A Nonparametric Off-Policy Policy Gradient

Samuele Tosatto[1] **João Carvalho**[1] **Hany Abdulsamad**[1] **Jan Peters**[1,2]
[1]Technische Universität Darmstadt
[2]Max Planck Institute for Intelligent Systems
{tosatto, carvalho, abdulsamad, peters}@ias.tu-darmstadt.de

Abstract

Reinforcement learning (RL) algorithms still suffer from high sample complexity despite outstanding recent successes. The need for intensive interactions with the environment is especially observed in many widely popular policy gradient algorithms that perform updates using on-policy samples. The price of such inefficiency becomes evident in real world scenarios such as interaction-driven robot learning, where the success of RL has been rather limited. We address this issue by building on the general sample efficiency of off-policy algorithms. With nonparametric regression and density estimation methods we construct a *nonparametric Bellman equation* in a principled manner, which allows us to obtain closed-form estimates of the value function, and to analytically express the *full* policy gradient. We provide a theoretical analysis of our estimate to show that it is consistent under mild smoothness assumptions and empirically show that our approach has better sample efficiency than state-of-the-art policy gradient methods.

1 Introduction

Reinforcement learning has made overwhelming progress in recent years (Mnih et al., 2015; Haarnoja et al., 2018; Schulman et al., 2015). However, the vast majority of reinforcement learning approaches are on-policy algorithms with limited applicability to real world scenarios, due to high sample complexity. 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 safe interaction with the environment. These two properties are of high importance when developing algorithms for real robots. However, classical off-policy algorithms like Q-learning with function approximation and fitted Q-iteration (Ernst et al., 2005; Riedmiller, 2005) are not guaranteed to converge (Baird, 1995; Lu et al., 2018), and allow only discrete actions. More recent semi-gradient[1] off-policy techniques, like Off-PAC (Degris et al., 2012) and DDPG (Silver et al., 2014; Lillicrap et al., 2016) often perform sub-optimally, especially when the collected data is strongly off-policy, due to the biased semi-gradient update (Fujimoto et al., 2019). Off-policy algorithms based on importance sampling (Shelton, 2001; Meuleau et al., 2001; Peshkin & Shelton, 2002)deliver an unbiased estimate of the gradient but suffer from high variance and are generally only applicable with stochastic policies. Moreover, they re-

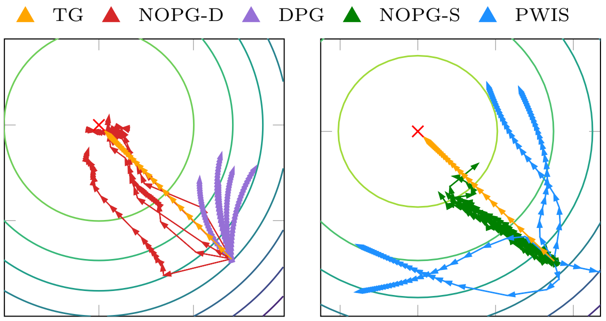

Figure 1: Example showing the bias of offline-DPG (left) and the variance of PWIS-G(PO)MDP (right) in the policy-parameter space of a 2d-LQR setting. Both algorithms diverge while they move away from the "on-policy" region. Our method in its deterministic and stochastic versions, NOPG-D and NOPG-S, shows better approximations of the true gradient (TG).

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

[1] We adopt the terminology from Imani et al. (2018).

quire the full knowledge of the behavioral policy, making them unsuitable when data stems from a human demonstrator. Additionally, model-based approaches like PILCO (Deisenroth & Rasmussen, 2011) 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), a full-gradient estimate based on the closed-form solution of a nonparametric Bellman equation. Furthermore, we avoid the high variance of importance sampling techniques and allow for the use of human demonstrations. Figure 1 qualitatively compares the gradient estimate of NOPG compared to that of semi-parametric approaches (DPG) and path-wise importance sampling (PWIS) techniques. Furthermore, unlike other nonparametric approaches like PILCO, our approach allows for multimodal state-transitions, and can handle the infinite-horizon setting. For empirical validation, we evaluate our approach on a number of classical control tasks. The results highlight the sample efficiency of our approach.

2 Notation and Background

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 $(\mathcal{S}, \mathcal{A}, \gamma, P, R)$ where $\mathcal{S} \equiv \mathbb{R}^{d_s}$ is the state space, $\mathcal{A} \equiv \mathbb{R}^{d_a}$ the action space and $\gamma \in [0, 1)$ the discount factor. 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 state-action pair $(\mathbf{s}, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}$ is drawn from a distribution $R(\mathbf{s}, \mathbf{a})$ with mean value $r(\mathbf{s}, \mathbf{a})$. The policy π, parameterized by θ, is a stochastic or deterministic mapping from $\mathcal{S}$ onto $\mathcal{A}$. Our objective is to maximize the expected return

$$J_\pi = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]. \tag{1}$$

Following Sutton et al. (2000), we define $\mu_\pi(\mathbf{s}) = \sum_{t=0}^{\infty} \gamma^t p(\mathbf{s}_t|\mathbf{s}_0, \pi)$ as the state-visitation function induced by the policy π_θ. A state-action value function $Q_\pi(\mathbf{s}, \mathbf{a})$ maps the state-action pair onto $\mathbb{R}$ and represents the expected discounted cumulative return following the policy π_θ. The state value function V_π is the expectation of Q_π under π_θ.

Policy Gradient Theorem. Objective (1) can be 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 (Sutton et al., 2000). In a direct episodic on-policy setting, the expected return Q_π can be estimated under the current state-action visitation $\mu_\pi(\mathbf{s})\pi_\theta(\mathbf{a}|\mathbf{s})$ via Monte-Carlo episodic rollouts of the current policy (Williams, 1992). This technique, however, may require excessive interactions with the environment, since the return and the expectations need to be approximated after each policy update.

Off-Policy Semi-Gradient. The off-policy policy gradient theorem was the first proposed off-policy actor-critic algorithm (Degris et al., 2012). Since then it has inspired a series of state-of-the-art off-policy algorithms (Silver et al., 2014; Lillicrap et al., 2016). Nonetheless, its important to note that this theorem and its successors, introduce two approximations to the original policy gradient theorem. Firstly, 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 the stationary state distribution under the behavioral policy π_β. Secondly, the gradient estimate is modified to be

$$\begin{aligned}
\nabla_\theta \hat{J}_\pi &= \nabla_\theta \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) V_\pi(\mathbf{s}) \, \mathrm{d}\mathbf{s} \\
&= \nabla_\theta \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) \int_{\mathcal{A}} \pi_\theta(\mathbf{a}|\mathbf{s}) Q_\pi(\mathbf{s}, \mathbf{a}) \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{s} \\
&= \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) \int_{\mathcal{A}} \underbrace{\nabla_\theta \pi_\theta(\mathbf{a}|\mathbf{s}) Q_\pi(\mathbf{s}, \mathbf{a})}_{\text{A}} \\
&\qquad + \underbrace{\pi_\theta(\mathbf{a}|\mathbf{s}) \nabla_\theta Q_\pi(\mathbf{s}, \mathbf{a})}_{\text{B}} \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{s} \qquad (2) \\
&\approx \int_{\mathcal{S}} \rho_\beta(\mathbf{s}) \int_{\mathcal{A}} \nabla_\theta \pi_\theta(\mathbf{a}|\mathbf{s}) Q_\pi(\mathbf{s}, \mathbf{a}) \, \mathrm{d}\mathbf{a} \, \mathrm{d}\mathbf{s},
\end{aligned}$$

where the term B related to the derivative of Q_π is ignored. In all fairness, the authors provide a proof that this biased gradient, or *semi-gradient*, still converges to the optimal policy in a discrete MDP setting (Degris et al., 2012; Imani et al., 2018).

Path-Wise Importance Sampling (PWIS). 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 (Meuleau et al., 2001; Shelton, 2001; Peshkin & Shelton, 2002). An example of the gradient estimation via G(PO)MDP with importance sampling is given by

$$\nabla_\theta J_\pi = \mathbb{E}\left[\sum_{t=0}^{T-1} \rho_t Q_\pi(\mathbf{s}_t, \mathbf{a}_t) \nabla_\theta \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t)\right], \tag{3}$$

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 (3) shows that PWIS needs a trajectory-based dataset, since it needs to keep track of the past in the correction term ρ_z, hence introducing more restrictions on its applicability. Additionally, importance sampling suffers from high variance (Owen, 2013), which grows multiplicatively in the number of steps (Equation 3). Despite these difficulties, many interesting recent advances have helped to make PWIS more reliable. For example, Imani et al. (2018) propose a trade-off between PWIS and semi-gradient approaches, Metelli et al. (2018) argue for the use of a surrogate objective which accounts for the variance of the estimate and Liu et al. (2018, 2019) apply importance sampling to the state distribution instead of the trajectories.

3 Nonparametric Off-Policy Policy Gradient

We introduce a new offline off-policy approach with a full-gradient estimate that does not suffer from the drawbacks of importance sampling and semi-gradient algorithms. Starting from a nonparametric Bellman equation, we derive an analytical expression of the gradient for deterministic and stochastic policies. Nonparametric Bellman equations have been developed in a number of prior works. Ormoneit & Sen (2002); Xu et al. (2007); Engel et al. (2005) used nonparametric models such as Gaussian processes for approximate dynamic programming. Taylor & Parr (2009) have shown that these methods differ mainly in their use of regularization. Kroemer & Peters (2011) provided a Bellman equation using kernel density-estimation and a general overview over 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.

Definition 1. *The discounted infinite-horizon objective is defined by $J_\pi = \int \mu_0(\mathbf{s})V_\pi(\mathbf{s})\,\mathrm{d}\mathbf{s}$, where μ_0 is the initial state distribution. Under a stochastic policy the objective is subject to the constraint*

$$V_\pi(\mathbf{s}) = \int_{\mathcal{A}} \pi_\theta(\mathbf{a}|\mathbf{s})\Bigg(r(\mathbf{s},\mathbf{a}) + \gamma \int_{\mathcal{S}} V_\pi(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\Bigg)\mathrm{d}\mathbf{a}, \qquad (4)$$

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 (Borrelli et al., 2017). 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 (Kroemer & Peters, 2011).

Nonparametric Modeling. Assume a set of n observations $D \equiv \{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s'}_i\}_{i=1}^{n}$ sampled from interaction with an environment, with $\mathbf{s}_i, \mathbf{a}_i \sim \beta(\cdot,\cdot)$, $\mathbf{s}_i' \sim p(\cdot|\mathbf{s}_i,\mathbf{a}_i)$ and $r_i \sim R(\mathbf{s}_i,\mathbf{a}_i)$. 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 $\mathbf{h}_\varphi, \mathbf{h}_\phi, \mathbf{h}_\psi$ 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 Kroemer & Peters (2011), 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 (Nadaraya, 1964; Watson, 1964) and kernel density estimation, respectively

$$\hat{r}(\mathbf{s},\mathbf{a}) := \frac{\sum_{i=1}^{n}\psi_i(\mathbf{s})\varphi_i(\mathbf{a})r_i}{\sum_{i=1}^{n}\psi_i(\mathbf{s})\varphi_i(\mathbf{a})}$$

$$p(\mathbf{s}'|\mathbf{s},\mathbf{a}) \approx \frac{\hat{p}(\mathbf{s}',\mathbf{a},\mathbf{s})}{\hat{p}(\mathbf{a},\mathbf{s})} := \hat{p}(\mathbf{s}'|\mathbf{s},\mathbf{a})$$

where $\hat{p}(\mathbf{s}',\mathbf{s},\mathbf{a}) = 1/n\sum_i \phi_i(\mathbf{s}')\psi_i(\mathbf{s})\varphi_i(\mathbf{a})$ and $\hat{p}(\mathbf{s},\mathbf{a}) = 1/n\sum_i \psi_i(\mathbf{s})\varphi_i(\mathbf{a})$.

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

$$\begin{aligned}\hat{V}_\pi(\mathbf{s}) &= \int_{\mathcal{A}} \pi_\theta(\mathbf{a}|\mathbf{s})\Bigg(\hat{r}(\mathbf{s},\mathbf{a}) + \gamma\int_{\mathcal{S}} \hat{V}_\pi(\mathbf{s}')\hat{p}(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\Bigg)\mathrm{d}\mathbf{a} \\ &= \sum_i \int_{\mathcal{A}} \frac{\pi_\theta(\mathbf{a}|\mathbf{s})\psi_i(\mathbf{s})\varphi_i(\mathbf{a})}{\sum_j \psi_j(\mathbf{s})\varphi_j(\mathbf{a})}\,\mathrm{d}\mathbf{a} \\ &\qquad \times \Bigg(r_i + \gamma\int_{\mathcal{S}} \phi_i(\mathbf{s}')\hat{V}_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}'\Bigg). \qquad (5)\end{aligned}$$

Equation (5) 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*

$$\hat{V}_\pi(\mathbf{s}) = \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}') \hat{V}_\pi(\mathbf{s}') \, \mathrm{d}\mathbf{s}' \right), \tag{6}$$

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

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

From Equation (6) 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,

$$\boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s})\mathbf{q}_\pi = \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}') \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \mathbf{q}_\pi \, \mathrm{d}\mathbf{s}' \right). \tag{7}$$

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

$$\mathbf{q}_\pi = \mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}') \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \mathbf{q}_\pi \, \mathrm{d}\mathbf{s}' \tag{8}$$

also satisfies Equation (7). Theorem 1 demonstrates that the algebraic solution of Equation (8) is the *only* solution of the nonparametric Bellman Equation (6).

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

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

with $\boldsymbol{\Lambda}_\pi := I - \gamma \hat{\mathbf{P}}_\pi$ *and* $\hat{\mathbf{P}}_{i,j}^\pi := \int \phi_i(\mathbf{s}') \varepsilon_j^\pi(\mathbf{s}') \, \mathrm{d}\mathbf{s}'$, *where* $\boldsymbol{\Lambda}_\pi$ *is always invertible since* $\hat{\mathbf{P}}_\pi$ *is a stochastic matrix and* $0 \leq \gamma < 1$.

Proof of Theorem 1 is provided in the supplementary material.

Policy Gradient. 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

$$\begin{aligned} \nabla_\theta \hat{V}_\pi^*(\mathbf{s}) &= \left(\frac{\partial}{\partial \theta} \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \right) \boldsymbol{\Lambda}_\pi^{-1} \mathbf{r} + \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \frac{\partial}{\partial \theta} \boldsymbol{\Lambda}_\pi^{-1} \mathbf{r} \\ &= \underbrace{\left(\frac{\partial}{\partial \theta} \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \right) \boldsymbol{\Lambda}_\pi^{-1} \mathbf{r}}_{\mathrm{A}} \\ &\quad \underbrace{+ \gamma \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \boldsymbol{\Lambda}_\pi^{-1} \left(\frac{\partial}{\partial \theta} \hat{\mathbf{P}}_\pi \right) \boldsymbol{\Lambda}_\pi^{-1} \mathbf{r}}_{\mathrm{B}}. \end{aligned} \tag{9}$$

Substituting the result of Equation (9) into the return specified in Definition 1, introducing $\boldsymbol{\varepsilon}_{\pi,0}^\intercal := \int \mu_0(\mathbf{s}) \boldsymbol{\varepsilon}_\pi^\intercal(\mathbf{s}) \, \mathrm{d}\mathbf{s}$, $\mathbf{q}_\pi = \boldsymbol{\Lambda}_\pi^{-1} \mathbf{r}$, and $\boldsymbol{\mu}_\pi = \boldsymbol{\Lambda}_\pi^{-\intercal} \boldsymbol{\varepsilon}_{\pi,0}$ we obtain

$$\nabla_\theta \hat{J}_\pi = \left(\frac{\partial}{\partial \theta} \boldsymbol{\varepsilon}_{\pi,0}^\intercal \right) \mathbf{q}_\pi + \gamma \boldsymbol{\mu}_\pi^\intercal \left(\frac{\partial}{\partial \theta} \hat{\mathbf{P}}_\pi \right) \mathbf{q}_\pi, \tag{10}$$

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 $\mathrm{t}_i$ indicates a terminal state, a policy $\pi_\theta$ and kernels $\psi, \phi, \varphi$ 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 $\mathrm{t}_i$ 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 $\theta$ using Equation (10).
end while
```

The terms A and B in Equation (9) correspond to the terms in Equation (2). In contrast to semi-gradient actor-critic methods, where the gradient bias is affected by both the critic bias and the semi-gradient approximation (Imani et al., 2018; Fujimoto et al., 2019), 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 4. The term $\boldsymbol{\mu}_\pi$ can be interpreted as the support of the state-distribution as it satisfies $\boldsymbol{\mu}_\pi^\intercal = \boldsymbol{\varepsilon}_{\pi,0}^\intercal + \gamma \boldsymbol{\mu}_\pi^\intercal \hat{\mathbf{P}}_\pi$. In Section 5, more specifically in Figure 3, 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 $\hat{\mathbf{P}}_\pi$ is of dimension $n \times n$, which can be memory-demanding. In practice, we notice that the matrix is often almost 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.

4 Error Analysis of Nonparametric Estimates

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 $\beta(x)$ (Fan, 1992; Wasserman, 2006). 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 (Tosatto et al., 2020). We show that under some Lipschitz conditions, 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*

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

where

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

$\mathbf{h} > 0 \in R^d$ *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 $|R(\mathbf{s},\mathbf{a})| \le R_{max}$ 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*

$$\left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \le \frac{1}{1-\gamma}\left(A_{Bias} + \gamma L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\right), \quad (11)$$

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

Theorem 3 shows that the value function provided by Theorem 1 is consistent. 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.

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.

5 Empirical Evaluation

For experimental validation we compare both the deterministic (NOPG-D) and stochastic (NOPG-S) versions of our algorithm to G(PO)MDP with PWIS (from here on PWIS). Additionally, we compare NOPG-D to state-of-the-art deterministic off-policy algorithms DPG and DDPG. In particular we want to address the following questions:

1. How do the bias and the variance compare to PWIS and semi-gradient approaches?
2. Does our method work in scenarios where PWIS is not applicable?
3. How is the sample efficiency of our methods compared to state-of-the-art off-policy approaches?

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

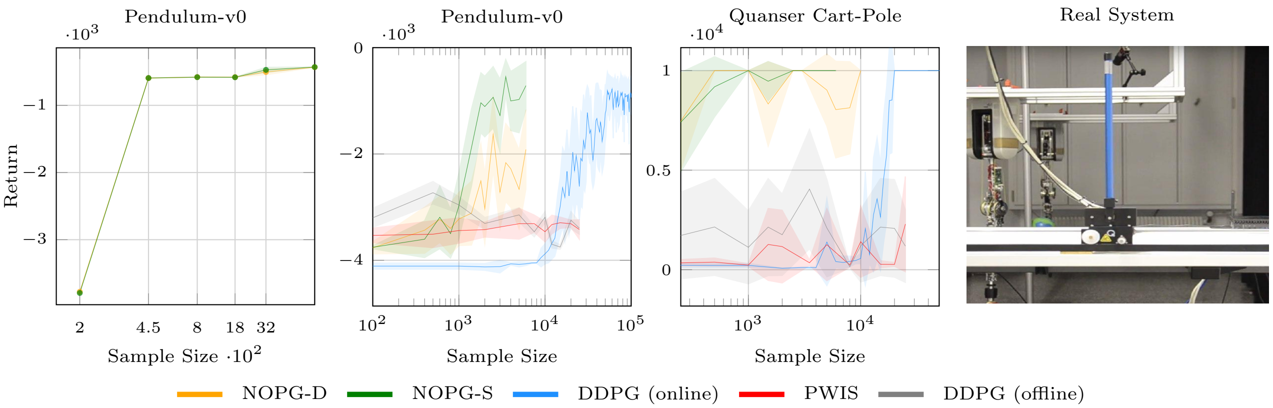

Figure 2: Comparison of NOPG in its deterministic and stochastic versions to state-of-the-art algorithms on continuous control tasks: Swing-Up Pendulum with **uniform grid** sampling (left), Swing-Up Pendulum with the **random agent** (center-left) and the Cart-Pole stabilization (center-right). The 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.

To answer the first question, we conduct an experiment on a 2-dimensional LQR problem, providing a graphical representation of the gradient updates using the mentioned algorithms. We address the second question by testing our algorithm on a uniform-grid dataset (i.e. no explicit trajectories) and on a test obtained from a human demonstrator. To test the sample efficiency we compare our methods against DDPG, offline DDPG, and PWIS on the swing-up pendulum and on the cart-pole stabilization[3]. The supplementary provides details of all hyper-parameters used, an implementation of NOPG and video of the final policy executed on a real cart-pole system.

5.1 Gradient Direction with LQR

We qualitatively demonstrate how the different gradient estimates work on a simple 2-dimensional problem. For this purpose we choose a linear-quadratic regulator setup, and use a linear policy encoded by the diagonal matrix $K = [[k_1, 0], [0, k_2]]$. Figure 1 illustrates experiments with deterministic and stochastic policies. In the experiment with deterministic policies we evaluate the performance of NOPG-D and offline-DPG over 5 datasets, each containing 100 trajectories of length 30. The experiment with stochastic policies compares the gradient estimates of NOPG-S and PWIS. Given that PWIS requires a single stochastic policy, we generated 5 datasets with 100 trajectories of length 30 from the interactions of a Gaussian policy with the environment. The results in Figure 1 show that PWIS suffers from high variance while DPG offers a biased estimate, which is consistent with our initial theoretical hypothesis. Moreover, it is interesting to observe how the error in DPG's biased estimate compounds after every iteration as the algorithm moves away from the initial "on-policy" region in the vicinity of the behavioral policy. NOPG on the other hand exhibits a more accurate gradient estimate in the off-policy region and with smaller variance when compared to PWIS.

5.2 Swing-Up Pendulum and Cart-Pole

The under-powered pendulum and the cart-pole are two classical control tasks often used in RL for empirical analysis. We use the OpenAI Gym framework (Brockman et al., 2016) for a pendulum simulation, and implement another environment that simulates the dynamics of a real cart-pole built by Quanser [4]

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 of 50 neurons with 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 bottom position. The leftmost plot in Figure 2, depicts the performance against different dataset sizes, showing that NOPG is

[3] The code of NOPG is available at `https://github.com/jacarvalho/nopg`.

[4] `https://www.quanser.com/products/linear-servo-base-unit-inverted-pendulum`

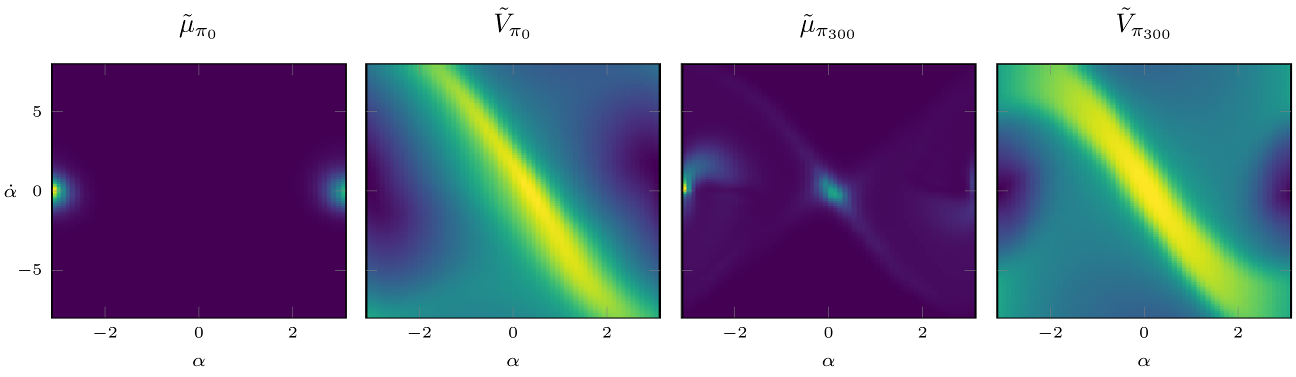

Figure 3: A phase portrait of the state distribution $\tilde{\mu}_\pi$ and value function $\tilde{V}_\pi$ estimated in the swing-up pendulum task with NOPG-D. Green corresponds to higher values. The two leftmost figures show the estimates before any policy improvement, while the two rightmost show them 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 ($(\alpha, \dot{\alpha}) = (0, 0)$).

able to solve the task with 450 samples. Figure 3 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.

Random Agent. 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 up-right position and applying a policy composed of a mixture of 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, PWIS and DDPG in an offline fashion with pre-collected samples, on different number of trajectories. In addition, we provide the learning curve with the classical online DDPG using the OpenAI Baselines implementation (Dhariwal et al., 2017).

We stress that, since offline DDPG and PWIS show an unstable learning curve, we always report the *best evaluation* obtained during the learning process, while with NOPG we report the last evaluation. The two center plots in Figure 2 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.

5.3 Mountain Car with Human Demonstrations

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 (Kober & Peters, 2009). 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 4 shows the average performance as a function of the number of demonstrations

	Human Demonstration	Unstructured Dataset	Deterministic Policies	Bias	Variance
Semi-Gradient	✓	✓	✓	high	low
PWIS	✗	✗	✗	low	high
NOPG	✓	✓	✓	low	low

Table 1: Applicability of off-policy algorithms. Our algorithm is applicable to a wider range of tasks in contrast to state-of-the-art techniques. NOPG is able to deal with human demonstrations and unstructured datasets by using either a stochastic or a deterministic policy, all while exhibiting lower bias and variance than its competitors.

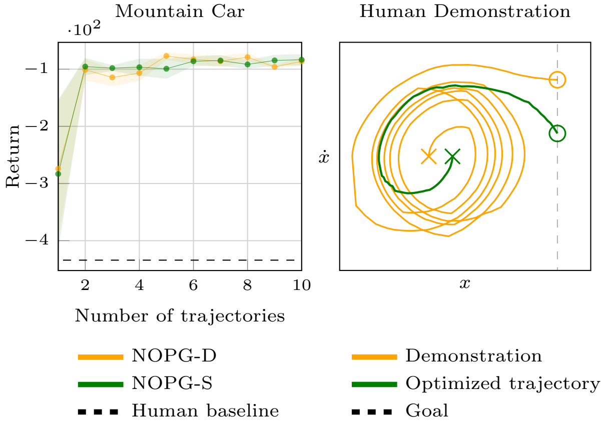

Figure 4: With a small amount of data NOPG is able to reach a policy that surpasses the human demonstrator (dashed line) in the mountain car environment. Depicted are the mean and 95% confidence over 10 trials (left). An example of a human-demonstrated trajectory and the relative optimized version obtained with NOPG (right). Although the human trajectories in the dataset are suboptimal, NOPG converges to an optimal solution (right).

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

We proposed a novel off-policy policy gradient method that is based on a nonparametric Bellman equation and provides a full-gradient estimate that can be computed in closed-form. Our approach avoids the issues of pathwise importance sampling and semi-gradient methods. More explicitly, the full-gradient estimate is less biased than the semi-gradient, and exhibits lower variance that the gradient estimates of importance-sampling-based approaches. Moreover, our algorithm enables learning from human demonstrations and non-trajectory-based datasets. An overview highlighting the pros and cons of all mentioned approaches is given in Table 1.

To support our argument and findings we provide both a theoretical and empirical analysis conducted in different scenarios and compare to state-of-the-art off-policy algorithms. Our theoretical analysis provides a bound on the estimation bias and highlights the impact of different factors. The empirical analysis shows that our method succeeded in learning near-optimal policies in off-policy settings where semi-gradient approaches fail. Furthermore, our approach was able to leverage unstructured datasets and human demonstrations, two scenarios where importance sampling techniques are not applicable.

The accurate gradient estimate delivered by our algorithm results in dramatically overall lower sample complexity when compared to state-of-the-art off-policy policy gradients. By relying on nonparametric statistics, we sacrifice scalability for higher sample efficiency and safety, bringing reinforcement learning one step closer to real world applications and robotics.

Future research will concentrate on extending our approach to parametric models to address scalability, exploring the possibility of using NOPG in a Bayesian framework in order to deal with the problems of uncertainty and model bias, and on enabling a principled online exploration and informative data collection.

7 Acknowledgment

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

References

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

Borrelli, F., Bemporad, A., and Morari, M. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, June 2017.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym. *arXiv:1606.01540*, 2016.

Degris, T., White, M., and Sutton, R. S. Off-Policy Actor-Critic. *arXiv:1205.4839 [cs]*, May 2012.

Deisenroth, M. P. and Rasmussen, C. E. PILCO: A Model-based and Data-efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 465–472. 2011.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai Baselines. *GitHub, GitHub repository*, 2017.

Engel, Y., Mannor, S., and Meir, R. Reinforcement Learning with Gaussian Processes. In *Proceedings of the 22nd International Conference On Machine Learning*, pp. 201–208. 2005.

Ernst, D., Geurts, P., and Wehenkel, L. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

Fan, J. Design-Adaptive Nonparametric Regression. *Journal of the American Statistical Association*, 87 (420):998–1004, 1992.

Fujimoto, S., Meger, D., and Precup, D. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceeding of the 36th International Conference on Machine Learning*, pp. 2052–2062, 2019.

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

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

Kober, J. and Peters, J. R. Policy Search for Motor Primitives in Robotics. In *Advances in Neural Information Processing Systems*, pp. 849–856, 2009.

Kroemer, O. B. and Peters, J. R. A Non-Parametric Approach to Dynamic Programming. In *Advances in Neural Information Processing Systems*, pp. 1719–1727. 2011.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous Control with Deep Reinforcement Learning. In *International Conference on Learning Representations*, 2016.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the Curse of Horizon: Infinite-Horizon Off-Policy Estimation. In *Advances in Neural Information Processing Systems*, pp. 5356–5366, 2018.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Off-Policy Policy Gradient with State Distribution Correction. *arXiv:1904.08473*, 2019.

Lu, T., Schuurmans, D., and Boutilier, C. Non-Delusional Q-learning and Value-Iteration. In *Advances in Neural Information Processing Systems*, pp. 9949–9959. 2018.

Metelli, A. M., Papini, M., Faccio, F., and Restelli, M. Policy Optimization via Importance Sampling. In *Advances in Neural Information Processing Systems*, pp. 5442–5454. Curran Associates, Inc., 2018.

Meuleau, N., Peshkin, L., and Kim, K.-E. Exploration in Gradient-Based Reinforcement Learning. Technical report, Massachusetts Institute of Technology, 2001. URL `https://dspace.mit.edu/handle/1721.1/6076`.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540):529–533, 2015.

Nadaraya, E. A. On Estimating Regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

Ormoneit, D. and Sen, S. Kernel-Based Reinforcement Learning. *Machine Learning*, 49(2):161–178, 2002.

Owen, A. B. *Monte Carlo Theory, Methods and Examples.* 2013.

Peshkin, L. and Shelton, C. R. Learning from Scarce Experience. In *Proceedings of the Nineteenth International Conference on Machine Learning*, 2002.

Riedmiller, M. Neural Fitted Q Iteration First Experiences with a Data Efficient Neural Reinforcement Learning Method. In *European Conference of Machine Learning*, pp. 317–328. 2005.

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

Shelton, C. R. Policy Improvement for POMDPs Using Normalized Importance Sampling. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pp. 496–503. 2001.

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

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

Taylor, G. and Parr, R. Kernelized Value Function Approximation for Reinforcement Learning. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 1017–1024. 2009.

Tosatto, S., Akrour, R., and Peters, J. An Upper Bound of the Bias of Nadaraya-Watson Kernel Regression under Lipschitz Assumptions. *arXiv preprint arXiv:2001.10972*, 2020.

Wasserman, L. *All of Nonparametric Statistics*. Springer, 2006.

Watson, G. S. Smooth Regression Analysis. *Sankhy: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

Williams, R. J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine learning*, 8(3-4):229–256, 1992.

Xu, X., Hu, D., and Lu, X. Kernel-Based Least Squares Policy Iteration for Reinforcement Learning. *IEEE Transactions on Neural Networks*, 18(4): 973–992, 2007.

Supplementary Material

A The Nonparametric Bellman Equation

This section contains the proofs of Theorem 1 and Theorem 3.

Proposition 1. *In the limit of infinite samples the NPBE defined in Definition 2 with a data-set* $\lim_{n\to\infty} D_n$ *collected under distribution* β *on the state-action space and MDP* $\mathcal{M}$ *converges to*

$$\begin{aligned}\hat{V}_\pi(\mathbf{s}) \quad &= \quad \int_{\mathcal{S}\times\mathcal{A}} \varepsilon_\pi(\mathbf{s},\mathbf{z},\mathbf{b})\left(R_{\mathbf{z},\mathbf{b}} + \gamma\int_{\mathcal{S}}\hat{V}_\pi(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\,\mathrm{d}\mathbf{s}'\right)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b},\\ &\quad\ \ \textit{with} \quad R_{\mathbf{z},\mathbf{b}} \sim R(\mathbf{z},\mathbf{b}) \quad \forall(\mathbf{z},\mathbf{b})\in\mathcal{S}\times\mathcal{A},\\ &\quad\ \ \textit{with} \quad \mathbf{z}'_{\mathbf{z},\mathbf{b}} \sim P(\cdot|\mathbf{z},\mathbf{b}) \quad \forall(\mathbf{z},\mathbf{b})\in\mathcal{S}\times\mathcal{A}.\end{aligned} \tag{12}$$

and

$$\begin{cases}\varepsilon_\pi(\mathbf{s},\mathbf{z},\mathbf{b}) := \int_{\mathcal{A}} \dfrac{\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a} & \textit{if } \pi \textit{ is stochastic,}\\ \varepsilon_i^\pi(\mathbf{s}) := \dfrac{\psi(\mathbf{s},\mathbf{z})\varphi(\pi(\mathbf{s}),\mathbf{b})}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\pi(\mathbf{s}),\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}} & \textit{otherwise.}\end{cases}$$

Proof.

$$\begin{aligned}\hat{V}_\pi(\mathbf{s}) \quad &= \quad \lim_{n\to\infty}\int_{\mathcal{A}} \frac{\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a})\left(r_i + \gamma\int_{\mathcal{S}}\phi_i(\mathbf{s}')\hat{V}_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}\right)}{\sum_{i=1}^n \psi_j(\mathbf{s})\varphi_j(\mathbf{a})}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\\ &= \quad \int_{\mathcal{A}} \frac{\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n \psi_i(\mathbf{s})\varphi_i(\mathbf{a})\left(r_i + \gamma\int_{\mathcal{S}}\phi_i(\mathbf{s}')\hat{V}_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}\right)}{\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^n \psi_j(\mathbf{s})\varphi_j(\mathbf{a})}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\\ &= \quad \int_{\mathcal{A}} \frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\left(R(\mathbf{z},\mathbf{b}) + \gamma\int_{\mathcal{S}}\phi(\mathbf{s}',\mathbf{z}')p(\mathbf{z}'|\mathbf{b},\mathbf{z})\hat{V}_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}\right)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}.\end{aligned}$$

Analogously we can derive the deterministic policy case. □

Proposition 2. *We want to show that, if a solution* $\hat{V}_\pi(\mathbf{s})$ *exists, it is bounded by* $|\hat{V}_\pi(\mathbf{s})| \leq R_{\max}/(1-\gamma)$*, where* $R_{\max} = \max_i |r_i|$.

Proof. Suppose by absurd that exists a state $\mathbf{z}\in\mathcal{S}$ such that $|\hat{V}_\pi(\mathbf{z})| = R_{\max}/(1-\gamma)+\epsilon$ with $\epsilon\in\mathbb{R}^+$. Then,

$$\frac{R_{\max}}{1-\gamma} + \epsilon = \boldsymbol{\varepsilon}^\intercal(\mathbf{z})\mathbf{r} + \gamma\boldsymbol{\varepsilon}^\intercal(\mathbf{z})\int_{\mathcal{S}}\boldsymbol{\phi}(\mathbf{s}')\hat{V}_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}'. \tag{13}$$

Since, by definition, the equation must be correctly fulfilled, we notice that $|\varepsilon_\pi^\intercal(\mathbf{z})| \leq R_{\max}$, since $\varepsilon_\pi(\mathbf{z})$ is a stochastic vector, therefore

$$\left|\frac{R_{\max}}{1-\gamma} + \epsilon - \gamma\boldsymbol{\varepsilon}^\intercal(\mathbf{z})\int_{\mathcal{S}}\boldsymbol{\phi}(\mathbf{s}')\hat{V}_\pi(\mathbf{s}')\,\mathrm{d}\mathbf{s}'\right| \leq R_{\max}. \tag{14}$$

However,

$$\begin{aligned}\frac{R_{\max}}{1-\gamma} + \epsilon - \gamma \boldsymbol{\varepsilon}^{\intercal}(\mathbf{z}) \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\hat{V}_{\pi}(\mathbf{s}') \,\mathrm{d}\mathbf{s}' &\geq \frac{R_{\max}}{1-\gamma} + \epsilon - \gamma \left(\frac{R_{\max}}{1-\gamma} + \epsilon\right) \\ &\geq R_{\max} + \gamma\epsilon\end{aligned}$$

this is in contraddiction with 14. □

Proposition 3. *If* $\mathbf{r}$ *is bounded by* R_{max} *and if* $f^* : \mathcal{S} \to \mathbb{R}$ *satisfies the NPBE, then there is no other function* $f : \mathcal{S} \to \mathbb{R}$ *for which* $\exists \mathbf{z} \in \mathcal{S}$ *and* $|f^*(\mathbf{z}) - f(\mathbf{z})| > 0$.

Proof. Suppose, by absurd assumption, that a function $g : \mathcal{S} \to \mathbb{R}$ exists such that $f(\mathbf{s}) + g(\mathbf{s})$ satisfies Equation (12) for every $\mathbf{s} \in \mathcal{S}$ and a constant $G \in \mathbb{R}^+$ exists for which $|g(\mathbf{z})| > G$. Note that the existence of $f : \mathcal{S} \to \mathbb{R}$ as a solution for the NPBE implies the existence of

$$\int_{\mathcal{S}} \boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s})\boldsymbol{\phi}(\mathbf{s}')f^*(\mathbf{s}') \,\mathrm{d}\mathbf{s}' \in \mathbb{R}, \tag{15}$$

and similarly, the existence of $f(\mathbf{s}) \in \mathbb{R}$ with $f(\mathbf{s}) = f^*(\mathbf{s}) + g(\mathbf{s})$ as a solution of the NPBE implies that

$$\int_{\mathcal{S}} \boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s})\boldsymbol{\phi}(\mathbf{s}')f^*(\mathbf{s}') + g(\mathbf{s}') \,\mathrm{d}\mathbf{s}' \in \mathbb{R}. \tag{16}$$

Note that the existence of the integral in Equations (15) and (16) implies

$$\int_{\mathcal{S}} \boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s})\boldsymbol{\phi}(\mathbf{s}')g(\mathbf{s}') \,\mathrm{d}\mathbf{s}' \in \mathbb{R}. \tag{17}$$

Note that

$$\begin{aligned}|f^*(\mathbf{s}) - f(\mathbf{s})| &= \left| f^*(\mathbf{s}) - \boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s})\left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\big(f(\mathbf{s}') + g(\mathbf{s}')\big) \,\mathrm{d}\mathbf{s}'\right)\right| \\ &= \left|\boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s})\left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')g(\mathbf{s}') \,\mathrm{d}\mathbf{s}'\right) - \boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s})\left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\big(f^*(\mathbf{s}') + g(\mathbf{s}')\big) \,\mathrm{d}\mathbf{s}'\right)\right| \\ &= \gamma\left|\boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s}) \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')g(\mathbf{s}') \,\mathrm{d}\mathbf{s}'\right| \\ \implies |g(\mathbf{s})| &= \gamma\left|\boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s}) \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')g(\mathbf{s}') \,\mathrm{d}\mathbf{s}'\right|.\end{aligned}$$

Using Jensen's inequality

$$|g(\mathbf{s})| \leq \gamma\boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s}) \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')|g(\mathbf{s}')| \,\mathrm{d}\mathbf{s}'.$$

Note that since both f^* and f are bounded by $\frac{R_{\max}}{1-\gamma}$ then $|g(\mathbf{s})| \leq \frac{2R_{\max}}{1-\gamma}$, thus

$$\begin{aligned}|g(\mathbf{s})| &\leq \gamma\boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s}) \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')|g(\mathbf{s}')| \,\mathrm{d}\mathbf{s}' \qquad (18) \\ &\leq \gamma 2\frac{R_{\max}}{1-\gamma}\boldsymbol{\varepsilon}_{\pi}^{T}(\mathbf{s}) \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}') \,\mathrm{d}\mathbf{s}' \\ &= \gamma 2\frac{R_{\max}}{1-\gamma} \\ \implies |g(\mathbf{s})| &\leq \gamma\frac{2R_{\max}}{1-\gamma} \\ \implies |g(\mathbf{s})| &\leq \gamma^2\frac{2R_{\max}}{1-\gamma} \qquad \text{using (18)} \\ \implies |g(\mathbf{s})| &\leq \gamma^3\frac{2R_{\max}}{1-\gamma} \qquad \text{using (18)} \\ &\dots \\ \implies |g(\mathbf{s})| &\leq 0,\end{aligned}$$

which is in clear disagreement with the assumption made. Again here a similar procedure shows the same result for the infinite case. □

Proof of Theorem 1

Proof. Saying that $\hat{V}_\pi^*$ is a solution for Equation (12) is equivalent to saying

$$\hat{V}_\pi^*(\mathbf{s}) - \boldsymbol{\varepsilon}^\pi(\mathbf{s})\left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\hat{V}_\pi^*(\mathbf{s}')\,\mathrm{d}\mathbf{s}'\right) = 0 \qquad \forall \mathbf{s} \in \mathcal{S}.$$

We can verify that by simple algebraic manipulation

$$\begin{aligned}
& \hat{V}_\pi^*(\mathbf{s}) - \boldsymbol{\varepsilon}_\pi^T(\mathbf{s})\left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\hat{V}_\pi^*(\mathbf{s}')\,\mathrm{d}\mathbf{s}'\right) \\
= \;& \boldsymbol{\varepsilon}_\pi^T(\mathbf{s})\boldsymbol{\Lambda}_\pi^{-1}\mathbf{r} - \boldsymbol{\varepsilon}^\pi(\mathbf{s})\left(\mathbf{r} + \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\boldsymbol{\varepsilon}_\pi^T(\mathbf{s}')\boldsymbol{\Lambda}_\pi^{-1}\mathbf{r}\,\mathrm{d}\mathbf{s}'\right) \\
= \;& \boldsymbol{\varepsilon}_\pi^T(\mathbf{s})\left(\boldsymbol{\Lambda}_\pi^{-1}\mathbf{r} - \mathbf{r} - \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\boldsymbol{\varepsilon}_\pi^T(\mathbf{s}')\boldsymbol{\Lambda}_\pi^{-1}\mathbf{r}\,\mathrm{d}\mathbf{s}'\right) \\
= \;& \boldsymbol{\varepsilon}_\pi^T(\mathbf{s})\left(\left(I - \gamma \int_{\mathcal{S}} \boldsymbol{\phi}(\mathbf{s}')\boldsymbol{\varepsilon}_\pi^T(\mathbf{s}')\,\mathrm{d}\mathbf{s}'\right)\boldsymbol{\Lambda}_\pi^{-1}\mathbf{r} - \mathbf{r}\right) \\
= \;& \boldsymbol{\varepsilon}_\pi^T(\mathbf{s})\left(\boldsymbol{\Lambda}_\pi\boldsymbol{\Lambda}_\pi^{-1}\mathbf{r} - \mathbf{r}\right) \\
= \;& 0. \qquad (19)
\end{aligned}$$

Since equation (12) has (at least) one solution, Proposition 3 guarantees that the solution ($\hat{V}_\pi^*$) is unique. □

Proof of Theorem 3.

Proof. We perform the derivation for the stochastic policy, however the same derivation applies for the deterministic case almost identically. Expanding $\left|\mathbb{E}_D[\overline{V}_D(\mathbf{s})] - V^*(\mathbf{s})\right|$ using the NPBE and the classic Bellman equation,

$$\begin{aligned}
\left|\mathop{\mathbb{E}}_D[\overline{V}_D(\mathbf{s})] - V^*(\mathbf{s})\right| \;=\; & \Bigg|\mathop{\mathbb{E}}_D\left[\int_{\mathcal{S}\times\mathcal{A}} \varepsilon_\pi(\mathbf{s},\mathbf{z},\mathbf{b})\left(R_{\mathbf{z},\mathbf{b}} + \gamma \int_{\mathcal{S}} V_D(\mathbf{s}')\phi(\mathbf{s}', \mathbf{z}'_{\mathbf{z},\mathbf{b}})\,\mathrm{d}\mathbf{s}\right)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}\right] \\
& - \int_{\mathcal{A}}\left(\overline{R}(\mathbf{s},\mathbf{a}) + \gamma \int_{\mathcal{S}} V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\right)\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\Bigg|. \qquad (20)
\end{aligned}$$

As can be easily verified, $\varepsilon_\pi(\mathbf{s},\mathbf{z},\mathbf{b})\beta(\mathbf{z},\mathbf{b})$ is a density distribution over $\mathbf{z},\mathbf{b}$. Hence Equation (20) can be rewritten

$$\begin{aligned}
& \Bigg|\mathop{\mathbb{E}}_{D}\left[\int_{\mathcal{S}\times\mathcal{A}}\varepsilon_\pi(\mathbf{s},\mathbf{z},\mathbf{b})\left(R_{\mathbf{z},\mathbf{b}}+\gamma\int_{\mathcal{S}}V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\,\mathrm{d}\mathbf{s}'\right)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}\right] \\
& -\int_{\mathcal{A}}\left(\overline{R}(\mathbf{s},\mathbf{a})+\gamma\int_{\mathcal{S}}V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\right)\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\Bigg| \\
= \quad & \Bigg|\mathop{\mathbb{E}}_{D}\left[\int_{\mathcal{A}}\frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(R_{\mathbf{z},\mathbf{b}}-\overline{R}(\mathbf{s},\mathbf{a})\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\right] \\
+ \quad & \gamma\int_{\mathcal{A}}\mathop{\mathbb{E}}_{D}\left[\frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}}V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\,\mathrm{d}\mathbf{s}'-\int_{\mathcal{S}}V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\right]\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\Bigg| \\
\leq \quad & \Bigg|\mathop{\mathbb{E}}_{D}\left[\int_{\mathcal{A}}\underbrace{\frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(R_{\mathbf{z},\mathbf{b}}-\overline{R}(\mathbf{s},\mathbf{a})\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}}_{\text{A}}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\right]\Bigg| \\
+ \quad & \gamma\Bigg|\int_{\mathcal{A}}\underbrace{\mathop{\mathbb{E}}_{D}\left[\frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}}V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\,\mathrm{d}\mathbf{s}'-\int_{\mathcal{S}}V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\right]}_{\text{B}}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\Bigg| \\
\leq \quad & \mathrm{A}_{\text{Bias}}+\gamma\mathrm{B}_{\text{Bias}}. \qquad (21)
\end{aligned}$$

It is evident that the term A is the Nadaraya-Watson kernel regression, as it is possible to observe in the beginnin of the proof at page twelve of Tosatto et al. (2020), therefore Theorem 2 applies

$$\mathrm{A}_{\text{Bias}} \quad = \quad \frac{L_R\sum_{k=1}^d\mathbf{h}_k\left(\prod_{i\neq k}^d e^{\frac{L_\beta^2\mathbf{h}_i^2}{2}}\left(1+\operatorname{erf}\left(\frac{\mathbf{h}_iL_\beta}{\sqrt{2}}\right)\right)\right)\left(\frac{1}{\sqrt{2\pi}}+L_\beta\mathbf{h}_k\frac{e^{\frac{L_\beta^2\mathbf{h}_k^2}{2}}}{2}\left(1+\operatorname{erf}\left(\frac{\mathbf{h}_kL_\beta}{\sqrt{2}}\right)\right)\right)}{\prod_{i=1}^d e^{\frac{L_\beta^2h_i^2}{2}}\left(1-\operatorname{erf}\left(\frac{\mathbf{h}_iL_\beta}{\sqrt{2}}\right)\right)},$$

where $\mathbf{h}=[\mathbf{h}_\psi,\mathbf{h}_\varphi]$ and $d=d_s+d_a$.
Returning to the estimate of B_{Bias}

$$\begin{aligned}
& \Bigg|\int_{\mathcal{A}}\mathop{\mathbb{E}}_{D}\left[\frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}}V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\,\mathrm{d}\mathbf{s}'-\int_{\mathcal{S}}V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\right]\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\Bigg| \\
= \quad & \Bigg|\int_{\mathcal{A}}\frac{\int_{\mathcal{S}\times\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}}\mathbb{E}\left[V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\right]\mathrm{d}\mathbf{s}'-\int_{\mathcal{S}}V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}}\psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}\pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a}\Bigg|
\end{aligned}$$

One my ask whether the terms in $\mathbb{E}[V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})]$ are uncorrelated. The answer it is affirmative, since, even if V_D depends by $\mathbf{z}_{\mathbf{z},\mathbf{b}}$ (integral in Equation (12)), this corresponds only in the variation of a single point in the integral, and therefore the overall estimate does not change. This argument, however, does not immediately hold for the case of an infinitesimal bandwidth, and therefore we provide the results for that case separately.

For Finite Bandwidth:

$$\begin{aligned}
& \left| \int_{\mathcal{A}} \frac{\int_{\mathcal{S}\times\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}} \mathbb{E}\left[V_D(\mathbf{s}')\phi(\mathbf{s}',\mathbf{z}'_{\mathbf{z},\mathbf{b}})\right] \mathrm{d}\mathbf{s}' - \int_{\mathcal{S}} V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}} \pi(\mathbf{a}|\mathbf{s})\,\mathrm{d}\mathbf{a} \right| \\
\leq \quad & \max_{\mathbf{s},\mathbf{a}} \left| \frac{\int_{\mathcal{S}\times\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}\times\mathcal{S}} \overline{V}(\mathbf{z}')\phi(\mathbf{z}',\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\,\mathrm{d}\mathbf{z}' - \int_{\mathcal{S}} V^*(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}} \right| \\
= \quad & \max_{\mathbf{s},\mathbf{a}} \left| \frac{\int_{\mathcal{S}\times\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}}\int_{\mathcal{S}} \big(\overline{V}(\mathbf{z}')\phi(\mathbf{z}',\mathbf{s}') - V^*(\mathbf{s}')\big)p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'\,\mathrm{d}\mathbf{z}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}} \right| \\
\leq \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \frac{\int_{\mathcal{S}\times\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\big(\int_{\mathcal{S}} \overline{V}(\mathbf{z}')\phi(\mathbf{z}',\mathbf{s}') - V^*(\mathbf{s}')\,\mathrm{d}\mathbf{z}'\big)\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}} \right| \\
= \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \frac{\int_{\mathcal{S}\times\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}}{\int_{\mathcal{S},\mathcal{A}} \psi(\mathbf{s},\mathbf{z})\varphi(\mathbf{a},\mathbf{b})\beta(\mathbf{z},\mathbf{b})\,\mathrm{d}\mathbf{z}\,\mathrm{d}\mathbf{b}} \left(\int_{\mathcal{S}} \overline{V}(\mathbf{z}')\phi(\mathbf{z}',\mathbf{s}') - V^*(\mathbf{s}')\,\mathrm{d}\mathbf{z}' \right) \right| \\
= \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \int_{\mathcal{S}} \overline{V}(\mathbf{z}')\phi(\mathbf{z}',\mathbf{s}') - V^*(\mathbf{s}')\,\mathrm{d}\mathbf{z}' \right| \\
= \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \int_{\mathcal{S}} \overline{V}(\mathbf{s}'+\mathbf{l})\phi(\mathbf{s}+\mathbf{l},\mathbf{s}') - V^*(\mathbf{s}')\,\mathrm{d}\mathbf{l} \right|. \qquad (22)
\end{aligned}$$

Note that

$$\phi(\mathbf{s}'+\mathbf{l},\mathbf{s}') = \prod_{i=1}^{d_s} \frac{e^{-\frac{l_i^2}{2h_{\phi,i}^2}}}{\sqrt{2\pi h_{\phi,i}^2}},$$

thus

$$\begin{aligned}
& \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \int_{\mathcal{S}} \overline{V}(\mathbf{s}'+\mathbf{l})\phi(\mathbf{s}+\mathbf{l},\mathbf{s}') - V^*(\mathbf{s}')\,\mathrm{d}\mathbf{l} \right| \\
\leq \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \overline{V}(\mathbf{s}') - V^*(\mathbf{s}') \right| + \int_{\mathcal{S}} L_V \left(\sum_{i=1}^{d_s} |l_i| \right) \prod_{i=1}^{d_s} \frac{e^{-\frac{l_i^2}{2h_{\phi,i}^2}}}{\sqrt{2\pi h_{\phi,i}^2}}\,\mathrm{d}.\mathbf{l}
\end{aligned}$$

Using Proposition **??**

$$\begin{aligned}
& \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \overline{V}(\mathbf{s}') - V^*(\mathbf{s}') \right| + L_V \int_{\mathcal{S}} \left(\sum_{i=1}^{d_s} |l_i| \right) \prod_{i=1}^{d_s} \frac{e^{-\frac{l_i^2}{2h_{\phi,i}^2}}}{\sqrt{2\pi h_{\phi,i}^2}}\,\mathrm{d}\mathbf{l} \\
= \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \overline{V}(\mathbf{s}') - V^*(\mathbf{s}') \right| + L_V \sum_{k=1}^{d_s} \left(\prod_{i\neq k}^{d_s} \int_{-\infty}^{+\infty} \frac{e^{-\frac{l_i^2}{2h_{\phi,i}^2}}}{\sqrt{2\pi h_{\phi,i}^2}}\,\mathrm{d}l_i \right) \int_{-\infty}^{+\infty} |l_k| \frac{e^{-\frac{l_k^2}{2h_{\phi,k}^2}}}{\sqrt{2\pi h_{\phi,k}^2}}\,\mathrm{d}l_k \\
= \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \overline{V}(\mathbf{s}') - V^*(\mathbf{s}') \right| + L_V 2 \sum_{k=1}^{d_s} \int_{0}^{+\infty} l_k \frac{e^{-\frac{l_k^2}{2h_{\phi,k}^2}}}{\sqrt{2\pi h_{\phi,k}^2}}\,\mathrm{d}l_k \\
= \quad & \max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \overline{V}(\mathbf{s}') - V^*(\mathbf{s}') \right| + L_V \sum_{k=1}^{d_s} \frac{h_{\phi,k}}{\sqrt{2\pi}}
\end{aligned} \qquad (23)$$

which means that when **h** not infinitesimal

$$\left| \overline{V}(\mathbf{s}) - V^*(\mathbf{s}) \right| \leq \mathrm{A}_{\mathrm{Bias}} + \gamma \left(\max_{\mathbf{s},\mathbf{a},\mathbf{s}'} \left| \overline{V}(\mathbf{s}') - V^*(\mathbf{s}') \right| + L_V \sum_{k=1}^{d_s} \frac{h_{\phi,k}}{\sqrt{2\pi}} \right).$$

It is however known that $\left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq 2\frac{R_{\max}}{1-\gamma}$, thus

$$\left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \mathrm{A}_{\text{Bias}} + \gamma\bigg(\max_{\mathbf{s},\mathbf{a},\mathbf{s}'}\left|\overline{V}(\mathbf{s}') - V^*(\mathbf{s}')\right| + L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\bigg) \tag{24}$$

$$\left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \mathrm{A}_{\text{Bias}} + \gamma\bigg(2\frac{R_{\max}}{1-\gamma} + L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\bigg)$$

$$\implies \quad \left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \mathrm{A}_{\text{Bias}} + \gamma\bigg(\mathrm{A}_{\text{Bias}} + \gamma\bigg(2\frac{R_{\max}}{1-\gamma} + L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\bigg) + L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\bigg) \qquad \text{using Equation (24)}$$

$$\implies \quad \left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \sum_{t=0}^{\infty}\gamma^t\bigg(\mathrm{A}_{\text{Bias}} + \gamma L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\bigg) \qquad \text{using Equation (24)}$$

$$\implies \quad \left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \frac{1}{1-\gamma}\bigg(\mathrm{A}_{\text{Bias}} + \gamma L_V\sum_{k=1}^{d_s}\frac{h_{\phi,k}}{\sqrt{2\pi}}\bigg).$$

For Infinitesimal Bandwidth: In the case of an infinitesimal bandwidth note that, even if V_D and ϕ are correlated the overall integral reduces only on a single point, and the same argument made in the case of finite bandwidth applies,

$$\int_{\mathcal{S}}\mathbb{E}\left[V_D(\mathbf{s}')\phi(\mathbf{s}', \mathbf{z}'_{\mathbf{z},\mathbf{b}})\right]\mathrm{d}\mathbf{s}' = \mathbb{E}\left[\int_{\mathcal{S}}V_D(\mathbf{s}')\phi(\mathbf{s}', \mathbf{z}'_{\mathbf{z},\mathbf{b}}\,\mathrm{d}\mathbf{s}')\,\mathrm{d}\mathbf{s}'\right] = \mathbb{E}\left[V_D(\mathbf{z}'_{\mathbf{z},\mathbf{b}})\right] = \int_{\mathcal{S}}\overline{V}_D(\mathbf{s}')p(\mathbf{s}'|\mathbf{s},\mathbf{a})\,\mathrm{d}\mathbf{s}'.$$

It follows that, proceeding similarly to Equation (22), we obtain

$$\left|\mathop{\mathbb{E}}_{D}[\overline{V}_D(\mathbf{s})] - V^*(\mathbf{s})\right| \leq \max_{\mathbf{s},\mathbf{a},\mathbf{s}'}\left|\overline{V}(\mathbf{s}') - V^*(\mathbf{s}')\right|, \tag{25}$$

which yields

$$\left|\overline{V}(\mathbf{s}) - V^*(\mathbf{s})\right| \leq \frac{1}{1-\gamma}\mathrm{A}_{\text{Bias}}. \tag{26}$$

□

B Empirical Evaluation Detail

B.1 Linear Quadratic Regulator Experiment

Here we detail the experiment presented in Figure 1. We use a discrete infinite-horizon discounted Linear Quadratic Regulator system of the form

$$\max_{\vec{x}_t,\vec{u}_t} J = \frac{1}{2}\sum_{t=0}^{\infty}\gamma^t\left(\vec{x}_t^\top\mathbf{Q}\vec{x}_t + \vec{u}_t^\top\mathbf{R}\vec{u}_t\right)$$
$$\vec{x}_{t+1} = \mathbf{A}\vec{x}_t + \mathbf{B}\vec{u}_t \quad \forall t,$$

where $\vec{x}_t \in \mathbb{R}^{d_x}$, $\vec{u}_t \in \mathbb{R}^{d_u}$, $\mathbf{Q} \in \mathbb{R}^{d_x \times d_x}$, $\mathbf{R} \in \mathbb{R}^{d_u \times d_u}$, $\mathbf{A} \in \mathbb{R}^{d_x \times d_x}$, $\mathbf{B} \in \mathbb{R}^{d_x \times d_u}$, $\gamma \in [0, 1)$ and $\vec{x}_0$ given.

In this example we use consider a 2-dimensional problem with the following quantities

$$\mathbf{A} = \begin{bmatrix} 1.2 & 0 \\ 0 & 1.1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix}$$
$$\mathbf{Q} = \begin{bmatrix} -0.5 & 0 \\ 0 & -0.25 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$
$$\vec{x}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \gamma = 0.9.$$

For this LQR problem we impose a linear controller as a diagonal matrix

$$\mathbf{K} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}. \tag{27}$$

B.1.1 Deterministic Experiment

For each dataset we run 100 trajectories of 30 steps. Each trajectory is generated by following the dynamics of the described LQR and using at each time step a fixed policy initialized as

$$\mathbf{K} = \begin{bmatrix} k_1 + \varepsilon & \varepsilon \\ \varepsilon & k_2 + \varepsilon \end{bmatrix}, \; \varepsilon \sim \mathcal{N}(0, 1),$$

where $k_1 = 0.7$ and $k_2 = -0.7$.

NOPG-D optimized for each dataset a policy encoded as in (27) with: learning rate 0.5 with ADAM optimizer; bandwidths (on average) for the state space $\vec{h}_\psi = [0.03, 0.05]$ and for the action space $\vec{h}_\varphi = [0.33, 0.27]$; discount factor $\gamma = 0.9$; and keeping 5 elements per row after sparsification of the $\mathbf{P}$ matrix.

DPG optimized for each dataset a policy encoded as in (27) with: learning rate 0.5 with ADAM optimizer; Q-function encoded as $Q(\vec{x}, \vec{u}) = \vec{x}^\top \mathbf{Q} \vec{x} + \vec{u}^\top \mathbf{R} \vec{u}$ (with $\mathbf{Q}$ and $\mathbf{R}$ to be learned); discount factor $\gamma = 0.9$; two target networks are kept to stabilize learning and soft-updated using $\tau = 0.01$ (similar to DDPG).

B.1.2 Stochastic Experiment

For each dataset we run 100 trajectories of 30 steps. Each trajectory is generated by following the dynamics of the described LQR, and using at each time step a stochastic policy as

$$\vec{u}_t = \mathbf{K}\vec{x}_t + \vec{\varepsilon}, \; \vec{\varepsilon} \sim \mathcal{N}\left(\vec{\mu} = \vec{0}, \mathbf{\Sigma} = \mathrm{diag}(0.01, 0.01)\right), \tag{28}$$

where $\mathbf{K} = \mathrm{diag}(0.35, -0.35)$.

NOPG-S optimized for each dataset a policy encoded as in (28) with: learning rate 0.25 with ADAM optimizer; bandwidths (on average) for the state space $\vec{h}_\psi = [0.008, 0.003]$ and for the action space $\vec{h}_\varphi = [0.02, 0.02]$; discount factor $\gamma = 0.9$; and keeping 10 elements per row after sparsification of the $\mathbf{P}$ matrix.

PWIS optimized for each dataset a policy encoded as in (28) with: learning rate 2.5×10^{-4} with ADAM optimizer; and discount factor $\gamma = 0.9$.

B.2 Other Experiments Configurations

We use a policy encoded as neural network with parameters $\vec{\theta}$. A deterministic policy is encoded with a neural network $\mathbf{a} = f_{\vec{\theta}}(\mathbf{s})$. The stochastic policy is encoded as a Gaussian distribution with parameters determined by a neural network with two outputs, the mean and covariance. In this case we represent by $f_{\vec{\theta}}(\mathbf{s})$ the slice of the output corresponding to the mean and by $g_{\vec{\theta}}(\mathbf{s})$ the part of the output corresponding to the covariance.

NOPG can be described with the following hyper-parameters

NOPG Parameters	Meaning
dataset sizes	number of samples contained in the dataset used for training
discount factor γ	usual discount factor in infinite horizon MDP
state $\vec{h}_{\text{factor}}$	constant used to decide the bandwidths for the state-space
action $\vec{h}_{\text{factor}}$	constant used to decide the bandwidths for the action-space
policy	parametrization of the policy
policy output	how is the output of the policy encoded
learning rate	the learning rate and the gradient ascent algorithm used
N_π^{MC} (NOPG-S)	number of samples drawn to compute the integral $\boldsymbol{\varepsilon}_\pi(\mathbf{s})$ with MonteCarlo sampling
N_ϕ^{MC}	number of samples drawn to compute the integral over the next state $\int \phi(\mathbf{s}')\,\mathrm{d}\mathbf{s}'$
$N_{\mu_0}^{\text{MC}}$	number of samples drawn to compute the integral over the initial distribution $\int \hat{V}_\pi(\mathbf{s})\mu_0(\mathbf{s})\,\mathrm{d}\mathbf{s}$
policy updates	number of policy updates before returning the optimized policy

A few considerations about NOPG parameters. If $N_{\phi}^{\text{MC}} = 1$ we use the mean of the kernel ϕ as a sample to approximate the integral over the next state. When optimizing a stochastic policy represented by a Gaussian distribution, we set and linearly decay the variance over the policy optimization procedure. The kernel bandwidths are computed in two steps: first we find the best bandwidth for each dimension of the state and action spaces using cross validation; second we multiply each bandwidth by an empirical constant factor ($\vec{h}_{\text{factor}}$). This second step is important to guarantee that the state and action spaces do not have a zero density. For instance, in a continuous action environment, when sampling actions from a uniform grid we have to guarantee that the space between the grid points have some density. The problem of estimating the bandwidth in kernel density estimation is well studied, but needs to be adapted to the problem at hand, specially with a low number of samples. We found this approach to work well for our experiments but it still can be improved.

B.2.1 Pendulum with Uniform Dataset

Tables 3 and 4 describe the hyper-parameters used to run the experiment shown in the first plot of Figure 2.

Dataset Generation The dataset have been generated using a grid over the state-action space $\theta, \dot{\theta}, u$, where θ and $\dot{\theta}$ are respectively angle and angular velocity of the pendulum, and u is the torque applied. In Table 3 are enumerated the different dataset used.

$\#\theta$	$\#\dot{\theta}$	$\#u$	Sample size
10	10	2	200
15	15	2	450
20	20	2	800
25	25	2	1250
30	30	2	1800
40	40	2	3200

Table 3: **Pendulum uniform grid dataset configurations** This table shows the level of discretization for each dimension of the state space ($\#\theta$ and $\#\dot{\theta}$) and the action space ($\#u$). Each line corresponds to a uniformly sampled dataset, where $\theta \in [-\pi, \pi]$, $\dot{\theta} \in [-8, 8]$ and $u \in [-2, 2]$. The entries under the states' dimensions and action dimension correspond to how many linearly spaced states or actions are to be queried from the corresponding intervals. The Cartesian product of states and actions dimensions is taken in order to generate the state-action pairs to query the environment transitions. The rightmost column indicates the total number of corresponding samples.

Algorithm details. The configuration used for NOPG-D and NOPG-S are listed in Table 4.

NOPG	
discount factor γ	0.97
state $\vec{h}_{\text{factor}}$	1.0 1.0 1.0
action $\vec{h}_{\text{factor}}$	50.0
policy	neural network parameterized by $\vec{\theta}$
	1 hidden layer, 50 units, ReLU activations
policy output	$2\tanh(f_{\vec{\theta}}(\mathbf{s}))$ (NOGP-D)
	$\mu = 2\tanh(f_{\vec{\theta}}(\mathbf{s}))$, $\sigma = \text{sigmoid}(g_{\vec{\theta}}(\mathbf{s}))$ (NOGP-S)
learning rate	10^{-2} with ADAM optimizer
N_{π}^{MC} (NOPG-S)	15
N_{ϕ}^{MC}	1
$N_{\mu_0}^{\text{MC}}$	(non applicable) fixed initial state
policy updates	$1.5 \cdot 10^3$

Table 4: **NOPG configurations for the Pendulum uniform grid experiment**

B.2.2 Pendulum with Random Agent

The following table shows the hyper-parameters used for generating the second plot starting from the left in Figure 2

NOPG	
dataset sizes	10^2, $5 \cdot 10^2$, 10^3, $1.5 \cdot 10^3$, $2 \cdot 10^3$, $3 \cdot 10^3$, $5 \cdot 10^3$, $7 \cdot 10^3$, $9 \cdot 10^3$, 10^4
discount factor γ	0.97
state $\vec{h}_{\text{factor}}$	1.0 1.0 1.0
action $\vec{h}_{\text{factor}}$	25.0
policy	neural network parameterized by $\vec{\theta}$ 1 hidden layer, 50 units, ReLU activations
policy output	$2\tanh(f_{\vec{\theta}}(\mathbf{s}))$ (NOGP-D) $\mu = 2\tanh(f_{\vec{\theta}}(\mathbf{s}))$, $\sigma = \text{sigmoid}(g_{\vec{\theta}}(\mathbf{s}))$ (NOGP-S)
learning rate	10^{-2} with ADAM optimizer
$N^{\text{MC}}_{\pi_0}$ (NOPG-S)	10
N^{MC}_{ϕ}	1
$N^{\text{MC}}_{\mu_0}$	(non applicable) fixed initial state
policy updates	$2 \cdot 10^3$

DDPG	
discount factor γ	0.97
rollout steps	1000
actor	neural network parameterized by $\vec{\theta}_{\text{actor}}$ 1 hidden layer, 50 units, ReLU activations
actor output	$2\tanh(f_{\vec{\theta}_{\text{actor}}}(\mathbf{s}))$
actor learning rate	10^{-3} with ADAM optimizer
critic	neural network parameterized by $\vec{\theta}_{\text{critic}}$ 1 hidden layer, 50 units, ReLU activations
critic output	$f_{\vec{\theta}_{\text{critic}}}(\mathbf{s}, \mathbf{a})$
critic learning rate	10^{-2} with ADAM optimizer
soft update	$\tau = 10^{-3}$
policy updates	$3 \cdot 10^5$

DDPG Offline	
dataset sizes	10^2, $5 \cdot 10^2$, 10^3, $2 \cdot 10^3$, $5 \cdot 10^3$, $7.5 \cdot 10^3$, 10^4, $1.2 \cdot 10^4$, $1.5 \cdot 10^4$, $2 \cdot 10^4$, $2.5 \cdot 10^4$
discount factor γ	0.97
actor	neural network parameterized by $\vec{\theta}_{\text{actor}}$ 1 hidden layer, 50 units, ReLU activations
actor output	$2\tanh(f_{\vec{\theta}_{\text{actor}}}(\mathbf{s}))$
actor learning rate	10^{-2} with ADAM optimizer
critic	neural network parameterized by $\vec{\theta}_{\text{critic}}$ 1 hidden layer, 50 units, ReLU activations
critic output	$f_{\vec{\theta}_{\text{critic}}}(\mathbf{s}, \mathbf{a})$
critic learning rate	10^{-2} with ADAM optimizer
soft update	$\tau = 10^{-3}$
policy updates	$2 \cdot 10^3$

PWIS

dataset sizes	10^2, $5 \cdot 10^2$, 10^3, $2 \cdot 10^3$, $5 \cdot 10^3$, $7.5 \cdot 10^3$, 10^4, $1.2 \cdot 10^4$, $1.5 \cdot 10^4$, $2 \cdot 10^4$, $2.5 \cdot 10^4$
discount factor γ	0.97
policy	neural network parameterized by $\vec{\theta}$ 1 hidden layer, 50 units, ReLU activations
policy output	$\mu = 2\tanh(f_{\vec{\theta}}(\mathbf{s}))$, $\sigma = \text{sigmoid}(g_{\vec{\theta}}(\mathbf{s}))$
learning rate	10^{-2} with ADAM optimizer
policy updates	$2 \cdot 10^3$

Table 5: **Algorithms configurations for the Pendulum random data experiment**

B.2.3 Cart-pole with Random Agent

The following tables show the hyper-parameters used to generate the third plot in Figure 2.

NOPG

dataset sizes	10^2, $2.5 \cdot 10^2$, $5 \cdot 10^2$, 10^3, $1.5 \cdot 10^3$, $2.5 \cdot 10^3$, $3 \cdot 10^3$, $5 \cdot 10^3$, $6 \cdot 10^3$, $8 \cdot 10^3$, 10^4
discount factor γ	0.99
state $\vec{h}_{\text{factor}}$	1.0 1.0 1.0
action $\vec{h}_{\text{factor}}$	20.0
policy	neural network parameterized by $\vec{\theta}$ 1 hidden layer, 50 units, ReLU activations
policy output	$5\tanh(f_{\vec{\theta}}(\mathbf{s}))$ (NOGP-D) $\mu = 5\tanh(f_{\vec{\theta}}(\mathbf{s}))$, $\sigma = \text{sigmoid}(g_{\vec{\theta}}(\mathbf{s}))$ (NOGP-S)
learning rate	$\cdot 10^{-2}$ with ADAM optimizer
N_{π}^{MC} (NOPG-S)	10
N_{ϕ}^{MC}	1
$N_{\mu_0}^{\text{MC}}$	15
policy updates	$2 \cdot 10^3$

DDPG

discount factor γ	0.99
rollout steps	1000
actor	neural network parameterized by $\vec{\theta}_{\text{actor}}$ 1 hidden layer, 50 units, ReLU activations
actor output	$5\tanh(f_{\vec{\theta}_{\text{actor}}}(\mathbf{s}))$
actor learning rate	10^{-3} with ADAM optimizer
critic	neural network parameterized by $\vec{\theta}_{\text{critic}}$ 1 hidden layer, 50 units, ReLU activations
critic output	$f_{\vec{\theta}_{\text{critic}}}(\mathbf{s}, \mathbf{a})$
critic learning rate	10^{-2} with ADAM optimizer
soft update	$\tau = 10^{-3}$
policy updates	$2 \cdot 10^5$

DDPG Offline

dataset sizes	10^2, $5 \cdot 10^2$, 10^3, $2 \cdot 10^3$, $3.5 \cdot 10^3$, $5 \cdot 10^3$, $8 \cdot 10^3$, 10^4, $1.5 \cdot 10^4$, $2 \cdot 10^4$, $2.5 \cdot 10^4$
discount factor γ	0.99

actor	neural network parameterized by $\vec{\theta}_{\text{actor}}$ 1 hidden layer, 50 units, ReLU activations
actor output	$5\tanh(f_{\vec{\theta}_{\text{actor}}}(\mathbf{s}))$
actor learning rate	10^{-2} with ADAM optimizer
critic	neural network parameterized by $\vec{\theta}_{\text{critic}}$ 1 hidden layer, 50 units, ReLU activations
critic output	$f_{\vec{\theta}_{\text{critic}}}(\mathbf{s}, \mathbf{a})$
critic learning rate	10^{-2} with ADAM optimizer
soft update	$\tau = 10^{-3}$
policy updates	$2 \cdot 10^3$

PWIS	
dataset sizes	10^2, $5 \cdot 10^2$, 10^3, $2 \cdot 10^3$, $3.5 \cdot 10^3$, $5 \cdot 10^3$, $8 \cdot 10^3$, 10^4, $1.5 \cdot 10^4$, $2 \cdot 10^4$, $2.5 \cdot 10^4$
discount factor γ	0.99
policy	neural network parameterized by $\vec{\theta}$ 1 hidden layer, 50 units, ReLU activations
policy output	$\mu = 5\tanh(f_{\vec{\theta}}(\mathbf{s}))$, $\sigma = \text{sigmoid}(g_{\vec{\theta}}(\mathbf{s}))$
learning rate	10^{-3} with ADAM optimizer
policy updates	$2 \cdot 10^3$

Table 6: **Algorithms configurations for the CartPole random data experiment**.

B.2.4 Mountain Car with Human Demonstrator

Here the detail of the experiment shown in Figure 4. The dataset in this experiment (10 trajectories) has been generated by a human demonstrator. The dataset used is available in the source code provided.

NOPG	
discount factor γ	0.99
state $\vec{h}_{\text{factor}}$	1.0 1.0
action $\vec{h}_{\text{factor}}$	50.0
policy	neural network parameterized by $\vec{\theta}$ 1 hidden layer, 50 units, ReLU activations
policy output	$1\tanh(f_{\vec{\theta}}(\mathbf{s}))$ (NOGP-D) $\mu = 1\tanh(f_{\vec{\theta}}(\mathbf{s}))$, $\sigma = \text{sigmoid}(g_{\vec{\theta}}(\mathbf{s}))$ (NOGP-S)
learning rate	10^{-2} with ADAM optimizer
N_{π}^{MC} (NOPG-S)	15
N_{ϕ}^{MC}	1
$N_{\mu_0}^{\text{MC}}$	15
policy updates	$1.5 \cdot 10^3$

Table 7: **NOPG configurations for the MountainCar experiment.**