An Analysis of Nonparametric Off-Policy Policy Gradient Estimation

Samuele Tosatto, João Carvalho, and Jan Peters

Abstract—Off-policy Reinforcement Learning (RL) holds the promise of better data efficiency as it allows sample reuse and potentially enables safe interaction with the environment. Current off-policy policy gradient methods either suffer from high bias or high variance, delivering often unreliable estimates. The price of inefficiency becomes evident in real world scenarios such as interaction-driven robot learning, where the success of RL has been rather limited, and a very high sample cost hinders straightforward application. In this paper we propose a nonparametric Bellman equation, which can be solved in closed form. The solution is differentiable w.r.t the policy parameters and gives access to an estimation of the policy gradient. In this way, we avoid the high variance of importance sampling approaches, and the high bias of semi-gradient methods. We empirically analyze the quality of our gradient estimate against state-of-the-art methods, and we show that it outperforms the baselines in terms of sample efficiency on classical control tasks.

Index Terms—Reinforcement Leanring, Policy Gradient, Nonparametric Estimation.

1 Introduction

Reinforcement learning has made overwhelming progress in recent years, especially when applied to board and computer games, or simulated tasks [1]–[3]. However, in comparison, only a little improvement has been achieved on real-world tasks. One of the reasons of this gap is that the vast majority of reinforcement learning approaches are on-policy. On-policy algorithms require that the samples are collected using the optimization policy; and therefore this implies that a) there is little control on the environment and b) samples must be discarded after each policy improvement, causing high sample inefficiency. In contrast, off-policy techniques are theoretically more sample efficient, because they decouple the procedures of data acquisition and policy update, allowing for the possibility of sample-reuse, and enable a higher degree of control on the data-acquisition process, which allows for safe interaction. These two properties are of crucial importance for real-world scenarios. However, classical off-policy algorithms like Q-learning with function approximation and fitted Q-iteration [4], [5] are not guaranteed to converge [6], [7], and allow only discrete actions. More recent semi-gradient[1] off-policy techniques, like Off-PAC [9] and DDPG [10], [11] often perform sub-optimally, especially when the collected data is strongly off-policy, due to the biased semi-gradient update [12]. Off-policy algorithms based on importance sampling [13]–[15] deliver an unbiased estimate of the gradient but suffer from high variance and are generally only applicable with stochastic policies. Moreover, they require the full knowledge of the behavioral policy, making them unsuitable when data stems from a human demonstrator. Additionally, model-based approaches like PILCO [16] may be considered to be off-policy. Such probabilistic nonlinear trajectory optimizers are limited to the finite-horizon domain and suffer from coarse approximations when propagating the state distribution through time. To address all previously highlighted issues in state-of-the-art off-policy approaches, we propose a new algorithm, the nonparametric off-policy policy gradient (NOPG) [17], a full-gradient estimate based on the closed-form solution of a nonparametric Bellman equation. We avoid the high variance of importance sampling techniques and allow for the use of human demonstrations, and unlike other nonparametric methods like PILCO, our approach allows for multimodal state-transitions, and can handle the infinite-horizon setting. Figure 1 shows the optimization cycle of NOPG. A behavioral policy, represented by a human demonstrator, provides (possibly suboptimal) trajectories that solve a task. NOPG optimizes a policy from off-line and off-policy samples. The two other approaches, semi-gradient and path-wise importance sampling, do not work in this scenario.

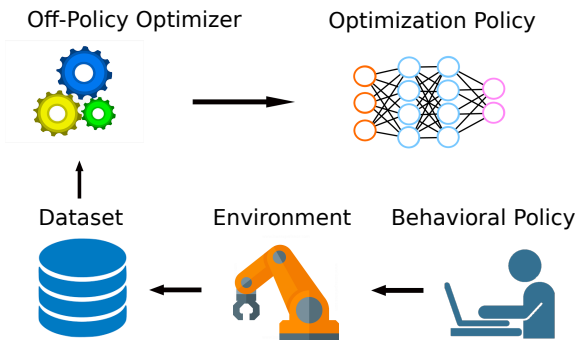


Fig. 1: In the off-policy reinforcement learning scheme, the policy can be optimized using an off-policy dataset. This allows for safer interaction with the system and for better sample efficiency.

- *All the authors are with the Technische Universität Darmstadt, ¨ Darmstadt, Germany, FG Intelligent Autonomous Systems. E-mail: {name}.{surname}@tu-darmstadt.de.*
- *Jan Peters is also with the Max-Planck-Institut für Intelligente Systeme, ¨ Tübingen, Germany.*

1. We adopt the terminology from [8].

In this paper we present both the theoretical foundations of our approach, and an empirical analysis to compare the quality of our gradient estimate and the sample efficiency w.r.t. state-of-the art techniques.

2 Problem Statement

Consider the reinforcement learning problem of an agent interacting with a given environment, as abstracted by a Markov decision process (MDP) and defined over the tuple $(\mathcal{S}, \mathcal{A}, \gamma, P, R, \mu_0)$ where $\mathcal{S} \equiv \mathbb{R}^{d_s}$ is the state space, $\mathcal{A} \equiv \mathbb{R}^{d_a}$ the action space, the transition-based discount factor γ is a stochastic mapping between $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ to $[0, 1)$, which allows for unification of episodic and continuing tasks [18], offering, among others, a natural representation of task termination (where $\gamma = 0$). The transition probability from a state $\mathbf{s}$ to $\mathbf{s}'$ given an action $\mathbf{a}$ is governed by the conditional density $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$. The stochastic reward signal R for a transition $(\mathbf{s}, \mathbf{a}, \mathbf{s}') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$ is drawn from a distribution $R(\mathbf{s}, \mathbf{a}, \mathbf{s}')$ with mean value $\mathbb{E}_{\mathbf{s}'}[R(\mathbf{s}, \mathbf{a}, \mathbf{s}')] = r(\mathbf{s}, \mathbf{a})$. The initial distribution $\mu_0(\mathbf{s})$ denotes the probability of the state $\mathbf{s} \in \mathcal{S}$ to be a starting state. A policy π is a stochastic or deterministic mapping from $\mathcal{S}$ onto $\mathcal{A}$, usually parametrized by a set of parameters θ.

We define an *episode* as $\tau \equiv \{\mathbf{s}_t, \mathbf{a}_t, r_t, \gamma_t\}_{t=1}^{\infty}$ where

$$\mathbf{s}_0 \sim \mu_0(\cdot); \mathbf{a}_t \sim \pi(\cdot \mid \mathbf{s}_t); \mathbf{s}_{t+1} \sim p(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$$
$$r_t \sim R(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}), \gamma_t \sim \gamma(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}).$$

In this paper we consider the discounted infinite-horizon setting, where the objective is to maximize the expected return

$$J_\pi = \mathop{\mathbb{E}}_{\tau}\left[\sum_{t=0}^{\infty} r_t \prod_{i=0}^{t} \gamma_i\right]. \tag{1}$$

It is propaedeutic to introduce two important quantities: the *stationary state visitation* μ_π and the *value function* V_π. We naturally extend the stationary state visitation defined by [19] with the transition-based discount factor

$$\mu(\mathbf{s}) = \mathop{\mathbb{E}}_{\tau}\left[\sum_{t=0}^{\infty} p(\mathbf{s} = \mathbf{s}_t|\pi, \mu_0) \prod_{i=1}^{t} \gamma_i\right],$$

or, equivalently, as the fixed point of

$$\mu(\mathbf{s}) = \mu_0(\mathbf{s}) + \int_{\mathcal{S}}\int_{\mathcal{A}} p_\gamma(\mathbf{s}|\mathbf{s}', \mathbf{a}')\pi(\mathbf{a}'|\mathbf{s}')\mu_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}'\,\mathrm{d}\mathbf{a}'$$

where, from now on, $p_\gamma(\mathbf{s}'|\mathbf{s}, \mathbf{a}) = p(\mathbf{s}'|\mathbf{s}, \mathbf{a})\mathbb{E}[\gamma(\mathbf{s}, \mathbf{a}, \mathbf{s}')]$. The value function

$$V_\pi(\mathbf{s}) = \mathop{\mathbb{E}}_{\tau}\left[\sum_{t=0}^{\infty} r_t p(r_t|\mathbf{s}_0 = \mathbf{s}, \pi) \prod_{i=0}^{t} \gamma_i\right],$$

corresponds to the fixed point of the Bellman equation,

$$V_\pi(\mathbf{s}) = \int_{\mathcal{A}} \pi(\mathbf{a}|\mathbf{s})\left(r(\mathbf{s}, \mathbf{a}) + \int_{\mathcal{S}} V_\pi(\mathbf{s}')p_\gamma(\mathbf{s}'|\mathbf{s}, \mathbf{a})\,\mathrm{d}\mathbf{s}'\right)\mathrm{d}\mathbf{a}.$$

The state-action value function is defined as

$$Q_\pi(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \int_{\mathcal{S}} V_\pi(\mathbf{s}')p_\gamma(\mathbf{s}'|\mathbf{s}, \mathbf{a})\,\mathrm{d}\mathbf{s}'.$$

The expected return (1) can be formulated as

$$J_\pi = \int_{\mathcal{S}} \mu_0(\mathbf{s})V_\pi(\mathbf{s})\,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}}\int_{\mathcal{A}} \mu_\pi(\mathbf{s})\pi(\mathbf{a}|\mathbf{s})r(\mathbf{s}, \mathbf{a})\,\mathrm{d}\mathbf{a}\,\mathrm{d}\mathbf{s}.$$

Policy Gradient Theorem. Objective (1) is usually maximized via gradient ascent. The gradient of J_π w.r.t. the policy parameters θ is

$$\nabla_\theta J_\pi = \int_{\mathcal{S}}\int_{\mathcal{A}} \mu_\pi(\mathbf{s})\pi_\theta(\mathbf{a}|\mathbf{s})Q_\pi(\mathbf{s}, \mathbf{a})\nabla_\theta \log \pi_\theta(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\,\mathrm{d}\mathbf{s},$$

as stated in the policy gradient theorem [19]. When it is possible to interact with the environment with the policy π_θ, one can approximate the integral by considering the state-action as a distribution (up to a normalization factor) and use the samples to perform a Monte-Carlo (MC) estimation [20]. The Q-function can be estimated via Monte-Carlo sampling, approximate dynamic programming or by direct Bellman minimization. In the off-policy setting, we do not have access to the state-visitation μ_π induced by the policy, but instead we observe a different state distribution. While estimating the Q-function with the new state distribution is well established in the literature [4], [21], the shift in the state visitation $\mu_\pi(\mathbf{s})$ is more difficult to obtain. State-of-the-art techniques either omit to consider this shift (we refer to these algorithms as *semi-gradient*), or they try to estimate it via importance sampling correction. These approaches will be discussed in detail in Section 4.

3 Nonparametric Off-Policy Policy Gradient

In this section we introduce a nonparametric Bellman equation with a closed form solution, which carries the dependency from the policy's parameters. We derive the gradient of the solution, and discuss the properties of the proposed estimator.

3.1 A Nonparametric Bellman Equation

Nonparametric Bellman equations have been developed in a number of prior works. [22]–[24] used nonparametric models such as Gaussian Processes for approximate dynamic programming. [25] have shown that these methods differ mainly in their use of regularization. [26] provided a Bellman equation using kernel density-estimation and a general overview on nonparametric dynamic programming. In contrast to prior work, our formulation preserves the dependency on the policy, enabling the computation of the policy gradient in closed-form. Moreover, we upper-bound the bias of the Nadaraya-Watson kernel regression to prove that our value function estimate is consistent w.r.t. the classical Bellman equation under smoothness assumptions. We focus on the maximization of the average return in the infinite horizon case formulated as a starting state objective $\int_{\mathbf{s}} \mu_0(\mathbf{s})V_\pi(\mathbf{s})\,\mathrm{d}\mathbf{s}$ [19].

Definition 1. *The discounted infinite-horizon objective is defined by* $J_\pi = \int \mu_0(\mathbf{s})V_\pi(\mathbf{s})\,\mathrm{d}\mathbf{s}$. *Under a stochastic policy the objective is subject to the Bellman equation constraint*

$$V_\pi(\mathbf{s}) = \int_{\mathcal{A}} \pi_\theta(\mathbf{a}|\mathbf{s})\left(r(\mathbf{s}, \mathbf{a}) + \gamma\int_{\mathcal{S}} V_\pi(\mathbf{s}')p(\mathbf{s}'|\mathbf{s}, \mathbf{a})\,\mathrm{d}\mathbf{s}'\right)\mathrm{d}\mathbf{a}, \tag{2}$$

while in the case of a deterministic policy the constraint is given as

$$V_\pi(\mathbf{s}) = r(\mathbf{s}, \pi_\theta(\mathbf{s})) + \gamma \int_{\mathcal{S}} V_\pi(\mathbf{s}') p(\mathbf{s}'|\mathbf{s}, \pi_\theta(\mathbf{s}))\, \mathrm{d}\mathbf{s}'.$$

Maximizing the objective in Definition 1 analytically is not possible, excluding special cases such as under linear-quadratic assumptions [27], or finite state-action space. Extracting an expression for the gradient of J_π w.r.t. the policy parameters θ is also not straightforward given the infinite set of possibly non-convex constraints represented in the recursion over V_π. Nevertheless, it is possible to transform the constraints in Definition 1 to a finite set of linear constraints via nonparametric modeling, thus leading to an expression of the value function with simple algebraic manipulation [26].

3.1.1 Nonparametric Modeling.

Assume a set of n observations $D \equiv \{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i, \gamma_i\}_{i=1}^n$ sampled from interaction with an environment, with $\mathbf{s}_i, \mathbf{a}_i \sim \beta(\cdot,\cdot)$, $\mathbf{s}'_i \sim p(\cdot|\mathbf{s}_i, \mathbf{a}_i)$, $r_i \sim R(\mathbf{s}_i, \mathbf{a}_i)$ and $\gamma \sim \gamma(\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i)$. We define the kernels $\psi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$, $\varphi : \mathcal{A} \times \mathcal{A} \to \mathbb{R}^+$ and $\phi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$, as normalized, symmetric and positive definite functions with bandwidths $\mathbf{h}_\psi, \mathbf{h}_\varphi, \mathbf{h}_\phi$ respectively. We define $\psi_i(\mathbf{s}) = \psi(\mathbf{s}, \mathbf{s}_i)$, $\varphi_i(\mathbf{a}) = \varphi(\mathbf{a}, \mathbf{a}_i)$, and $\phi_i(\mathbf{s}) = \phi(\mathbf{s}, \mathbf{s'}_i)$. Following [26], the mean reward $r(\mathbf{s}, \mathbf{a})$ and the transition conditional $p(\mathbf{s}'|\mathbf{s}, \mathbf{a})$ are approximated by the Nadaraya-Watson regression [28], [29] and kernel density estimation, respectively

$$\begin{aligned}
\hat{r}(\mathbf{s}, \mathbf{a}) &\coloneqq \frac{\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a}) r_i}{\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a})}, \\
\hat{p}(\mathbf{s}'|\mathbf{s}, \mathbf{a}) &\coloneqq \frac{\sum_{i=1}^n \phi_i(\mathbf{s}')\psi_i(\mathbf{s})\varphi_i(\mathbf{a})}{\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a})}, \\
\hat{\gamma}(\mathbf{s}, \mathbf{a}, \mathbf{s}') &\coloneqq \frac{\sum_{i=1}^n \gamma_i \psi_i(\mathbf{s})\varphi_i(\mathbf{a})\phi_i(\mathbf{s}')}{\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a})\phi_i(\mathbf{s}')}
\end{aligned}$$

and, therefore, by the product of $\hat{p}$ and $\hat{\gamma}$ we obtain

$$\begin{aligned}
\hat{p}_\gamma(\mathbf{s}'|\mathbf{s}, \mathbf{a}) &\coloneqq \hat{p}(\mathbf{s}'|\mathbf{s}, \mathbf{a})\hat{\gamma}(\mathbf{s}, \mathbf{a}, \mathbf{s}') \\
&= \frac{\sum_{i=1}^n \gamma_i \psi_i(\mathbf{s})\varphi_i(\mathbf{a})\phi_i(\mathbf{s}')}{\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a})}.
\end{aligned}$$

Inserting the reward and transition kernels into the Bellman Equation for the stochastic policy case we obtain the nonparametric Bellman equation (NPBE)

$$\begin{aligned}
\hat{V}_\pi(\mathbf{s}) &= \int_{\mathcal{A}} \pi_\theta(\mathbf{a}|\mathbf{s}) \left(\hat{r}(\mathbf{s}, \mathbf{a}) + \int_{\mathcal{S}} \hat{V}_\pi(\mathbf{s}') \hat{p}_\gamma(\mathbf{s}'|\mathbf{s}, \mathbf{a})\, \mathrm{d}\mathbf{s}' \right) \mathrm{d}\mathbf{a} \\
&= \sum_i \int_{\mathcal{A}} \frac{\pi_\theta(\mathbf{a}|\mathbf{s})\psi_i(\mathbf{s})\varphi_i(\mathbf{a})}{\sum_j \psi_j(\mathbf{s})\varphi_j(\mathbf{a})}\, \mathrm{d}\mathbf{a} \\
&\quad \times \left(r_i + \gamma_i \int_{\mathcal{S}} \phi_i(\mathbf{s}') \hat{V}_\pi(\mathbf{s}')\, \mathrm{d}\mathbf{s}' \right). \qquad (3)
\end{aligned}$$

Equation (3) can be conveniently expressed in matrix form by introducing the vector of responsibilities $\varepsilon_i(\mathbf{s}) = \int \pi_\theta(\mathbf{a}|\mathbf{s})\psi_i(\mathbf{s})\varphi_i(\mathbf{a}) / \sum_j \psi_j(\mathbf{s})\varphi_j(\mathbf{a})\, \mathrm{d}\mathbf{a}$, which assigns each state $\mathbf{s}$ a weight relative to its distance to a sample i under the current policy.

Definition 2. *The nonparametric Bellman equation on the dataset D is formally defined as*

$$\hat{V}_\pi(\mathbf{s}) = \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \left(\mathbf{r} + \int_{\mathcal{S}} \boldsymbol{\phi}_\gamma(\mathbf{s}') \hat{V}_\pi(\mathbf{s}')\, \mathrm{d}\mathbf{s}' \right), \qquad (4)$$

with $\boldsymbol{\phi}_\gamma(\mathbf{s}) = [\gamma_1\phi_1(\mathbf{s}), \ldots, \gamma_n\phi_n(\mathbf{s})]^\intercal$, $\mathbf{r} = [r_1, \ldots, r_n]^\intercal$,
$\boldsymbol{\varepsilon}_\pi(\mathbf{s}) = [\varepsilon_1^\pi(\mathbf{s}), \ldots, \varepsilon_n^\pi(\mathbf{s})]^\intercal$,

$$\varepsilon_i^\pi(\mathbf{s}) = \begin{cases} \int \pi_\theta(\mathbf{a}|\mathbf{s}) \frac{\psi_i(\mathbf{s})\varphi_i(\mathbf{a})}{\sum_j \psi_j(\mathbf{s})\varphi_j(\mathbf{a})}\, \mathrm{d}\mathbf{a} & \text{if } \pi \text{ is stochastic} \\ \frac{\psi_i(\mathbf{s})\varphi_i(\pi_\theta(\mathbf{s}))}{\sum_j \psi_j(\mathbf{s})\varphi_j(\pi_\theta(\mathbf{s}))} & \text{otherwise.} \end{cases}$$

From Equation (4) we deduce that the value function must be of the form $\boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s})\mathbf{q}_\pi$, indicating that it can also be seen as a form of Nadaraya-Watson kernel regression,

$$\boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s})\mathbf{q}_\pi = \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \left(\mathbf{r} + \int_{\mathcal{S}} \boldsymbol{\phi}_\gamma(\mathbf{s}') \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}')\mathbf{q}_\pi\, \mathrm{d}\mathbf{s}' \right). \qquad (5)$$

Notice that, trivially, every $\mathbf{q}_\pi$ which satisfies

$$\mathbf{q}_\pi = \mathbf{r} + \int_{\mathcal{S}} \boldsymbol{\phi}_\gamma(\mathbf{s}') \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}')\mathbf{q}_\pi\, \mathrm{d}\mathbf{s}' \qquad (6)$$

also satisfies Equation (5). Theorem 1 demonstrates that the algebraic solution of Equation (6) is the *only* solution of the nonparametric Bellman Equation (4).

Theorem 1. *The nonparametric Bellman equation has a unique fixed-point solution*

$$\hat{V}_\pi^*(\mathbf{s}) \coloneqq \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \boldsymbol{\Lambda}_\pi^{-1} \mathbf{r},$$

with $\boldsymbol{\Lambda}_\pi \coloneqq I - \hat{\mathbf{P}}_\pi^\gamma$ *and* $\hat{\mathbf{P}}_\pi^\gamma \coloneqq \int_{\mathcal{S}} \boldsymbol{\phi}_\gamma(\mathbf{s}') \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}')\, \mathrm{d}\mathbf{s}'$, *where* $\boldsymbol{\Lambda}_\pi$ *is always invertible since* $\hat{\mathbf{P}}^{\pi,\gamma}$ *is a strictly sub-stochastic matrix (Frobenius' Theorem). The statement is valid also for $n \to \infty$, provided bounded R.*

Proof of Theorem 1 is provided in the supplementary material.

3.2 Nonparametric Gradient Estimation

With the closed-form solution of $\hat{V}_\pi^*(\mathbf{s})$ from Theorem 1, it is possible to compute the analytical gradient of J_π w.r.t. the policy parameters

$$\begin{aligned}
\nabla_\theta \hat{V}_\pi^*(\mathbf{s}) &= \left(\frac{\partial}{\partial\theta} \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \right) \boldsymbol{\Lambda}_\pi^{-1}\mathbf{r} + \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \frac{\partial}{\partial\theta} \boldsymbol{\Lambda}_\pi^{-1}\mathbf{r} \\
&= \underbrace{\left(\frac{\partial}{\partial\theta} \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \right) \boldsymbol{\Lambda}_\pi^{-1}\mathbf{r}}_{\mathrm{A}} \\
&\quad + \underbrace{\boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \boldsymbol{\Lambda}_\pi^{-1} \left(\frac{\partial}{\partial\theta} \hat{\mathbf{P}}_\pi^\gamma \right) \boldsymbol{\Lambda}_\pi^{-1}\mathbf{r}}_{\mathrm{B}}. \qquad (7)
\end{aligned}$$

Substituting the result of Equation (7) into the return specified in Definition 1, introducing $\boldsymbol{\varepsilon}_{\pi,0}^\intercal \coloneqq \int \mu_0(\mathbf{s}) \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s})\, \mathrm{d}\mathbf{s}$, $\mathbf{q}_\pi = \boldsymbol{\Lambda}_\pi^{-1}\mathbf{r}$, and $\boldsymbol{\mu}_\pi = \boldsymbol{\Lambda}_\pi^{-\intercal} \boldsymbol{\varepsilon}_{\pi,0}$ we obtain

$$\nabla_\theta \hat{J}_\pi = \left(\frac{\partial}{\partial\theta} \boldsymbol{\varepsilon}_{\pi,0}^\intercal \right) \mathbf{q}_\pi + \boldsymbol{\mu}_\pi^\intercal \left(\frac{\partial}{\partial\theta} \hat{\mathbf{P}}_\pi^\gamma \right) \mathbf{q}_\pi, \qquad (8)$$

where $\mathbf{q}_\pi$ and $\boldsymbol{\mu}_\pi$ can be estimated via conjugate gradient to avoid the inversion of $\boldsymbol{\Lambda}_\pi$.

Algorithm 1 Nonparametric Off-Policy Policy Gradient

input: dataset $\{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i, \mathrm{t}_i\}_{i=1}^n$ where t_i indicates a terminal state, a policy π_θ and kernels ψ, ϕ, φ respectively for state, action and next-state.
while not converged **do**
  Compute $\boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s})$ as in Definition 2 and $\boldsymbol{\varepsilon}_{\pi,0}^\intercal := \int \mu_0(\mathbf{s})\boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s})\,\mathrm{d}\mathbf{s}$.
  Estimate $\hat{\mathbf{P}}_\pi$ as defined in Theorem 1 using MC ($\phi(\mathbf{s})$ is a distribution).
  Set each row i of $\hat{\mathbf{P}}_\pi$ to 0 if t_i is a terminal state.
  Solve $\mathbf{r} = \boldsymbol{\Lambda}_\pi \mathbf{q}_\pi$ and $\boldsymbol{\varepsilon}_{\pi,0} = \boldsymbol{\Lambda}_\pi^\intercal \boldsymbol{\mu}_\pi$ for $\mathbf{q}_\pi$ and $\boldsymbol{\mu}_\pi$ using conjugate gradient.
  Update θ using Equation (8).
end while

The terms A and B in Equation (7) correspond to the terms in Equation (10). In contrast to semi-gradient actor-critic methods, where the gradient bias is affected by both the critic bias and the semi-gradient approximation [8], [12], our estimate is the *full gradient* and the only source of bias is introduced by the estimation of $\hat{V}_\pi$, which we analyze in Section 3.3. The term $\boldsymbol{\mu}_\pi$ can be interpreted as the support of the state-distribution as it satisfies $\boldsymbol{\mu}_\pi^\intercal = \boldsymbol{\varepsilon}_{\pi,0}^\intercal + \boldsymbol{\mu}_\pi^\intercal \hat{\mathbf{P}}_\pi^\gamma$. In Section 5, more specifically in Figure 6, we empirically show that $\varepsilon_\pi^\intercal(\mathbf{s})\boldsymbol{\mu}_\pi$ provides an estimate of the state distribution over the whole state-space. Implementation-wise, the quantities $\boldsymbol{\varepsilon}_{\pi,0}^\intercal$ and $\hat{\mathbf{P}}_{i,j}^\pi$ are estimated via Monte-Carlo sampling, which is unbiased but computationally demanding, or using other techniques such as unscented transform or numerical quadrature. The matrix $\hat{\mathbf{P}}_\pi^\gamma$ is of dimension $n \times n$, which can be memory-demanding. In practice, we notice that the matrix is often sparse. By taking advantage of conjugate gradient and sparsification we are able to achieve computational complexity of $\mathcal{O}(n^2)$ per policy update and memory complexity of $\mathcal{O}(n)$. A schematic of our implementation is summarized in Algorithm 1.

3.3 A theoretical Analysis

Nonparametric estimates of the transition dynamics and reward enjoy favorable properties for an off-policy learning setting. A well-known asymptotic behavior of the Nadaraya-Watson kernel regression,

$$\mathbb{E}\left[\lim_{n\to\infty} \hat{f}_n(x)\right] - f(x) \approx h_n^2 \left(\frac{1}{2} f''(x) + \frac{f'(x)\beta'(x)}{\beta(x)}\right) \int u^2 K(u)\,\mathrm{d}u,$$

shows how the bias is related to the regression function $f(x)$, as well as to the samples' distribution $\beta(x)$ [30], [31]. However, this asymptotic behavior is valid only for infinitesimal bandwidth, infinite samples ($h \to 0, nh \to \infty$) and requires the knowledge of the regression function and of the sampling distribution.

In a recent work, we propose an upper bound of the bias that is also valid for finite bandwidths [32]. We show under some Lipschitz conditions that the bound of the Nadaraya-Watson kernel regression bias does not depend on the samples' distribution, which is a desirable property in off-policy scenarios. The analysis is extended to multidimensional input space. For clarity of exposition, we report the main result in its simplest formulation, and later use it to infer the bound of the NPBE bias.

Theorem 2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a Lipschitz continuous function with constant L_f. Assume a set $\{\mathbf{x}_i, y_i\}_{i=1}^n$ of i.i.d. samples from a log-Lipschitz distribution β with a Lipschitz constant L_β. Assume $y_i = f(\mathbf{x}_i) + \epsilon_i$, where $f : \mathbb{R}^d \to \mathbb{R}$ and ϵ_i is i.i.d. and zero-mean. The bias of the Nadaraya-Watson kernel regression with Gaussian kernels in the limit of infinite samples $n \to \infty$ is bounded by*

$$\left|\mathbb{E}\left[\lim_{n\to\infty} \hat{f}_n(\mathbf{x})\right] - f(\mathbf{x})\right| \leq \frac{L_f \sum_{k=1}^d \mathbf{h}_k \left(\prod_{i\neq k}^d \chi_i\right) \left(\frac{1}{\sqrt{2\pi}} + \frac{L_\beta \mathbf{h}_k}{2}\chi_k\right)}{\prod_{i=1}^d e^{\frac{L_\beta^2 h_i^2}{2}} \left(1 - \operatorname{erf}\left(\frac{\mathbf{h}_i L_\beta}{\sqrt{2}}\right)\right)},$$

where

$$\chi_i = e^{\frac{L_\beta^2 \mathbf{h}_i^2}{2}} \left(1 + \operatorname{erf}\left(\frac{\mathbf{h}_i L_\beta}{\sqrt{2}}\right)\right),$$

$\mathbf{h} > 0 \in R^d$ is the vector of bandwidths and erf *is the error function.*

Building on Theorem 2 we show that the solution of the NPBE is consistent with the solution of the true Bellman equation. Moreover, although the bound is not affected directly by $\beta(\mathbf{s})$, a smoother sample distribution $\beta(\mathbf{s})$ plays favorably in the bias term (a low L_β is preferred).

Theorem 3. *Consider an arbitrary MDP $\mathcal{M}$ with a transition density p and a stochastic reward function $R(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \epsilon_{\mathbf{s},\mathbf{a}}$, where $r(\mathbf{s}, \mathbf{a})$ is a Lipschitz continuous function with L_R constant and $\epsilon_{\mathbf{s},\mathbf{a}}$ denotes zero-mean noise. Assume $|R(\mathbf{s}, \mathbf{a})| \leq R_{max}$ and a dataset D_n sampled from a log-Lipschitz distribution β defined over the state-action space with Lipschitz constant L_β. Let V_D be the unique solution of a nonparametric Bellman equation with Gaussian kernels ψ, φ, ϕ with positive bandwidths $\mathbf{h}_\psi, \mathbf{h}_\varphi, \mathbf{h}_\phi$ defined over the dataset $\lim_{n\to\infty} D_n$. Assume V_D to be Lipschitz continuous with constant L_V. The bias of such estimator is bounded by*

$$\left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \frac{1}{1-\gamma}\left(A_{Bias} + \gamma L_V \sum_{k=1}^{d_s} \frac{h_{\phi,k}}{\sqrt{2\pi}}\right), \qquad (9)$$

where $\overline{V}(\mathbf{s}) = \mathbb{E}_D[V_D(\mathbf{s})]$, A_{Bias} is the bound of the bias provided in Theorem 2 with $L_f = L_R$, $\mathbf{h} = [\mathbf{h}_\psi, \mathbf{h}_\varphi]$, $d = d_s + d_a$ and $V^*(\mathbf{s})$ is the fixed point of the ordinary Bellman equation. [2]

Theorem 3 shows that the value function provided by Theorem 1 is consistent when the bandwidth approaches infinitesimal vales. Moreover, it is interesting to notice that the error can be decomposed in A_{Bias}, which is the bias component dependent on the reward's approximation, and the remaining term that depends on the smoothness of the value function and the bandwidth of ϕ, which can be read as the error of the transition's model.

2. Complete proofs of the theorems and precise definitions can be found in the supplementary material.

The independence from the sampling distribution suggests that, under these assumptions, nonparametric estimation is particularly suited for off-policy setting, as the bias is not affected by different behavioral policies. More in detail, the bound shows that smoother reward functions, state-transitions and sample distributions play favorably against the estimation bias.

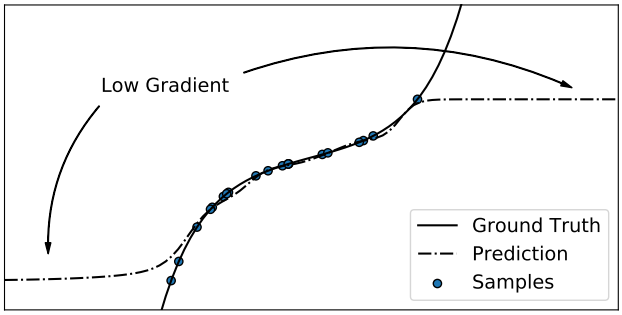


Fig. 2: The classic effect (known as boundary-bias) of the Nadaraya-Watson regression predicting a constant function in low-density regions is beneficial in our case, as it prevents the policy from moving in those areas as the gradient gets close to zero.

3.3.1 Blocking Effect

Very commonly, in order to prevent harmful policy optimization, the policy is constrained to stay close to the data [33], to avoid taking large steps [3], [34] or to circumvent large variance in the estimation [35], [36]. These techniques prevent incorrect and dangerous estimates of the gradient. Even if we do not include any explicit constraint of this kind, the Nadaraya-Watson kernel regression automatically discourages policy improvements towards low-data areas. In fact, as depicted in Figure 2, the Nadaraya-Watson kernel regression, tends to predict a constant function in low density regions. Usually, this characteristic is regarded as an issue, as it causes the so-called boundary-bias. In our case, this effect turns out to be beneficial, as it constrains the policy to stay close to the samples, where the model is more correct.

4 Related Work

Off-policy policy gradient estimation can be divided in three different techniques: semi-gradient approaches, importance sampling correction, and model based estimation. Semi-gradient approaches omit one term in the gradient computation, which causes an estimation bias [8]. The importance sampling correction, although unbiased, suffers from high variance, which makes it often unpractical [37]. Model based approaches rely on a model's estimation, and optimize the policy following this model. However, the model's error propagates in the number of steps, adding also a significant source of bias [16].

4.1 Semi-Gradient Approaches

The off-policy policy gradient theorem was the first proposed off-policy actor-critic algorithm [9]. Since then, it has been used by the vast majority of state-of-the-art off-policy algorithms [2], [3], [10], [11], [34]. Nonetheless, it is important to note that this theorem and its successors, introduce two approximations to the original policy gradient theorem [19]. First, semi-gradient approaches consider a modified discounted infinite-horizon return objective $\hat{J}_\pi = \int \rho_\beta(\mathbf{s}) V_\pi(\mathbf{s}) \, \mathrm{d}\mathbf{s}$, where $\rho_\beta(\mathbf{s})$ is state distribution under the behavioral policy π_β. Second, the gradient estimate is modified to be

$$\begin{aligned}
\nabla_\theta \hat{J}_\pi &= \nabla_\theta \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) V_\pi(\mathbf{s}) \, \mathrm{d}\mathbf{s} \\
&= \nabla_\theta \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) \int_{\mathcal{A}} \pi_\theta(\mathbf{a}|\mathbf{s}) Q_\pi(\mathbf{s},\mathbf{a}) \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{s} \\
&= \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) \int_{\mathcal{A}} \underbrace{\nabla_\theta \pi_\theta(\mathbf{a}|\mathbf{s}) Q_\pi(\mathbf{s},\mathbf{a})}_{\text{A}} \\
&\quad + \underbrace{\pi_\theta(\mathbf{a}|\mathbf{s}) \nabla_\theta Q_\pi(\mathbf{s},\mathbf{a})}_{\text{B}} \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{s} \qquad (10) \\
&\approx \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) \int_{\mathcal{A}} \nabla_\theta \pi_\theta(\mathbf{a}|\mathbf{s}) Q_\pi(\mathbf{s},\mathbf{a}) \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{s},
\end{aligned}$$

where the term B related to the derivative of Q_π is ignored. The authors provide a proof that this biased gradient, or *semi-gradient*, still converges to the optimal policy in a tabular setting [8], [9]. However, further approximation (e.g., given by the critic and by finite sample size), might disallow the convergence to a satisfactory solution. It might be deceiving to think that these algorithms are in fact off-policy: although they work correctly sampling from the replay memory (which discards *the oldest* samples), they have shown to fail with samples generated via a completely different process [8]. For this reason, we don't consider semi-gradient approaches to be promising for off-policy optimization.

4.2 Importance Sampling Approaches

One way to obtain an unbiased estimate of the policy gradient in an off-policy scenario is to re-weight every trajectory via importance sampling [13]–[15]. An example of the gradient estimation via G(PO)MDP [38] with importance sampling is given by

$$\nabla_\theta J_\pi = \mathbb{E}\left[\sum_{t=0}^{T-1} \rho_t \left(\prod_{j=0}^{t-1} \gamma_j \right) r_t \sum_{i=0}^{t} \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \right], \quad (11)$$

where $\rho_t = \prod_{z=0}^{t} \pi_\theta(\mathbf{a}_z|\mathbf{s}_z) / \pi_\beta(\mathbf{a}_z|\mathbf{s}_z)$. This technique applies only to stochastic policies and requires the knowledge of the behavioral policy π_β. Moreover, Equation (11) shows that path-wise importance sampling (PWIS) needs a trajectory-based dataset, since it needs to keep track of the past in the correction term ρ_t, hence introducing more restrictions on its applicability. Additionally, importance sampling suffers from high variance [37]. Recent works have helped to make PWIS more reliable. For example, [8], building on the emphatic weighting framework [39], proposed a trade-off between PWIS and semi-gradient approaches. Another possibility consists in restricting the gradient improvement to a safe-region, where the importance sampling does not suffer from too high variance [35]. Another interesting line of research is to estimate the importance

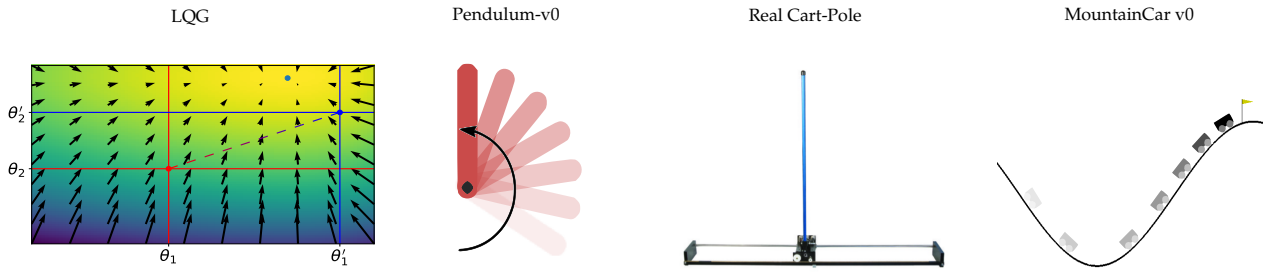


Fig. 3: Benchmarking tasks.

sampling correction on a state-distribution level instead of on the classic trajectory level [40]–[42]. We note that, all these promising algorithms have been applied on low-dimensional problems, as importance sampling suffers from the curse of dimensionality. Our proposed solution suffers also from this problem, but we believe that our approach, as well as these recent advances in importance sampling correction, might serve for further well-theoretically-defined off-policy techniques.

4.3 Model Based

Another natural approach which comes to mind when thinking about off-policy optimization, is to use a learned model of the transition. This model allows to generate new samples and therefore to optimize the policy potentially off-line. The proclaimed efficiency of model-based techniques relies on the fact that they allow off-policy optimization. However, model-based techniques are also problematic: the model error propagates in the Bellman recursion (or in the number of steps, if we prefer), often resulting in bad policy improvements. PILCO [16] aims to optimize the policy using probabilistic inference based on Gaussian Processes to model the estimation's uncertainty. However, it works on a finite horizon setting, is restricted to unimodal state-transitions, and a particular shape of reward. PETS [36], an improved version of PILCO, builds a probabilitstic model using a bootstrapped ensemble of neural-networks, and propagates the state-distribution using particles. This method, still requires a finite horizon. Furthermore, PETS does not make use of a parametrized policy, but instead a model predictive control. The controller requires multiple neural network evaluations, which can result in an issue when interacting with a real-time system. Our method, in contrast, works on the infinite-horizon setting, and the usage of a parametrized policy is more suitable for real-time operations.

5 Empirical Evaluation

In this section, we analyze our method. Therefore, we divide our experiments in two logical sections: the analysis of the gradient, and the analysis of the policy optimization using a gradient ascent technique. The analysis of the gradient comprises an empirical evaluation of the bias, the variance and the gradient direction w.r.t. the ground truth, in relation to some quantities such as the size of the dataset or its degree of "off-policiness". In the policy optimization analysis, instead, we aim to both compare the sample efficiency of our method in comparison to state-of-the-art policy gradient algorithms, and to study its applicability to unstructured and human-demonstrated datasets.

5.1 Benchmarking Tasks

In the following, we give a brief description of the tasks involved in the empirical analysis.

5.1.1 *Linear Quadratic Gaussian Controller*

A very classical control problem consists of linear dynamics, quadratic reward and Gaussian noise. The main advantage of this control problem relies in the fact that it is fully solvable in closed-form, using the Riccati equations, which makes it appropriate for verifying the correctness of our algorithm. In our specific scenario, we have a policy encoded with two parameters for illustration purposes. The LQG is defined as

$$\begin{aligned} &\max_{\boldsymbol{\theta}} \sum_{t=0}^{\infty} \gamma^t r_t \\ &\text{s.t.} \quad \mathbf{s}_{t+1} = A\mathbf{s}_t + B\mathbf{a}_t; \quad r_t = -\mathbf{s}_t^\intercal Q \mathbf{s}_t - \mathbf{a}_t^\intercal R \mathbf{a}_t \\ &\qquad \mathbf{a}_{t+1} = \Theta \mathbf{s}_t + \Sigma \epsilon_t; \quad \epsilon_t \sim \mathcal{N}(0, I), \end{aligned}$$

with A, B, Q, R, Σ diagonal matrix and $\Theta = \mathrm{diag}(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are considered the policy's parameters. In the stochastic policy experiments, $\pi_\theta(\mathbf{a}|\mathbf{s}) = \mathcal{N}(\mathbf{a}|\Theta\mathbf{s}; \Sigma)$, while for the deterministic case $\Sigma = \mathbf{0}$ and $\pi_\theta(\mathbf{s}) = \Theta\mathbf{s}$. For further details, please refer to the supplementary material.

5.1.2 *OpenAI Pendulum-v0*

The OpenAI Pendulum-v0 [43] is a popular benchmark in reinforcement learning. It simulates a simple under-actuated inverted-pendulum. The goal is to swing the pendulum until it reaches the top position, and then to keep it stable. The state of the system is fully described by the angle of the pendulum ω and its angular velocity $\dot{\omega}$. The applied torque $\tau \in [-2, 2]$ corresponds to the agent's action. One of the advantages of such a system, is that its well-known value function is two-dimensional.

5.1.3 *Quanser Cart-pole*

The cart-pole is another classical task in reinforcement learning. It consists of an actuated cart moving on a track, to which a pole is attached. The goal is to actuate the cart in a way to balance the pole in the top position. Differently from the inverted pendulum, the system has a further degree of

complexity, and the state space requires the position on the track x, the velocity of the cart $\dot{x}$, the angle of the pendulum ω and its angular velocity $\dot{\omega}$.

5.1.4 OpenAI Mountain-Car

The mountain-car (also known as car-on-hill), consists on an under-powered car that must reach the top of a hill. The car is placed in the valley connecting two hills. In order to reach the goal position, it must first go in opposite direction in order to gain momentum. Its state is described by the x-position of the car, and by its velocity $\dot{x}$. The episodes terminate when the car reaches the goal. In contrast to the swing-up pendulum, which is hardly controllable by a human-being, this car system is ideal to provide human-demonstrated data.

5.2 Algorithms Used for Comparisons

In order to provide an analysis of the gradient, we compare our algorithm against G(PO)MDP with importance sampling, and with off-line DPG. Instead of using the naïve form of G(PO)MDP with importance sampling, which suffers from high variance, we used the normalized importance sampling [44], [45] (which introduces some bias but drastically reduces the variance), and the generalized baselines [46] (which also introduce some bias, as they are estimated from the same dataset). The off-line version of DPG, suffers from three different sources of bias: the semi-gradient, the critic approximation and the improper use of the discounted state distribution [47], [48]. In order to mitigate these issues and focus more on the semi-gradient contribution to the bias, we provide an oracle Q-function (we denote this version as DPG+Q). For the policy improvement, instead, we compare to more sophisticated and recent deep reinforcement learning techniques, such as TD3 [12] and SAC [2]. A full list of the algorithms used in the comparisons with a brief description is available in Table 1.

5.3 Analysis of the Gradient

We want to compare the bias and variance of our gradient estimator w.r.t. the already discussed classical estimators. Therefore, we use the LQG setting described in Section 5.1.1,

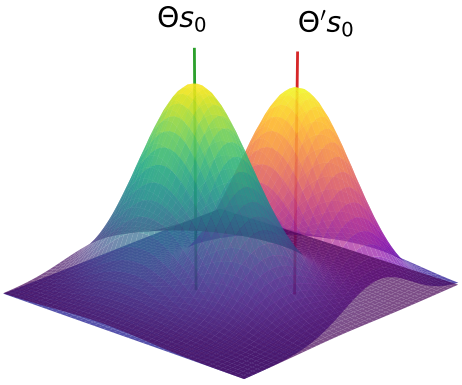


Fig. 4: Evaluated in the initial state, the optimization policy having parameters θ_1, θ_2 and the behavioral policy having parameters θ_1', θ_2' exhibit a fair distance in probability space.

Acronym	Description	Typology
NOPG-D	Our method with deterministic policy.	NOPG
NOPG-S	Our method with stochastic policy.	
G(PO)MDP+N	G(PO)MDP with normalized importance sampling.	PWIS
G(PO)MDP+BN	G(PO)MDP with normalized importance sampling and generalized baselines.	
DPG+Q	Offline version of the deterministic policy gradient theorem with an oracle for the Q-function.	SG
DDPG-Off	Offline version of the deep deterministic policy gradient theorem.	
DDPG-On	Classic version of DDPG.	
TD3	Improved version of DDPG.	
SAC	Classic version of SAC.	

TABLE 1: Acronyms used in the paper to refer to practical implementation of the algorithms.

which allows us to compute the true gradient. Our goal is to estimate the gradient w.r.t. the policy π_θ diagonal parameters θ_1, θ_2, while sampling from a policy which is a linear combination of Θ and Θ'. The hyper-parameter α determines the mixing between the two parameters. When $\alpha = 1$ the behavioral policy will have parameters Θ', while when $\alpha = 0$ the dataset will be sampled using Θ. In Figure 4, we can visualize the difference of the two policies with parameters Θ and Θ'. Although not completely disjoint, they are fairly far in the probability space, especially if we take into account that such distance propagates in the length of the trajectories.

5.3.1 Sample Analysis

We want to study how the bias, the variance and the direction of the estimated gradient vary w.r.t. the dataset's size. We are particularly interested in the off-policy strategy for sampling, and in this set of experiments we will use constant $\alpha = 0.5$. Figure 5a depicts these quantities w.r.t. the number of collected samples. As expected, a general trend for all algorithms is that with a higher number of samples we are able to reduce the variance. The importance sampling based G(PO)MDP algorithms eventually obtain a low bias as well. Remarkably, NOPG has significantly both lower bias and variance, and its gradient direction is also more accurate w.r.t. the G(PO)MDP algorithms (note the different scales of the y-axis). Between DPG+Q and NOPG there is no sensible difference, but we should take into account the already-mentioned advantage of DPG+Q to have access to the true Q-function.

5.3.2 Off-Policy Analysis

We want to estimate the bias and the variance w.r.t. different degrees of "off-policiness" α. We want to highlight that in the deterministic experiment the behavioral policy remains stochastic. This is needed to ensure the stochastic generation of datasets, which is essential to estimate the bias and the variance of the estimator. As depicted in Figure 5, the variance in importance sampling based techniques tends to increase when the dataset is off-policy. On the contrary, NOPG seems to be more subject to an increase of bias. This trend is also noticeable in DPG+Q, where the component of

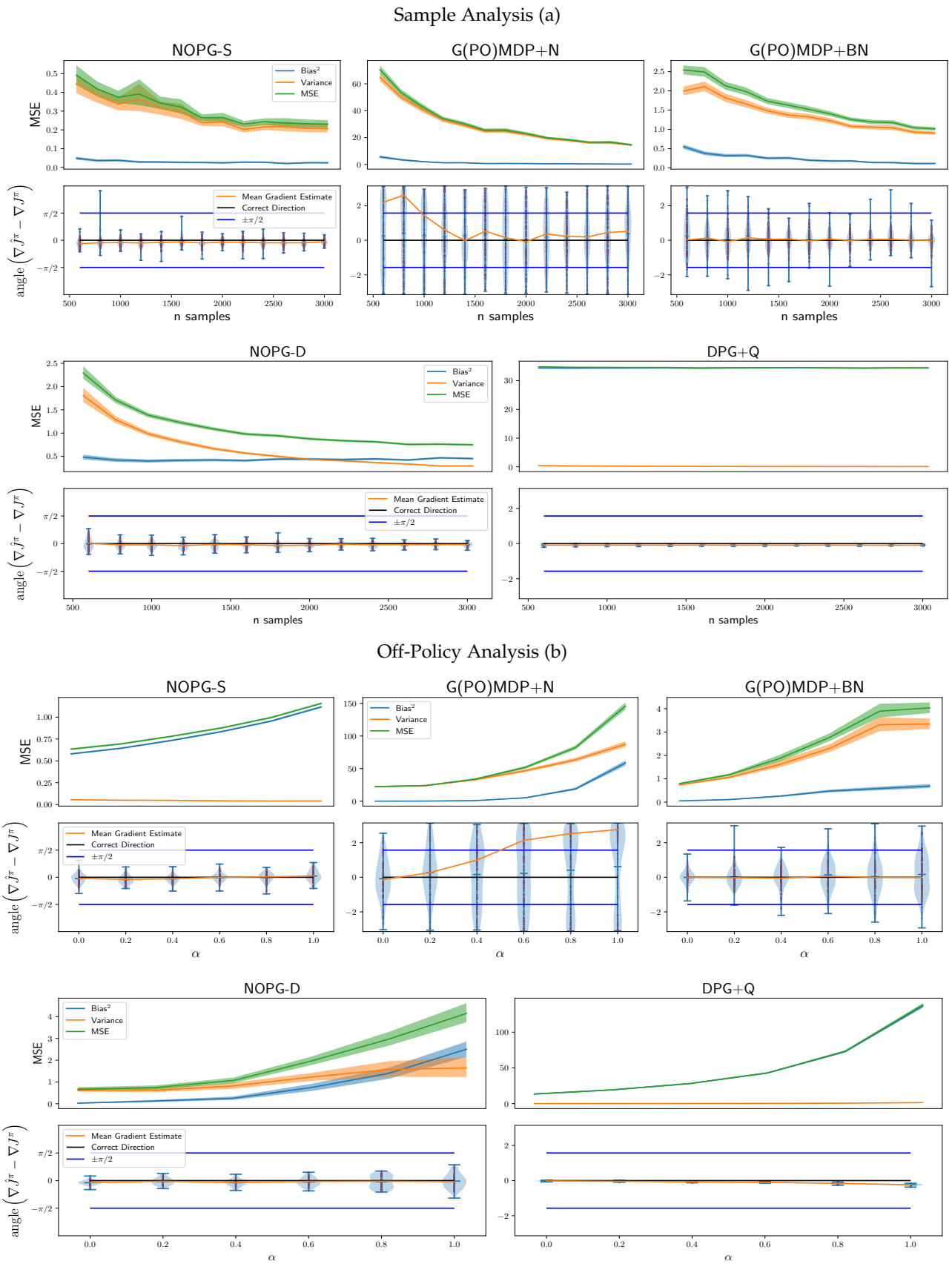


Fig. 5: Bias, variance, MSE and gradient direction analysis. The MSE plots are equipped with a 95% interval using bootstrapping techniques. The direction analysis plots describe the distribution of angle between the estimates and the ground truth gradient. NOPG exhibits favorable bias, variance and gradient direction compared to PWIS and semi-gradient.

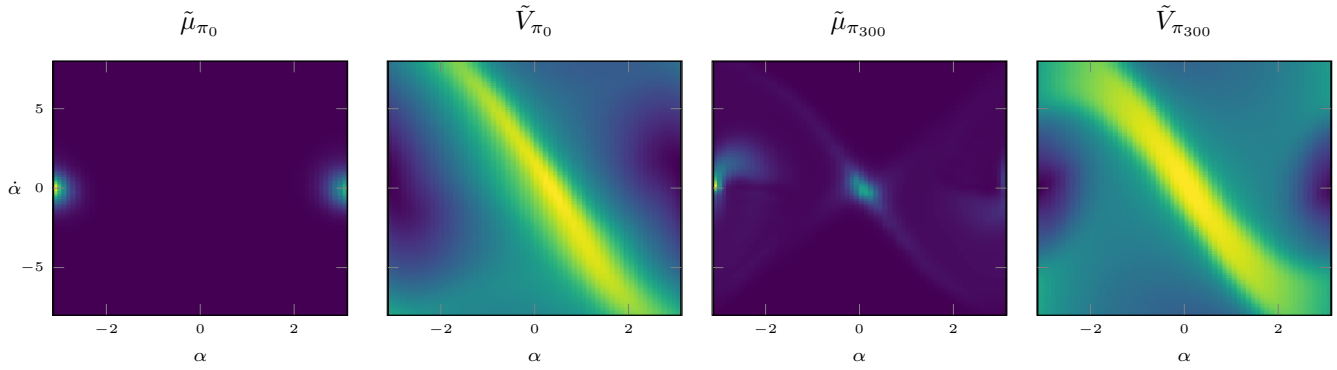


Fig. 6: A phase portrait of the state distribution $\tilde{\mu}_\pi$ and value function $\tilde{V}_\pi$ estimated in the swing-up pendulum task with NOPG-D. Green corresponds to higher values. The two leftmost figures show the estimates before any policy improvement, while the two rightmost show them after 300 offline updates of NOPG-D. Notice that the algorithm finds a very good approximation of the optimal value function and is able to predict that the system will reach the goal state ($(\alpha, \dot{\alpha}) = (0, 0)$).

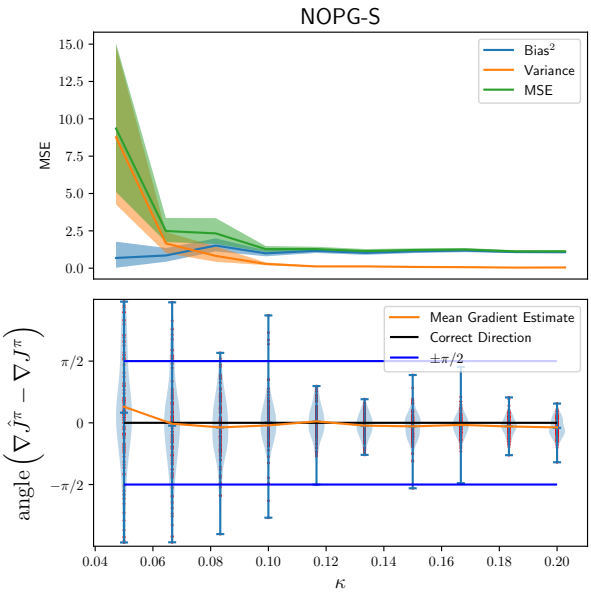


Fig. 7: A lower bandwidth corresponds to higher variance, while higher bandwidth increases the bias up to a plateau.

the bias is the one playing a major role in the mean squared error. The gradient direction of NOPG seems however unbiased, while DPG+Q has a slight bias but remarkably less variance (note the different scales of the y-axis). We remark that DPG+Q uses an oracle for the Q-function, which supposedly results in lower variance and bias[3]. The positive bias of DPG+Q in the on-policy case ($\alpha = 0$) is caused by the improper use of discounting. In general, NOPG shows a decrease in bias and variance in order of magnitudes when compared to the other algorithms.

3. Furthermore, we suspect that the particular choice of a LQG task tends to mitigate the problems of DPG, as the fast convergence to a stationary distribution due to the stable attractor, united with the improper discounting, results in a coincidental correction of the state-distribution.

5.3.3 *Bandwidth Analysis*

In the previous analysis, we kept the bandwidth's parameters of our algorithm fixed, even though a dynamic adaptation of this parameter w.r.t. the size of the dataset might have improved the bias/variance trade-off. We are now interested in studying how the bandwidth impacts the gradient estimation. For this purpose, we generated datasets of 1000 samples with $\alpha = 0.5$. We set all the bandwidths of state, action and next state, for each dimension equal to κ. From Figure 7 we evince that a lower bandwidth corresponds to a higher variance, while a larger bandwidth approaches a constant bias and the variance tends to zero. This result is in line with the theory.

5.4 Policy Improvement

In the previous section, we analyzed the statistical properties of our estimator. Conversely, in this section, we use the NOPG estimate to fully optimize the policy. At the current state, NOPG is a batch algorithm, meaning that it receives as input a set of data, and it outputs an optimized policy, without any interaction with the environment. We study the sample efficiency of the overall algorithm. We compare it with both other batch and online algorithms. Please notice that online algorithms, such as DDPG-On, TD3 and SAC, can acquire more valuable samples during the optimization process. Therefore, in a direct comparison, batch algorithms are in disadvantage.

5.4.1 *Uniform Grid*

In this experiment we analyze the performance of NOPG under a uniformly sampled dataset, since, as the theory suggests, this scenario should yield the least biased estimate of NOPG. We generate datasets from a grid over the state-action space of the pendulum environment with different granularities. We test our algorithm by optimizing a policy encoded with a neural-network for a fixed amount of iterations. The policy is composed of a single hidden layer with 50 neurons and ReLU activations. This configuration is fixed across all the different experiments and algorithms for the remainder of this document. The resulting policy is evaluated on trajectories of 500 steps starting from the

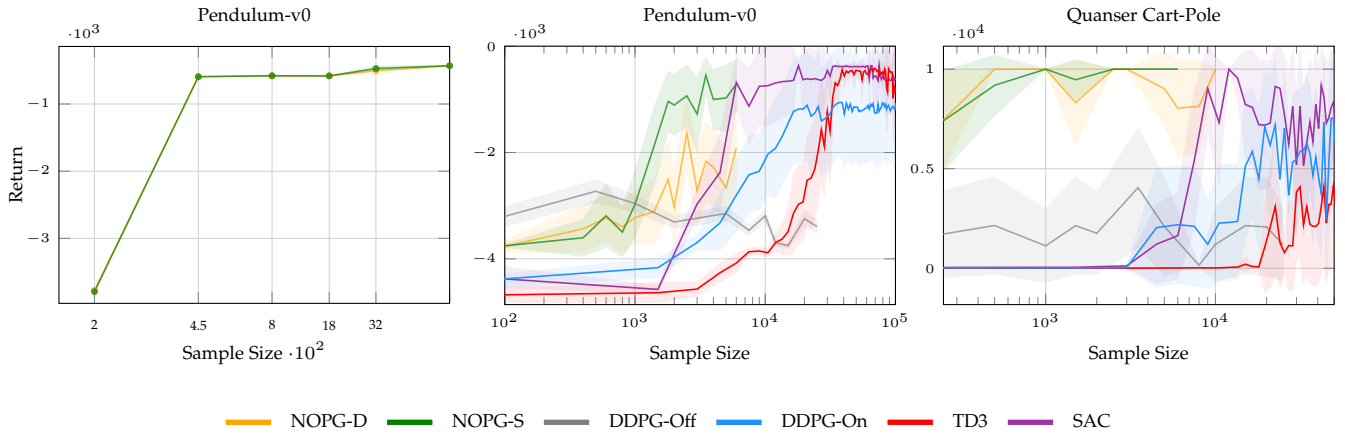


Fig. 8: Comparison of NOPG in its deterministic and stochastic versions to state-of-the-art algorithms on continuous control tasks: Swing-Up Pendulum with **uniform grid** sampling (left), Swing-Up Pendulum with the **random agent** (center-left) and the Cart-Pole stabilization (center-right). The figures depict the mean and 95% confidence interval over 10 trials. NOPG outperforms the baselines w.r.t the sample complexity. **Note the log-scale along the x-axis**. The right most picture shows the real cart-pole platform from Quanser.

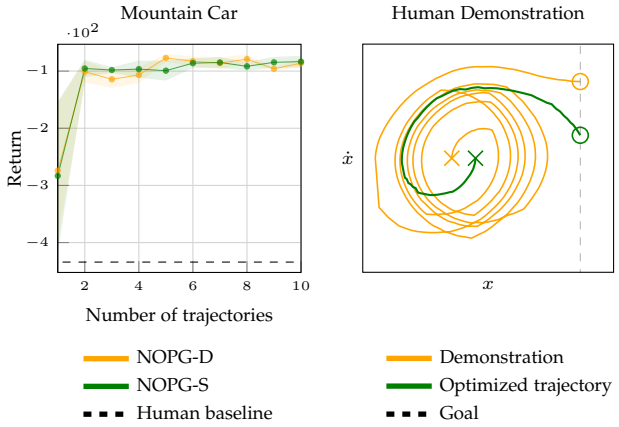


Fig. 9: With a small amount of data NOPG is able to reach a policy that surpasses the human demonstrator (dashed line) in the mountain car environment. Depicted are the mean and 95% confidence over 10 trials (left). An example of a human-demonstrated trajectory and the relative optimized version obtained with NOPG (right). Although the human trajectories in the dataset are suboptimal, NOPG converges to an optimal solution (right).

bottom position. The leftmost plot in Figure 8, depicts the performance against different dataset sizes, showing that NOPG is able to solve the task with 450 samples. Figure 6 is an example of the value function and state distribution estimates of NOPG-D at the beginning and after 300 optimization steps. The ability to predict the state-distribution is particularly interesting for robotics, as it is possible to predict in advance whether the policy will move towards dangerous states. Note that this experiment is not applicable to PWIS, as it does not admit non-trajectory-based data.

5.4.2 *Online Setting*

In contrast to the uniform grid experiment, here we collect the datasets using trajectories from a random agent in the pendulum and the cart-pole environments. In the pendulum task, the trajectories are generated starting from the upright position and applying a policy composed of a mixture of two Gaussians. The policies are evaluated starting from the bottom position with an episode length of 500 steps. The datasets used in the cart-pole experiments are collected using a uniform policy starting from the upright position until the end of the episode, which occurs when the absolute value of the angle θ surpasses 3 deg. The optimization policy is evaluated for 10^4 steps. The reward is $r_t = \cos\theta_t$. Since θ is defined as 0 in the top-right position, a return of 10^4 indicates an optimal policy behavior.

We analyze the sample efficiency by testing NOPG and DDPG-Off in an offline fashion with pre-collected samples, on a different number of trajectories. In addition, we provide the learning curve of DDPG-On, TD3 and SAC using the implementation in Mushroom [49]. For a fixed size of the dataset, we optimize DDPG-Off and NOPG for a fixed number of steps. Since DDPG-Off exhibits an unstable learning, we select the best policy obtained during the learning process. For NOPG, instead, we select the policy from the last optimization step. The two rightmost plots in Figure 8 highlight that our algorithm has superior sample efficiency by more than one order of magnitude (note the log-scale on the x-axis).

To validate the resulting policy learned in simulation, we apply the final learned controller on a real Quanser cart-pole, and observe a successful stabilizing behavior as can be seen in the supplementary video.

5.4.3 *Human Demonstrated Data*

In robotics, learning from human demonstrations is crucial in order to obtain better sample efficiency and to avoid dangerous policies. This experiment is designed to showcase the ability of our algorithm to deal with such demonstrations without the need for explicit knowledge of the underlying behavioral policy. The experiment is executed in a completely offline fashion after collecting the human dataset, i.e., without any further interaction with the environment. This setting is different from the classical imitation learning

and subsequent optimization [50]. As an environment we choose the continuous mountain car task from OpenAI. We provide 10 demonstrations recorded by a human operator and assigned a reward of -1 to every step. A demonstration ends when the human operator surpasses the limit of 500 steps, or arrives at the goal position. The human operator explicitly provides sub-optimal trajectories, as we are interested in analyzing whether NOPG is able to take advantage of the human demonstrations to learn a better policy than that of the human, without any further interaction with the environment. To obtain a sample analysis, we evaluate NOPG on randomly selected sub-sets of the trajectories from the human demonstrations. Figure 9 shows the average performance as a function of the number of demonstrations as well as an example of a human-demonstrated trajectory. Notice that both NOPG-S and NOPG-D manage to learn a policy that surpasses the human operator's performance and reach the optimal policy with two demonstrated trajectories.

6 Conclusion and Future Work

In this paper, we presented and analyzed an off-policy gradient technique *Nonparametric Off-policy Policy Gradient* (NOPG) [17]. Our estimator overcomes the main issues of the techniques of off-policy gradient estimation. On the one hand, in contrast to semi-gradient approaches, it delivers a full-gradient estimate; and on the other hand, it avoids the high variance of importance sampling, by phrasing the problem with nonparametric techniques. The empirical analysis clearly showed a better gradient estimate in terms of bias, variance, and direction. Our experiments also showed that our method has high sample efficiency and that our algorithm can be behavioral-agnostic and cope with unstructured data.

However, our algorithm, which is built on nonparametric techniques, suffers from the curse of dimensionality. Furthermore, it currently does not account for the exploration problem, which is important to avoid local optima. As a future work, we will study a parametric approximation of the Bellman equation, which similarly to NOPG allows for a full-gradient estimate, scales better with the number of samples, and with the help of Bayesian techniques we will tackle the problem of safe exploration.

Acknowledgments

The research is financially supported by the Bosch-Forschungsstiftung program and the European Union's Horizon 2020 research and innovation program under grant agreement #640554 (SKILLS4ROBOTS).

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. [Online]. Available: http://www.nature.com/articles/nature14236

[2] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceeding of the 35th International Conference on Machine Learning*, 2018, pp. 1856–1865.

[3] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust Region Policy Optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1889–1897.

[4] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-Based Batch Mode Reinforcement Learning," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 503–556, 2005. [Online]. Available: http://www.jmlr.org/papers/v6/ernst05a.html

[5] M. Riedmiller, "Neural Fitted Q Iteration – First Experiences with a Data Efficient Neural Reinforcement Learning Method," in *European Conference of Machine Learning*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 317–328.

[6] L. Baird, "Residual Algorithms: Reinforcement Learning with Function Approximation," *Machine Learning Proceedings*, pp. 30–37, 1995.

[7] T. Lu, D. Schuurmans, and C. Boutilier, "Non-Delusional Q-learning and Value-Iteration," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, pp. 9949–9959. [Online]. Available: http://papers.nips.cc/paper/8200-non-delusional-q-learning-and-value-iteration.pdf

[8] E. Imani, E. Graves, and M. White, "An Off-Policy Policy Gradient Theorem Using Emphatic Weightings," in *Advances in Neural Information Processing Systems*, 2018, pp. 96–106.

[9] T. Degris, M. White, and R. S. Sutton, "Off-Policy Actor-Critic," *arXiv:1205.4839 [cs]*, May 2012, arXiv: 1205.4839. [Online]. Available: http://arxiv.org/abs/1205.4839

[10] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *Proceedings of the 31 st International Conference on Machine Learning*, 2014.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," in *International Conference on Learning Representations*, 2016, arXiv: 1509.02971. [Online]. Available: http://arxiv.org/abs/1509.02971

[12] S. Fujimoto, D. Meger, and D. Precup, "Off-Policy Deep Reinforcement Learning without Exploration," in *Proceeding of the 36th International Conference on Machine Learning*, 2019, pp. 2052–2062. [Online]. Available: http://proceedings.mlr.press/v97/fujimoto19a/fujimoto19a.pdf

[13] C. R. Shelton, "Policy Improvement for POMDPs Using Normalized Importance Sampling," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'01. Morgan Kaufmann Publishers Inc., 2001, pp. 496–503, event-place: Seattle, Washington. [Online]. Available: http://dl.acm.org/citation.cfm?id=2074022.2074083

[14] N. Meuleau, L. Peshkin, and K.-E. Kim, "Exploration in Gradient-Based Reinforcement Learning," Massachusetts Institute of Technology, Tech. Rep., 2001. [Online]. Available: https://dspace.mit.edu/handle/1721.1/6076

[15] L. Peshkin and C. R. Shelton, "Learning from Scarce Experience," in *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, arXiv: cs/0204043. [Online]. Available: http://arxiv.org/abs/cs/0204043

[16] M. P. Deisenroth and C. E. Rasmussen, "PILCO: A Model-based and Data-efficient Approach to Policy Search," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. Omnipress, 2011, pp. 465–472, event-place: Bellevue, Washington, USA. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104482.3104541

[17] S. Tosatto, J. Carvalho, H. Abdulsamad, and J. Peters, "A Nonparametric Off-Policy Policy Gradient," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, S. Chiappa and R. Calandra, Eds., Palermo, Italy, 2020.

[18] M. White, "Unifying Task Specification in Reinforcement Learning," in *Proceedings of the 34th International Conference on Machine Learning*. JMLR. org, 2017, pp. 3742–3750.

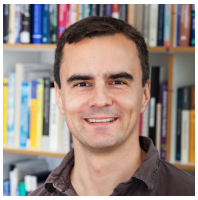
[19] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Advances in Neural Information Processing Systems*, 2000, pp. 1057–1063.

[20] R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[21] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992. [Online]. Available: https://doi.org/10.1007/BF00992698
[22] D. Ormoneit and S. Sen, "Kernel-Based Reinforcement Learning," *Machine Learning*, vol. 49, no. 2, pp. 161–178, 2002. [Online]. Available: https://doi.org/10.1023/A:1017928328829
[23] X. Xu, D. Hu, and X. Lu, "Kernel-Based Least Squares Policy Iteration for Reinforcement Learning," *IEEE Transactions on Neural Networks*, vol. 18, no. 4, pp. 973–992, 2007.
[24] Y. Engel, S. Mannor, and R. Meir, "Reinforcement Learning with Gaussian Processes," in *Proceedings of the 22nd International Conference On Machine Learning*. ACM, 2005, pp. 201–208.
[25] G. Taylor and R. Parr, "Kernelized Value Function Approximation for Reinforcement Learning," in *Proceedings of the 26th International Conference on Machine Learning*, ser. ICML '09. ACM, 2009, pp. 1017–1024, event-place: Montreal, Quebec, Canada. [Online]. Available: http://doi.acm.org/10.1145/1553374.1553504
[26] O. B. Kroemer and J. R. Peters, "A Non-Parametric Approach to Dynamic Programming," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2011, pp. 1719–1727. [Online]. Available: http://papers.nips.cc/paper/4182-a-non-parametric-approach-to-dynamic-programming.pdf
[27] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, Jun. 2017, google-Books-ID: 7NUoDwAAQBAJ.
[28] E. A. Nadaraya, "On Estimating Regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.
[29] G. S. Watson, "Smooth Regression Analysis," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.
[30] J. Fan, "Design-Adaptive Nonparametric Regression," *Journal of the American Statistical Association*, vol. 87, no. 420, pp. 998–1004, 1992.
[31] L. Wasserman, *All of Nonparametric Statistics*. Springer, 2006. [Online]. Available: https://books.google.it/books?hl=it&lr=&id=MRFlzQfRg7UC&oi=fnd&pg=PA2&dq=wasserman+2006+all&ots=SPSQp53XJz&sig=R9JPan0NnS8GkezXCj85U2ndFmc#v=onepage&q=wasserman%202006%20all&f=false
[32] S. Tosatto, R. Akrour, and J. Peters, "An Upper Bound of the Bias of Nadaraya-Watson Kernel Regression under Lipschitz Assumptions," *arXiv preprint arXiv:2001.10972*, 2020.
[33] J. Peters, K. Mulling, and Y. Altun, "Relative Entropy Policy Search," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.
[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.
[35] A. M. Metelli, M. Papini, F. Faccio, and M. Restelli, "Policy Optimization via Importance Sampling," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, pp. 5442–5454. [Online]. Available: http://papers.nips.cc/paper/7789-policy-optimization-via-importance-sampling.pdf
[36] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, pp. 4754–4765.
[37] A. B. Owen, *Monte Carlo Theory, Methods and Examples*, 2013.
[38] J. Baxter and P. L. Bartlett, "Infinite-Horizon Policy-Gradient Estimation," *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, 2001.
[39] R. S. Sutton, A. R. Mahmood, and M. White, "An Emphatic Approach to the Problem of Off-Policy Temporal-Difference Learning," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2603–2631, 2016, publisher: JMLR. org.
[40] Q. Liu, L. Li, Z. Tang, and D. Zhou, "Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation," in *Advances in Neural Information Processing Systems*, 2018, pp. 5356–5366.
[41] Y. Liu, A. Swaminathan, A. Agarwal, and E. Brunskill, "Off-Policy Policy Gradient with State Distribution Correction," *arXiv:1904.08473*, 2019, arXiv: 1904.08473. [Online]. Available: http://arxiv.org/abs/1904.08473
[42] O. Nachum, B. Dai, I. Kostrikov, Y. Chow, L. Li, and D. Schuurmans, "AlgaeDICE: Policy Gradient from Arbitrary Experience," *arXiv:1912.02074v1*, 2019.
[43] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv:1606.01540*, 2016, arXiv: 1606.01540. [Online]. Available: http://arxiv.org/abs/1606.01540
[44] C. R. Shelton, "Policy Improvement for POMDPs Using Normalized Importance Sampling," *arXiv preprint arXiv:1301.2310*, 2013.
[45] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016, vol. 10.
[46] T. Jie and P. Abbeel, "On a Connection Between Importance Sampling and the Likelihood Ratio Policy Gradient," in *Advances in Neural Information Processing Systems*, 2010, pp. 1000–1008.
[47] P. Thomas, "Bias in Natural Actor-Critic Algorithms," in *International Conference on Machine Learning*, 2014, pp. 441–448.
[48] C. Nota and P. S. Thomas, "Is the Policy Gradient a Gradient?" in *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2020.
[49] C. D'Eramo, D. Tateo, A. Bonarini, M. Restelli, and J. Peters, *MushroomRL: Simplifying Reinforcement Learning Research*, 2020, publication Title: arXiv preprint arXiv:2001.01102. [Online]. Available: https://github.com/MushroomRL/mushroom-rl
[50] J. Kober and J. R. Peters, "Policy Search for Motor Primitives in Robotics," in *Advances in Neural Information Processing Systems*, 2009, pp. 849–856.

Samuele Tosatto received his M.Sc. degree in Software Engineering from the Polytechnic University of Milan in 2017. Currently, he is pursuing his Ph.D. at the Intelligent Autonomous Systems Group at the Computer Science Department of the Technical University of Darmstadt. His research interests center around reinforcement learning with a specific focus on its application to robotic systems.

João Carvalho is currently a Ph.D. student at the Intelligent Autonomous Systems group of the Technical University of Darmstadt. Previously, he completed a M.Sc. degree in Computer Science from the Albert-Ludwigs-Universität Freiburg, and studied Electrical and Computer Engineering at the Instituto Superior Técnico of the University of Lisbon. His research is focused on devising learning algorithms targeted for control and robotics.



Jan Peters is a full professor (W3) for Intelligent Autonomous Systems at the Computer Science Department of the Technical University of Darmstadt, and at the same time a senior research scientist and group leader at the Max-Planck Institute for Intelligent Systems, where he heads the interdepartmental Robot Learning Group. Jan Peters has received the Dick Volz Best 2007 US Ph.D. Thesis Runner-Up Award, the Robotics: Science & Systems - Early Career Spotlight, the INNS Young Investigator Award, and the IEEE Robotics & Automation Society's Early Career Award as well as numerous best paper awards. In 2015, he received an ERC Starting Grant and in 2019, he was appointed as an IEEE Fellow.