nasrabadi, 2006).

Let us make explicit how samples from the distribution $(\mathcal{T}^\pi)^N d_\psi(\bar{s}|s,a)$ are generated using the MDP. By definition of the operator $\mathcal{T}^\pi$, the distribution $(\mathcal{T}^\pi)^N d_\psi(\bar{s}|s,a)$ is a

mixture where the probability of samples is a weighted sum composed of the N first multi-step transition probabilities in the MDP and the conditional state visitation model

$$(\mathcal{T}^\pi)^N d_\psi(\bar{s}|s,a) = \left(\sum_{\Delta=1}^{N}(1-\gamma)\gamma^{\Delta-1}p_\Delta^\pi(\bar{s}|s,a)\right) + \gamma^N \mathop{\mathbb{E}}_{\substack{s'\sim p_N^\pi(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}[d_\psi(\bar{s}|s',a')] \tag{5.10}$$

$$= \sum_{\Delta=1}^{\infty}\mathcal{G}_{1-\gamma}(\Delta)b_{\psi,\pi}^\beta(\bar{s}|s,a,\Delta)\Big|_{\beta=\pi}, \tag{5.11}$$

where $\mathcal{G}_{1-\gamma}(\Delta)$ is the probability of the result Δ from a geometric distribution of parameter $1-\gamma$, and where

$$b_{\psi,\pi}^\beta(\bar{s}|s,a,\Delta) = \begin{cases} p_\Delta^\beta(\bar{s}|s,a) & \Delta \leq N \\ \mathop{\mathbb{E}}_{\substack{s'\sim p_N^\beta(\cdot|s,a)\\ a'\sim\pi(\cdot|s')}}[d_\psi(\bar{s}|s',a')] & \Delta > N \end{cases} \tag{5.12}$$

Sampling from $(\mathcal{T}^\pi)^N d_\psi(\bar{s}|s,a)$ consists in sampling from the mixture. First, Δ is drawn from a geometric distribution of parameter $1-\gamma$. Second, a state is sampled as $\bar{s} \sim p_\Delta^\pi(\cdot|s,a)$ if $\Delta \leq N$ or as $\bar{s} \sim d_\psi(\cdot|s',a')$ otherwise; where $s' \sim p_N^\pi(\cdot|s,a)$ and $a' \sim \pi(\cdot|s')$.

Finally, we reformulate the problem equation (5.9) such that it can be estimated from trajectories sampled from any policy β in the MDP. To that end, we apply importance weighting and get the equivalent optimization problem

$$\arg\min_\psi \mathop{\mathbb{E}}_{\substack{s,a\sim g(\cdot,\cdot)\\ \Delta\sim\mathcal{G}_{1-\gamma}(\cdot)\\ \bar{s}\sim b_{\psi,\pi}^\beta(\cdot|s,a,\Delta)}} \left[-\frac{b_{\psi,\pi}^\pi(\bar{s}|s,a,\Delta)}{b_{\psi,\pi}^\beta(\bar{s}|s,a,\Delta)}\log d_\psi(\bar{s}|s,a)\right]. \tag{5.13}$$

In the particular cases where $\beta = \pi$ or where $N = 1$, the importance weight simplifies to one, otherwise it can be simplified to a (finite) product of ratios of policies. We do not delve into more details as we neglected this factor in Section 5.3.3 when using stochastic gradient descent.

5.3.3 *Practical MaxEntRL Exploration Algorithms*

Existing algorithms can finally be adapted to implement the MaxEntRL framework from Section 5.3.1 without substantial modifications. An additional learning step is integrated to update the conditional state visitation model d_ψ to minimize the objective from equation (5.13). The intrinsic reward can then be computed, and this intrinsic reward and the MDP reward are jointly optimized.

LEARNING THE CONDITIONAL STATE VISITATION. At each learning iteration of the RL algorithm, the parameter ψ of the visitation model d_ψ is also updated. First, the objective function from equation (5.13) is estimated by Monte Carlo using trajectories simulated

in the MDP from an arbitrary policy β. The importance weight is neglected. Second, this estimate is differentiated and the parameter ψ is updated by gradient descent steps. Formally, let us assume that N-step transitions $(s_{t:t+N}, a_{t:t+N-1})$ computed from an arbitrary policy β are sampled and stored as a batch or in a replay buffer $\mathcal{D}$. The state-action pair (s_t, a_t) distribution is denotes as g, which depends on the generation procedure of the transitions. The visitation distribution d_ψ corresponding to the optimized policy π_θ is iteratively updated performing stochastic gradient descent steps on the loss function

$$\mathcal{L}(\psi) = -\sum_{s_t, a_t \in \mathcal{D}} \log d_\psi(\bar{s}|s_t, a_t)\,. \tag{5.14}$$

The state $\bar{s} = s_{t+\Delta}$ is available in the batch or replay buffer if $\Delta \leq N$, or $\bar{s} \sim d_{\psi'}(\cdot|s_{t+N}, a_{t+N'})$ is bootstrapped otherwise; where $a_{t+N'} \sim \pi(\cdot|s_{t+N})$ and where ψ' is the target network parameter. In the latter bootstrapping operation, an action is sampled from the policy π, making the algorithm off-policy.

In practice, the gradients generated by differentiating this loss function are biased estimates of the gradients from the objective function equation (5.13). The influence of the parameter ψ on the probability of the sample $\bar{s}$ is neglected, i.e., the partial derivative of $(\mathcal{T}^\pi)^N d_\psi(\bar{s}|s_t, a_t)$ with respect to ψ is neglected, and a target network is used. This is analogue to SARSA and TD-learning strategies (Sutton and Barto, 2018). Furthermore, the importance weights from equation (5.13) is neglected too. It introduces a dependency of the distribution d_ψ on the policy β for the N first steps, which is again similar to multi-step SARSA and multi-step TD-learning.

COMPUTING THE INTRINSIC REWARD. The final modification to adapt existing RL algorithms to this new MaxEntRL framework is to compute the intrinsic reward function every time the reward is processed by the algorithm. For that step, the entropy of the distribution q^π is estimated with

$$R^{int}(s_t, a_t) = \log q^*(z_t) - \log q^\pi(z_t|s_t, a_t)\,, \tag{5.15}$$

where $z_t \sim q^\pi(\cdot|s_t, a_t)$ and where $q^\pi(z_t|s_t, a_t)$ is approximated by Monte Carlo integration of the integral equation (5.5). Note that this integral may have a closed-form depending on the choice of feature space and feature distribution.

In conclusion, existing algorithms can be adapted straightforwardly by adding an additional learning step, and evaluating the intrinsic reward function. This learning step can be integrated to on-policy algorithms, or to off-policy algorithms using solely transitions generated from the MDP when $N = 1$. In practice, we nevertheless observed that choosing the value of N larger than one could drastically improve the learning process. The latter may require a slight adaptation to store multi-step transitions instead of one-step transitions. In Appendix 5.B, off-policy actor-critic (Degris et al., 2012) and soft actor-critic (Haarnoja, Zhou, Abbeel, and Levine, 2018) are adapted as advocated.

5.4 EXPERIMENTS

In this section, we empirically validate the previous MaxEntRL framework by integrating the learning steps into existing algorithms, and evaluating the quality of the exploration and control policies resulting from the algorithms.

5.4.1 *Experimental Setting*

Illustrative experiments are performed on adapted environments from the Minigrid suite (Chevalier-Boisvert et al., 2023). In the latter, an agent must travel across a grid containing walls and passages in order to reach a goal. The size of the grid and the number of passages and walls depend on the environment. The state space is composed of the agent's orientation, its position on the grid, as well as the positions of the passages in the walls and their orientations. In some environments, the goal to be reached is randomly generated and is also part of the state. The agent can take four different actions: turn left, turn right, move forward or stand still. The need for exploration comes from the sparsity of the reward function, which is zero everywhere and equals one in the state to be reached.

In the experiments, we asses the new MaxEntRL framework introduced in Section 5.3.1. In practice off-policy actor-critic (Degris et al., 2012), i.e., an approximate policy iteration algorithm, is adapted to the MaxEntRL framework as advocated in Section 5.3.3. This new algorithm is detailed in Appendix 5.B and is called off-policy actor-critic with conditional visitation measures (OPAC+CV) in the remaining of the chapter. For the Minigrid environments, the features $z \in \mathcal{Z}$ are the pairs of horizontal and vertical positions of the agent in the environment, the function h is a deterministic mapping that computes these positions based on the state-action pairs, and the relative measure q^* is uniform. The state and action space representations, additional details about the function approximators, and other hyperparameters are provided in Appendix 5.C.

The new MaxEntRL algorithm is compared to two alternative algorithms. The first concurrent method is soft actor-critic (SAC) (Haarnoja, Zhou, Abbeel, and Levine, 2018). The latter is a commonly-used MaxEntRL algorithm where the feature space is the action space $\mathcal{Z} = \mathcal{A}$, the conditional distribution is the policy $q^\pi(z|s,a) = \pi(z|s)$ for all a, and the relative measure q^* is uniform. To the best of our knowledge, the MaxEntRL framework used in soft actor-critic is also the unique alternative framework where policies can eventually be computed off-policy when the state and action space is large or continuous. The second concurrent method is a combination between off-policy actor-critic (Degris et al., 2012), and the intrinsic reward function from Lee et al. (2019) and C. Zhang et al. (2021). We refer to that algorithm as off-policy actor-critic with marginal visitation measures (OPAC+MV). Here, the feature space $\mathcal{Z}$ is the same as in OPAC+CV, the conditional distribution $q^\pi(z|s,a)$ is the discounted visitation measure of features for all state s and action a, and the relative measure q^* is uniform. In practice, the state visitation measure is computed by maximum likelihood estimation (Lee et al., 2019), and the feature probability

and intrinsic reward is computed as for OPAC+CV; more details are available in Appendix 5.B. This algorithm is on-policy.

5.4.2 *Exploring Sparse-Reward Environments*

The feature space coverage of optimal exploration policies computed with OPAC+CV, OPAC+MV, and SAC are first compared. In Figure 5.4.1, the evolution of the entropy of the discounted visitation measure of features is represented as a function of the number of algorithm iterations, when only the intrinsic rewards are considered. For each environment, the entropy increases rapidly with the algorithm OPAC+CV and OPAC+MV, and a high-entropy policy results from the optimization. In most environments, OPAC+MV has the largest entropy, followed closely by OPAC+CV, and SAC does not perform well. It is worth noticing that OPAC+CV does not stricto sensus optimize the discounted visitation measure but still challenges the concurrent method that does optimize this objective.

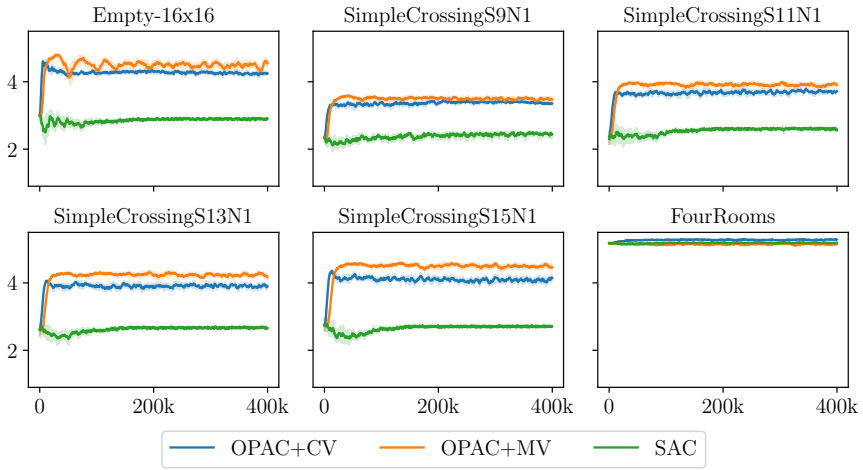


Figure 5.4.1: Evolution of the entropy of the discounted visitation probability measure of the position of the agent on the grid when computing exploration policies (i.e., when neglecting the rewards of the MDP). The entropy is computed empirically with Monte Carlo simulations. For each iteration, the interquartile mean over 15 runs is reported, along with its 95% confidence interval.

In Figure 5.4.2, the evolution of the expected return of the policies is reported during the learning iterations. As can be seen, optimizing the exploration objective presented in Section 5.3.1 with OPAC+CV provides optimal exploration policies with significantly higher expected return compared to OPAC+MV and SAC. Importantly, comparing Figure 5.4.1 and Figure 5.4.2, one can see that policies with small difference of entropy of the discounted visitation measure may achieve very different expected return. This is mostly due to the γ-discounting effect in the definition of the visitation measure. States observed after several time steps in trajectories have small influence on the distribution, but may still provide large rewards, and therefore influence the expected return. The smaller the discount factor, the larger this effect, and the closer the entropy of optimal exploration policies computed with OPAC+CV or OPAC+MV.

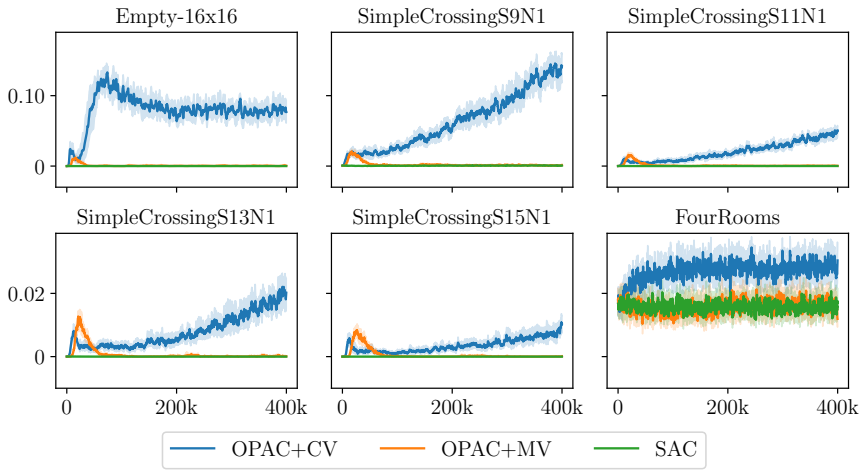


Figure 5.4.2: Expected return during the (exploration) policy optimization with OPAC+CV and OPAC+MV. The expectation is computed empirically with Monte Carlo simulations. For each iteration, the interquartile mean over 15 runs is reported, along with its 95% confidence interval.

In the literature, feature exploration is usually used to compute optimal exploration policies as an initialization when rewards are not available. Our method is an off-policy alternative yielding policies with good feature space coverage and larger expected return.

5.4.3 *Controlling Sparse-Reward Environments*

The objective of MaxEntRL is to provide intrinsic motivations to exploration in order to compute a high-performance policy. In Figure 5.4.3, the expected return of OPAC+CV and OPAC+MV is compared to the expected return of SAC, the most commonly used off-policy MaxEntRL algorithm. As can be seen, our method always performs at least as well as SAC. In the `SimpleCrossing`-environments, the two methods perform equivalently for the first one, OPAC+CV performs similarly to the lucky realizations of SAC for the second one, and only OPAC+CV computes (with high probability) policies with non-zero expected return for the last two. These environments are open grids of different sizes where the agent shall cross a wall through a small passage to reach the target. The larger the environment, the lower the probability of reaching the goal with a uniform policy, and the worst the performance of SAC. The same can be observed in the `Empty-16x16`-environment. On the contrary, both MaxEntRL methods perform equivalently in the `FourRooms`-environment, where complex exploration is apparently not necessary to solve the problem. Finally, our method slightly outperforms OPAC+MV in all environments, except in `SimpleCrossingS15N1` where the concurrent method performs best. Two factors may influence the performance. First, the intrinsic reward functions have different scales, and the weight λ is constant. Second, the expected returns of optimal exploration polices are different, see Figure 5.4.2. Probably the most important is that both methods allow to compute policies with non-zero rewards. With an appropriate scheduling on λ, both methods could eventually compute high-performing policies.

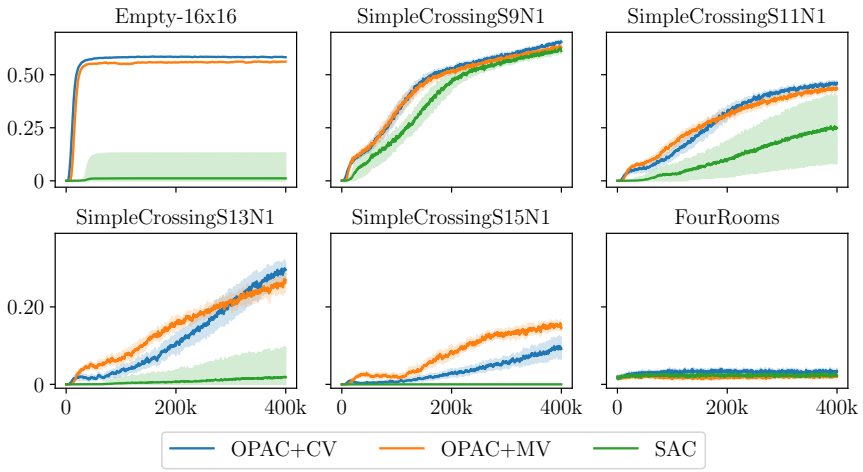


Figure 5.4.3: Expected return during policy optimization with OPAC+CV, OPAC+MV, and SAC. The expectation is approximated with Monte Carlo simulations. For each iteration, the interquartile mean over 15 runs is reported, along with its 95% confidence interval.

5.5 CONCLUSION

In this chapter, we presented a new MaxEntRL framework providing intrinsic reward bonuses proportional to the entropy of the distribution of features built from the states and actions visited by the agent in future time steps. The reward bonus can be estimated efficiently by sampling from the conditional distribution of states visited, which we proved to be the fixed point of a contraction mapping and can be learned for any policy relying on batches of arbitrary transitions. In this MaxEntRL framework, we propose an end-to-end off-policy algorithm that allows to effectively explore the state and action spaces. The algorithm is benchmarked on several control problems. The method we developed is easy to implement, works with a large range of parameters across many environments and can be integrated into already existing RL algorithms.

Future works include testing and adapting the algorithm to continuous state-action spaces, which can straightforwardly be done using continuous neural density estimators, like normalizing flows. Furthermore, in this chapter, the feature space to explore is fixed a priori, but could be learned. A potential avenue is to explore reward-predictive feature spaces. Finally, the distribution that is learned for exploration purpose can be used to generate new samples to enhance the sample efficiency when learning the critic. The integration of the latter into the MaxEntRL framework is left for future work.

APPENDIX

5.A PROOFS THEOREMS

PROOF THEOREM 3. Let us express the state-action value function as a function of the conditional state-action visitation distribution (Eysenbach, Salakhutdinov, and Levine, 2020; Janner et al., 2020)

$$
\begin{aligned}
Q^{\pi}(s,a) &= \frac{1}{1-\gamma}\int d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)R(\bar{s},\bar{a})\,d\bar{s}\,d\bar{a} && (5.16)\\
&= \frac{1}{1-\gamma}\int \frac{d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)}{d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a)} d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a)R(\bar{s},\bar{a})\,d\bar{s}\,d\bar{a} && (5.17)\\
&\geq Q^{\pi*}(s,a)\inf_{\bar{s},\bar{a}} \frac{d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)}{d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a)} && (5.18)\\
&= Q^{\pi*}(s,a)\exp\inf_{\bar{s},\bar{a}}\left(\log \frac{d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)}{d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a)}\right) && (5.19)\\
&= Q^{\pi*}(s,a)\exp\left(\inf_{\bar{s},\bar{a}}\left(\log d^{\pi,\gamma}(s_\infty,\bar{a}|s,a) - \log d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a)\right)\right) && (5.20)\\
&= Q^{\pi*}(s,a)\exp\left(-\sup_{\bar{s},\bar{a}}\left(\log d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a) - \log d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)\right)\right) && (5.21)\\
&\geq Q^{\pi*}(s,a)\exp\left(-\sup_{\bar{s},\bar{a}}\left|\log d^{\pi^*,\gamma}(\bar{s},\bar{a}|s,a) - \log d^{\pi,\gamma}(\bar{s},\bar{a}|s,a)\right|\right) && (5.22)\\
&= Q^{\pi*}(s,a)\exp\left(-\|\log d^{\pi^*,\gamma}(\cdot,\cdot|s,a) - \log d^{\pi,\gamma}(\cdot,\cdot|s,a)\|_\infty\right)\,. && (5.23)
\end{aligned}
$$

Inequation (5.18) holds by the monotonicity of the (Lebesgue) integral, and inequation (5.22) holds as $\sup_x f(x) \leq \sup_x |f(x)|$ for any function f. The theorem generalizes the results from Kakade and Langford, 2002.

□

PROOF THEOREM 4. For all conditional distributions p and q

$$\sup_{s,a} L_n(\mathcal{T}^\pi p(\cdot|s,a), \mathcal{T}^\pi q(\cdot|s,a))^n = \sup_{s,a} \int \|\mathcal{T}^\pi p(\bar{s}|s,a) - \mathcal{T}^\pi q(\bar{s}|s,a)\|_n \, d\bar{s} \tag{5.24}$$

$$= \gamma \sup_{s,a} \int \left\| \mathop{\mathbb{E}}_{\substack{s' \sim p_1(\cdot|s,a) \\ a' \sim \pi(\cdot|s')}} [p(\bar{s}|s',a') - q(\bar{s}|s',a')] \right\|_n d\bar{s} \tag{5.25}$$

$$\leq \gamma \sup_{s,a} \int \mathop{\mathbb{E}}_{\substack{s' \sim p_1(\cdot|s,a) \\ a' \sim \pi(\cdot|s')}} [\|p(\bar{s}|s',a') - q(\bar{s}|s',a')\|_n] \, d\bar{s} \tag{5.26}$$

$$= \gamma \sup_{s,a} \mathop{\mathbb{E}}_{\substack{s' \sim p_1(\cdot|s,a) \\ a' \sim \pi(\cdot|s')}} \left[\int \|p(\bar{s}|s',a') - q(\bar{s}|s',a')\|_n \, d\bar{s} \right] \tag{5.27}$$

$$\leq \gamma \sup_{s,a} \sup_{s',a'} \left(\int \|p(\bar{s}|s',a') - q(\bar{s}|s',a')\|_n \, d\bar{s} \right) \tag{5.28}$$

$$= \gamma \sup_{s',a'} \int \|p(\bar{s}|s',a') - q(\bar{s}|s',a')\|_n \, d\bar{s} \tag{5.29}$$

$$= \gamma \sup_{s,a} L_n(p(\cdot|s,a), q(\cdot|s,a))^n \tag{5.30}$$

□

5.B SOFT AND OFF-POLICY ACTOR-CRITIC WITH CONDITIONAL VISITATION MEASURE

In the following, we adapt soft actor-critic (Haarnoja, Zhou, Abbeel, and Levine, 2018), itself an adaptation of off-policy actor-critic (Degris et al., 2012), according to the procedure from Section 5.3.3. In essence, soft actor-critic estimates the state-action value function with a parameterized critic Q_ϕ, which is learned using expected SARSA (sometimes called generalized SARSA), and updates the parameterized policy π_θ with approximate policy iteration (i.e., off-policy policy gradient), all based on one-step transitions stored in a replay buffer $\mathcal{D}$. The actor and critic loss functions are furthermore extended with the log-likelihood of actions weighted by the parameter λ_{SAC}, therefore called soft and considered a MaxEntRL algorithm using the entropy of policies as intrinsic reward. In the particular case where λ equals zero, the algorithm boils down to a slightly revisited implementation of off-policy actor-critic.

Soft actor-critic is adapted to MaxEntRL with the intrinsic reward function defined in Section 5.3.1, as follows. First, N-step transitions are stored in the buffer $\mathcal{D}$ instead of one-step transitions. Second, the conditional state visitation distribution is estimated with a function approximator d_ψ and learned with stochastic gradient descent applied on the loss function defined in equation (5.14). Third, at each iteration of the critic updates, the reward provided by the MDP is extended with the intrinsic reward.

Formally, the parameterized critic Q_ϕ is iteratively updated performing stochastic gradient descent steps on the loss function

$$\mathcal{L}(\phi) = \mathop{\mathbb{E}}_{s_t,a_t\sim\mathcal{D}}\left[\left(Q_\phi(s_t,a_t) - y\right)^2\right] \tag{5.31}$$

$$y = R(s_t,a_t) + \lambda R^{int}(s_t,a_t) + \gamma\left(Q_{\phi'}(s_{t+1},a_{t+1'}) - \lambda_{SAC}\log\pi_\theta(a_{t+1'}|s_{t+1})\right)\,, \tag{5.32}$$

where $a_{t+1'}\sim\pi_\theta(\cdot|s_{t+1})$, and where ϕ' is the target network parameter.

Furthermore, the policy π_θ is updated performing gradient descent steps on the loss function

$$\mathcal{L}(\theta) = -\mathop{\mathbb{E}}_{s_t,a_t\sim\mathcal{D}}\left[\log\pi_\theta(a_{t'}|s_t)A(s_t,a_{t'})\right] \tag{5.33}$$

$$A(s_t,a_{t'}) = Q_\phi(s_t,a_{t'}) - \lambda_{SAC}\log\pi_\theta(a_{t'}|s_t)\,, \tag{5.34}$$

where $a_{t'}\sim\pi_\theta(\cdot|s_t)$.

Algorithm 5.1 summarizes the learning steps during each iteration. It differs slightly from the original soft actor-critic (Haarnoja, Zhou, Abbeel, and Levine, 2018). The loss equation (5.33) is based on the log-trick instead of the reparametrization trick, the expected SARSA update in equation (5.31) is approximated by sampling, and a single value function is learned, as implemented in CleanRL (S. Huang et al., 2022). These changes are of minor importance in our experiments.

Algorithm 5.1 SAC with conditional visitation measure for exploration

Initialize the policy π_θ, the soft critic Q_ϕ, and the visitation model d_ψ
Initialize the critic target $Q_{\phi'}$ and visitation target $d_{\psi'}$
Initialize the replay buffer with random N-step transitions
while Learning **do**
    Sample transitions from the policy π_θ and add them to the buffer
    while Update the visitation model **do**
        Sample a batch of N-step transitions from the buffer
        Perform a stochastic gradient descent step on $\mathcal{L}(\psi)$
    end while
    while Update the critic **do**
        Sample a batch of N-step transitions from the buffer (use only the 1-step transitions)
        For each element of the batch sample $z_t\sim q^\pi(\cdot|s_t,a_t)$
        Estimate the intrinsic reward $R^{int}(s_t,a_t) = \log q^*(z_t) - \log q^\pi(z_t|s_t,a_t)$
        Perform a stochastic gradient descent step on $\mathcal{L}(\phi)$
    end while
    Sample a batch of N-step transitions from the buffer (use only the 1-step transitions)
    Perform a stochastic gradient descent step on $\mathcal{L}(\theta)$
    Update the target parameters with Poliak averaging
end while

5.C EXPERIMENT HYPERPARAMETERS

In this section, we detail implementation details for reproducing the experiments. In practice, the agent observes the concatenation of the one-hot-encoding of the components of the state space and takes actions in one-hot-encoding format too. The policy π_θ is a neural network that outputs a categorical distribution over the action representation. The critic Q_ϕ is a neural network that takes as input the concatenation of the state and action representations and outputs a scalar. In OPAC+CV, the visitation distribution model d_ψ is also a neural network that takes the same input as the critic Q_ϕ and outputs, for each component of the state space, a categorical distribution over its one-hot-encoding representation. In OPAC+MV, the visitation distribution model d_ψ is a marginal distribution over the same one-hot-encoding representation. In both algorithms, this amounts to assuming the conditional independence of the future state components given the state and action taken as input. This implementation choice mitigates the curse of dimensionality. In addition, it allows to compute the probability of a feature in closed form. The probability equals the product of the probability of the vertical position and the probability of horizontal position provided in one-hot-encoding by the model d_ψ. Table 5.C.1 summarizes the hyperparameters used in the experiments. In practice, the parameter λ_{SAC} is constant for SAC, OPAC+CV, and OPAC+MV simulations.

Table 5.C.1: Hyperparameters

Parameter	Value
Neurons for each network layers	256
Layers policy	2
Layers critic	2
Learning rate policy	10^{-5}
Learning rate critic	10^{-4}
Maximum trajectory length	200
Buffer size	1000
Batch size	32
Critic target update weight τ	0.1
Discount factor γ	0.98
SAC λ_{SAC}	0.002
Layers visitation model OPAC+CV	2
Learning rate visitation model	10^{-5}
MaxEntRL λ	0.01
Density model target update weight τ	1

6

DISCUSSION

This thesis contributes to improving the understanding of the influence of policy stochasticity and what constitutes exploration in policy gradients, and to the development of practical methods to encourage algorithms to explore.

The first question we attempted to answer concerned the influence of policy stochasticity on the learning objective function and, by extension, on the final policy. In particular, why does optimizing a deterministic policy fail, while a disturbed or stochastic policy succeeds? Here, the main conclusion is that optimizing the expected return of affine Gaussian policy that are sufficiently stochastic, i.e., either by manually fixing its variance or by regularizing its entropy, corresponds to optimizing a surrogate to the expected return of an underlying deterministic policy. This surrogate is the expected return of the underlying policy where noise is injected into the policy parameters, similar to applying a filter. This noise depends on the variance of the original Gaussian policy and, in practice, filters out the local optima of the expected return of the underlying deterministic policy. Two conclusions can be drawn from this observation. First, the stochasticity of a policy acts as a filter on the objective function. This observation is consistent with previous results from the literature. Second, if we assume that the goal is to optimize an underlying deterministic policy, which is difficult due to local extrema, a disturbed policy should be optimized instead, and the variance of that policy should not be optimized to locally improve the expected return but rather to avoid local optima. Moreover, from this point of view, there is no obvious reason to be restricted to Markov policies, and allowing the variance to be history-dependent could enable better filtering of local optima.

As explained, one way to enforce policy stochasticity is through entropy regularization and, more generally, by adding intrinsic reward bonuses. The second part of the thesis provides an analysis of such learning objective functions and the gradient estimates built with policy gradient algorithms. Criteria are introduced for the learning objective and gradient estimates, under which policy gradient algorithms are prone to succeed in computing optimal policies. The analysis is mainly a proof of concept, but experiments performed in a simplified context indicate that intrinsic bonuses exploring in the maximum entropy fashion help to satisfy the defined criteria. Here, we provide a framework for studying and developing intrinsic reward bonuses and validate their influence and importance on the convergence of policy gradient algorithms.

Let us re-emphasize the importance of maintaining policies stochastic, which can be achieved using intrinsic reward bonuses. These bonuses implement a form of exploration that aligns with intuitive understanding, and the corresponding learning objective functions appear to have the right properties for optimization using policy gradient methods. In practice, designing bonuses that enable proper exploration, are easy to compute, and can be integrated into policy gradients is a challenging task. In the last part of the thesis, we proposed a new intrinsic reward bonus, introducing a new framework for maximum entropy reinforcement learning. For each state and action, this intrinsic reward is the relative entropy of the discounted distribution of states and actions (or features derived from these states and actions) visited during the next time steps. Computing these rewards requires an additional step in the policy gradient algorithm to learn an additional density estimator, but which can be done efficiently off-policy. An end-to-end off-policy policy gradient algorithm is finally developed. This algorithm demonstrates efficiency compared to alternative methods in the literature and introduces a fundamentally different approach to exploration compared with existing strategies.

Throughout this thesis, additional insights into exploration are provided, with the analysis of optimized objective functions, in alignment with active lines of research. The link between deterministic and stochastic policy optimization has been further refined recently by Montenegro et al. (2024). They proved the convergence of algorithms optimizing a deterministic policy by following the gradient of that policy with disturbed actions under specific assumptions. Regarding the study of the objective function with reward shaping, this thesis extends the work of Ahmed et al. (2019), who had already empirically observed that entropy regularization removes local optima. There remains uncertainty as to how to leverage the various observations from these research lines to improve algorithm efficiency in practice. The choice of policy stochasticity or action disturbance remains an open question, and to the best of our knowledge, the convergence of methods incorporating intrinsic reward bonuses beyond entropy regularization has not yet been proven. Our analysis was similarly limited to a proof of concept when studying the objective and convergence properties, without reaching formal conclusions unless imposing overly restrictive assumptions. The final part of this thesis is more practical, focusing on the development of a new maximum-entropy reinforcement learning framework. This work is also part of an ongoing search for new state-space exploration methods, building on the pioneering work of Hazan, Kakade, et al. (2019). We propose a method that also explores states but in a fundamentally different way. While it has theoretical advantages, including the ability to be off-policy, there are no theoretical convergence guarantees yet. As the method is still very recent, it has not been tested on a wide range of environments, and it is too early to conclude about its performance compared to competing methods.

It is worth re-emphasizing that this thesis opens up new perspectives for understanding exploration in policy gradient methods and also proposes a new method supported by preliminary results. A central observation is that theorists have only recently begun to show interest in studying the convergence of policy gradient algorithms. In these studies,

sufficient exploration appears to be at the core of the required assumptions, yet little theoretically supported guidance has been provided so far to ensure these assumptions are respected in practice. Practitioners, on the contrary, are increasingly engineering new exploration methods. It remains nevertheless unclear whether more complex solutions truly outperform those of seminal works. The future of policy gradients will likely lie in resolving this dichotomy; through the standardization of empirical protocols for comparing algorithms, with theoretical studies on these methods providing practical guidance, and, we all hope, the development of algorithms computing optimal policies efficiently.

BIBLIOGRAPHY

Agarwal, A., Kakade, S. M., Lee, J. D., & Mahajan, G. (2020). Optimality and approximation with policy gradient methods in markov decision processes. *Conference on Learning Theory*, 64–66.

Ahmed, Z., Le Roux, N., Norouzi, M., & Schuurmans, D. (2019). Understanding the impact of entropy on policy optimization. *International Conference on Machine Learning*, 151–160.

Aittahar, S., Bolland, A., Derval, G., & Ernst, D. (2024). Optimal control of renewable energy communities subject to network peak fees with model predictive control and reinforcement learning algorithms. *arXiv preprint arXiv:2401.16321*.

Ajalloeian, A., & Stich, S. U. (2020). On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*.

Allgower, E. L., & Georg, K. (1980). *Numerical continuation methods: An introduction* (Vol. 13). Springer Series in Computational Mathematics.

Amari, S.-I. (1997). Neural learning in structured parameter spaces-natural riemannian gradient. *Advances in Neural Information Processing Systems*, 127–133.

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., et al. (2020). What matters in on-policy reinforcement learning? A large-scale empirical study. *arXiv preprint arXiv:2006.05990*.

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

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., & Silver, D. (2017). Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, *30*.

Baxter, J., & Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, *15*, 319–350.

Bedi, A. S., Chakraborty, S., Parayil, A., Sadler, B. M., Tokekar, P., & Koppel, A. (2022). On the hidden biases of policy mirror ascent in continuous action spaces. *International Conference on Machine Learning*, 1716–1731.

Bedi, A. S., Parayil, A., Zhang, J., Wang, M., & Koppel, A. (2021). On the sample complexity and metastability of heavy-tailed policy search in continuous control. *arXiv preprint arXiv:2106.08414*.

Bellemare, M., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., & Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. *Advances in Neural Information Processing Systems*, *29*.

Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, *2*(1), 1–127.

Bhandari, J., & Russo, D. (2019). Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*.

Bhandari, J., & Russo, D. (2020). A note on the linear convergence of policy gradient methods. *arXiv preprint arXiv:2007.11120*, *79*.

Bhatt, S., Koppel, A., & Krishnamurthy, V. (2019). Policy gradient using weak derivatives for reinforcement learning. *Conference on Decision and Control (CDC)*, *58*, 5531–5537.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Blake, A., & Zisserman, A. (1987). *Visual reconstruction*. MIT press.

Bolland, A., Lambrechts, G., & Ernst, D. (2024). Behind the myth of exploration in policy gradients. *arXiv preprint arXiv:2402.00162*.

Bolland, A., Louppe, G., & Ernst, D. (2023). Policy gradient algorithms implicitly optimize by continuation. *Transactions on Machine Learning Research*.

Bottou, L. (1998). Online learning and stochastic approximations. *Online learning in neural networks*, *17*(9), 142.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of compstat'2010* (pp. 177–186). Springer.

Boukas, I., Ernst, D., Théate, T., Bolland, A., Huynen, A., Buchwald, M., Wynants, C., & Cornélusse, B. (2021). A deep reinforcement learning framework for continuous intraday market bidding. *Machine Learning*, *110*, 2335–2387.

Brekelmans, R., Genewein, T., Grau-Moya, J., Delétang, G., Kunesch, M., Legg, S., & Ortega, P. (2022). Your policy regularizer is secretly an adversary. *arXiv preprint arXiv:2203.12592*.

Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., & Efros, A. A. (2018). Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*.

Busoniu, L., Babuska, R., De Schutter, B., & Ernst, D. (2017). *Reinforcement learning and dynamic programming using function approximators*. CRC press.

Cen, S., Cheng, C., Chen, Y., Wei, Y., & Chi, Y. (2022). Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, *70*(4), 2563–2578.

Chen, J., & Luss, R. (2018). Stochastic gradient descent with biased but consistent gradient estimators. *arXiv preprint arXiv:1807.11880*.

Chernoff, H., & Moses, L. E. (2012). *Elementary decision theory*. Courier Corporation.

Chevalier-Boisvert, M., Dai, B., Towers, M., de Lazcano, R., Willems, L., Lahlou, S., Pal, S., Castro, P. S., & Terry, J. (2023). Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, *abs/2306.13831*.

Chou, P.-W., Maturana, D., & Scherer, S. (2017). Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. *International Conference on Machine Learning*, 834–843.

Chung, W., Thomas, V., Machado, M. C., & Le Roux, N. (2021). Beyond variance reduction: Understanding the true impact of baselines on policy optimization. *International Conference on Machine Learning*, 1999–2009.

Dann, C., Lattimore, T., & Brunskill, E. (2017). Unifying pac and regret: Uniform pac bounds for episodic reinforcement learning. *Advances in Neural Information Processing Systems*, *30*.

Degris, T., White, M., & Sutton, R. S. (2012). Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., & Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. *International Conference on Machine Learning*, 1329–1338.

Eysenbach, B., Myers, V., Levine, S., & Salakhutdinov, R. (2023). Contrastive representations make planning easy. *NeurIPS 2023 Workshop on Generalization in Planning*.

Eysenbach, B., Salakhutdinov, R., & Levine, S. (2020). C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*.

Eysenbach, B., Zhang, T., Levine, S., & Salakhutdinov, R. R. (2022). Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, *35*, 35603–35620.

Fazel, M., Ge, R., Kakade, S., & Mesbahi, M. (2018). Global convergence of policy gradient methods for the linear quadratic regulator. *International conference on machine learning*, 1467–1476.

Forbes, G. C., Gupta, N., Villalobos-Arias, L., Potts, C. M., Jhala, A., & Roberts, D. L. (2024). Potential-based reward shaping for intrinsic motivation. *arXiv preprint arXiv:2402.07411*.

Fujita, Y., & Maeda, S.-i. (2018). Clipped action policy gradient. *International Conference on Machine Learning*, 1597–1606.

Glynn, P. W. (1990). Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, *33*(10), 75–84.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, *5*(9).

Guo, Z. D., Azar, M. G., Saade, A., Thakoor, S., Piot, B., Pires, B. A., Valko, M., Mesnard, T., Lattimore, T., & Munos, R. (2021). Geometric entropic exploration. *arXiv preprint arXiv:2101.02055*.

Haarnoja, T., Pong, V., Zhou, A., Dalal, M., Abbeel, P., & Levine, S. (2018). Composable deep reinforcement learning for robotic manipulation. *2018 IEEE international conference on robotics and automation (ICRA)*, 6244–6251.

Haarnoja, T., Tang, H., Abbeel, P., & Levine, S. (2017). Reinforcement learning with deep energy-based policies. *International Conference on Machine Learning*, 1352–1361.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International conference on machine learning*, 1861–1870.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2019). Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*.

Hansson, S. O. (2005). Decision theory: A brief introduction.

Harutyunyan, A., Devlin, S., Vrancx, P., & Nowé, A. (2015). Expressing arbitrary reward functions as potential-based advice. *Proceedings of the AAAI conference on artificial intelligence*, *29*(1).

Hazan, E., Kakade, S., Singh, K., & Van Soest, A. (2019). Provably efficient maximum entropy exploration. *International Conference on Machine Learning*, 2681–2691.

Hazan, E., Levy, K. Y., & Shalev-Shwartz, S. (2016). On graduated optimization for stochastic non-convex problems. *International Conference on Machine Learning*, 1833–1841.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning. Lecture 6a. Overview of mini-batch gradient descent.

Houthooft, R., Chen, Y., Isola, P., Stadie, B., Wolski, F., Jonathan Ho, O., & Abbeel, P. (2018). Evolved policy gradients. *Advances in Neural Information Processing Systems*, *31*.

Huang, S., Dossa, R. F. J., Ye, C., Braga, J., Chakraborty, D., Mehta, K., & Araújo, J. G. (2022). Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, *23*(274), 1–18.

Husain, H., Ciosek, K., & Tomioka, R. (2021). Regularized policies are reward robust. *International Conference on Artificial Intelligence and Statistics*, 64–72.

Islam, R., Ahmed, Z., & Precup, D. (2019). Marginalized state distribution entropy regularization in policy optimization. *arXiv preprint arXiv:1912.05128*.

Jain, A. K., Lehnert, L., Rish, I., & Berseth, G. (2024). Maximum state entropy exploration using predecessor and successor representations. *Advances in Neural Information Processing Systems*, *36*.

Janner, M., Mordatch, I., & Levine, S. (2020). Generative temporal difference learning for infinite-horizon prediction. *arXiv preprint arXiv:2010.14496*.

Kakade, S. (2001). A natural policy gradient. *Advances in Neural Information Processing Systems*, *14*.

Kakade, S., & Langford, J. (2002). Approximately optimal approximate reinforcement learning. *International Conference on Machine Learning*, *19*.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018). Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*.

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, L., Eysenbach, B., Parisotto, E., Xing, E., Levine, S., & Salakhutdinov, R. (2019). Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*.

Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, B., Cai, Q., Yang, Z., & Wang, Z. (2019). Neural proximal/trust region policy optimization attains globally optimal policy (2019). *arXiv preprint arXiv:1906.10306*.

Liu, H., & Abbeel, P. (2021). Behavior from the void: Unsupervised active pre-training. *Advances in Neural Information Processing Systems*, *34*, 18459–18473.

Liu, Y., Zhang, K., Basar, T., & Yin, W. (2022). An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, *33*, 7624–7636.

Ma, Y. J., Sivakumar, K., Yan, J., Bastani, O., & Jayaraman, D. (2023). Learning policy-aware models for model-based reinforcement learning via transition occupancy matching. *Learning for Dynamics and Control Conference*, 259–271.

Mangasarian, O. L. (1975). Pseudo-convex functions. In *Stochastic optimization models in finance* (pp. 23–32). Elsevier.

Manne, A. S. (1960). Linear programming and sequential decisions. *Management Science*, *6*(3), 259–267.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization science*, *2*(1), 71–87.

Mazoure, B., Bruce, J., Precup, D., Fergus, R., & Anand, A. (2023). Accelerating exploration and representation learning with offline pre-training. *arXiv preprint arXiv:2304.00046*.

Mazoure, B., Eysenbach, B., Nachum, O., & Tompson, J. (2023). Contrastive value learning: Implicit models for simple offline rl. *Conference on Robot Learning*, 1257–1267.

Mazoure, B., Talbott, W., Bautista, M. A., Hjelm, D., Toshev, A., & Susskind, J. (2023). Value function estimation using conditional diffusion models for control. *arXiv preprint arXiv:2306.07290*.

Mei, J., Dai, B., Xiao, C., Szepesvari, C., & Schuurmans, D. (2021). Understanding the effect of stochasticity in policy optimization. *Advances in Neural Information Processing Systems*, *34*, 19339–19351.

Mei, J., Xiao, C., Dai, B., Li, L., Szepesvári, C., & Schuurmans, D. (2020). Escaping the gravitational pull of softmax. *Advances in Neural Information Processing Systems*, *33*, 21130–21140.

Mei, J., Xiao, C., Szepesvari, C., & Schuurmans, D. (2020). On the global convergence rates of softmax policy gradient methods. *International conference on machine learning*, 6820–6829.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533.

Mobahi, H., & Fisher, J. W. (2015). On the link between gaussian homotopy continuation and convex envelopes. *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 43–56.

Mobahi, H., & Fisher III, J. (2015). A theoretical analysis of optimization by gaussian continuation. *Conference on Artificial Intelligence*, *29*(1).

Mobahi, H., Zitnick, C. L., & Ma, Y. (2012). Seeing through the blur. *Conference on Computer Vision and Pattern Recognition*, 1736–1743.

Montenegro, A., Mussi, M., Metelli, A. M., & Papini, M. (2024). Learning optimal deterministic policies with stochastic policy gradients. *arXiv preprint arXiv:2405.02235*.

Murray, W., & Ng, K.-M. (2010). An algorithm for nonlinear optimization problems with binary variables. *Computational optimization and applications*, *47*(2), 257–288.

Mutti, M., De Santi, R., & Restelli, M. (2022). The importance of non-markovianity in maximum state entropy exploration. *arXiv preprint arXiv:2202.03060*.

Mutti, M., Pratissoli, L., & Restelli, M. (2021). Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(10), 9028–9036.

Mutti, M., & Restelli, M. (2020). An intrinsically-motivated approach for learning highly exploring and fast mixing policies. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 5232–5239.

Nachum, O., Norouzi, M., & Schuurmans, D. (2016). Improving policy gradient by exploring under-appreciated rewards. *arXiv preprint arXiv:1611.09321*.

Nesterov, Y., & Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, *17*(2), 527–566.

Neu, G., & Pike-Burke, C. (2020). A unifying view of optimism in episodic reinforcement learning. *Advances in Neural Information Processing Systems*, *33*, 1392–1403.

Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *Icml*, *99*, 278–287.

Papini, M., Battistello, A., & Restelli, M. (2020). Balancing learning speed and stability in policy gradient via adaptive exploration. *International conference on artificial intelligence and statistics*, 1188–1199.

Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. *International conference on machine learning*, 2778–2787.

Pathak, H. N., & Paffenroth, R. (2019). Parameter continuation methods for the optimization of deep neural networks. *International Conference on Machine Learning And Applications (ICMLA)*, *18*, 1637–1643.

Patil, G., Mahajan, A., & Precup, D. (2022). On learning history based policies for controlling markov decision processes. *arXiv preprint arXiv:2211.03011*.

Peters, J., & Schaal, S. (2007). Reinforcement learning by reward-weighted regression for operational space control. *International conference on Machine learning*, *24*, 745–750.

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

Precup, D. (2000). Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 80.

Rajeswaran, A., Lowrey, K., Todorov, E. V., & Kakade, S. M. (2017). Towards generalization and simplicity in continuous control. *Advances in Neural Information Processing Systems*, *30*.

Rao, C. R. (1973). *Linear statistical inference and its applications* (Vol. 2). Wiley New York.

Salimans, T., Ho, J., Chen, X., Sidor, S., & Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*.

Scherrer, B., & Geist, M. (2014). Local policy search in a convex space and conservative policy iteration as boosted policy search. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part III 14*, 35–50.

Schulman, J., Chen, X., & Abbeel, P. (2017). Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. *International Conference on Machine Learning*, 1889–1897.

Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., & Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Networks*, *23*(4), 551–559.

Seo, Y., Chen, L., Shin, J., Lee, H., Abbeel, P., & Lee, K. (2021). State entropy maximization with random encoders for efficient exploration. *International Conference on Machine Learning*, 9443–9454.

Shani, L., Efroni, Y., & Mannor, S. (2020). Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. *Conference on Artificial Intelligence*, *34*(4), 5668–5675.

Shao, W., Geißler, C., & Sivrikaya, F. (2019). Graduated optimization of black-box functions. *arXiv preprint arXiv:1906.01279*.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. *International Conference on Machine Learning*, 387–395.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354–359.

Simon, H. A. (1955). A behavioral model of rational choice. *The quarterly journal of economics*, *69*(1), 99–118.

Siotani, M. (1967). Some applications of loewner's ordering on symmetric matrices. *Annals of the Institute of Statistical Mathematics*, *19*, 245–259.

Staines, J., & Barber, D. (2012). Variational optimization. *arXiv preprint arXiv:1212.4507*.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., McAllester, D. A., Singh, S. P., Mansour, Y., et al. (1999). Policy gradient methods for reinforcement learning with function approximation. *Neural Information Processing Systems*, *99*, 1057–1063.

Tiapkin, D., Belomestny, D., Calandriello, D., Moulines, E., Munos, R., Naumov, A., Perrault, P., Tang, Y., Valko, M., & Menard, P. (2023). Fast rates for maximum entropy exploration. *International Conference on Machine Learning*, 34161–34221.

Toussaint, M. (2009). Robot trajectory optimization using approximate inference. *International Conference on Machine Learning*, *26*, 1049–1056.

Wang, L., Cai, Q., Yang, Z., & Wang, Z. (2019). Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*.

Watson, L. T., & Haftka, R. T. (1989). Modern homotopy methods in optimization. *Computer Methods in Applied Mechanics and Engineering*, *74*(3), 289–305.

Weaver, L., & Tao, N. (2013). The optimal reward baseline for gradient-based reinforcement learning. *arXiv preprint arXiv:1301.2315*.

Wierstra, D., Schaul, T., Peters, J., & Schmidhuber, J. (2008). Episodic reinforcement learning by logistic reward-weighted regression. *International Conference on Artificial Neural Networks*, 407–416.

Wiewiora, E., Cottrell, G. W., & Elkan, C. (2003). Principled methods for advising reinforcement learning agents. *Proceedings of the 20th international conference on machine learning (ICML-03)*, 792–799.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, *8*(3-4), 229–256.

Williams, R. J., & Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, *3*(3), 241–268.

Xiong, H., Xu, T., Zhao, L., Liang, Y., & Zhang, W. (2022). Deterministic policy gradient: Convergence analysis. *Conference on Uncertainty in Artificial Intelligence*, *28*.

Yarats, D., Fergus, R., Lazaric, A., & Pinto, L. (2021). Reinforcement learning with prototypical representations. *International Conference on Machine Learning*, 11920–11931.

Yuan, R., Gower, R. M., & Lazaric, A. (2022). A general sample complexity analysis of vanilla policy gradient. *International Conference on Artificial Intelligence and Statistics*, 3332–3380.

Zhang, C., Cai, Y., Huang, L., & Li, J. (2021). Exploration by maximizing rényi entropy for reward-free rl framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 10859–10867.

Zhang, J. [Jiaxin], Tran, H., Lu, D., & Zhang, G. (2020). A novel evolution strategy with directional gaussian smoothing for blackbox optimization. *arXiv preprint arXiv:2002.03001*.

Zhang, J. [Junyu], Koppel, A., Bedi, A. S., Szepesvari, C., & Wang, M. (2020). Variational policy gradient method for reinforcement learning with general utilities. *Advances in Neural Information Processing Systems*, *33*, 4572–4583.

Zhang, J. [Junzi], Kim, J., O'Donoghue, B., & Boyd, S. (2021). Sample efficient reinforcement learning with reinforce. *Conference on Artificial Intelligence*, *35*(12), 10887–10895.

Zhang, K., Koppel, A., Zhu, H., & Basar, T. (2020). Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, *58*(6), 3586–3612.

Zhang, T., Rashidinejad, P., Jiao, J., Tian, Y., Gonzalez, J. E., & Russell, S. (2021). Made: Exploration via maximizing deviation from explored regions. *Advances in Neural Information Processing Systems*, *34*, 9663–9680.

Zhang, T., Xu, H., Wang, X., Wu, Y., Keutzer, K., Gonzalez, J. E., & Tian, Y. (2021). Noveld: A simple yet effective exploration criterion. *Advances in Neural Information Processing Systems*, *34*, 25217–25230.

Zhao, M., Li, Y., & Wen, Z. (2019). A stochastic trust-region framework for policy optimization. *arXiv preprint arXiv:1911.11640*.

Ziebart, B. D. (2010). *Modeling purposeful adaptive behavior with the principle of maximum causal entropy* [Doctoral dissertation, Carnegie Mellon University].

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. *Aaai*, *8*, 1433–1438.