Journal of Machine Learning Research 21 (2020) 1-75 Submitted 2/20; Revised 6/20; Published 7/20

Importance Sampling Techniques for Policy Optimization

Alberto Maria Metelli albertomaria.metelli@polimi.it
Matteo Papini matteo.papini@polimi.it
Nico Montali[*] nico.montali@mail.polimi.it
Marcello Restelli marcello.restelli@polimi.it
Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)
Piazza Leonardo da Vinci 32
Milano, 20133, Italy

Editor: George Konidaris

Abstract

How can we effectively exploit the collected samples when solving a continuous control task with Reinforcement Learning? Recent results have empirically demonstrated that multiple policy optimization steps can be performed with the same batch by using off–distribution techniques based on importance sampling. However, when dealing with off–distribution optimization, it is essential to take into account the uncertainty introduced by the importance sampling process. In this paper, we propose and analyze a class of model-free, policy search algorithms that extend the recent Policy Optimization via Importance Sampling (Metelli et al., 2018) by incorporating two advanced variance reduction techniques: per–decision and multiple importance sampling. For both of them, we derive a high–probability bound, of independent interest, and then we show how to employ it to define a suitable surrogate objective function that can be used for both action–based and parameter–based settings. The resulting algorithms are finally evaluated on a set of continuous control tasks, using both linear and deep policies, and compared with modern policy optimization methods.

Keywords: Reinforcement Learning, Policy Optimization, Importance Sampling, Per–Decision Importance Sampling, Multiple Importance Sampling

1. Introduction

In recent years, policy search methods (Deisenroth et al., 2013) have proved to be valuable Reinforcement Learning (RL, Sutton and Barto, 1998) approaches thanks to their achievements in continuous control tasks (e.g., Lillicrap et al., 2015; Schulman et al., 2015a,b, 2017), robotic locomotion (e.g., Tedrake et al., 2004; Kober et al., 2013; Heess et al., 2017) and manipulation (e.g., OpenAI et al., 2018, 2019), videogames (e.g., OpenAI, 2018) and partially observable environments (e.g., Ng and Jordan, 2000). These algorithms can be roughly classified into two categories: *action–based* methods (Sutton et al., 2000; Peters and Schaal, 2008b) and *parameter–based* methods (Sehnke et al., 2008). The former, usually known as policy gradient (PG) methods, perform a search in a parametric policy space by following the gradient of the utility function estimated by means of a batch of trajectories collected from the environment (Sutton and Barto, 1998). In contrast, in parameter–based

[*] Now at Waymo.

methods, the search over the space of policy parameters is carried out in a black–box fashion by exploiting global optimizers (e.g., Rubinstein, 1999; Hansen and Ostermeier, 2001; Stanley and Miikkulainen, 2002; Szita and Lörincz, 2006) or following a proper gradient direction like in Policy Gradients with Parameter–based Exploration (PGPE, Sehnke et al., 2008; Wierstra et al., 2008; Sehnke et al., 2010). A major question in policy search methods is: "*how should we use a batch of trajectories in order to exploit its information in the most efficient way?*"

On the one hand, *on–policy* methods leverage the batch to perform a single gradient step, after which new trajectories are collected with the updated policy (e.g., Williams, 1992; Baxter and Bartlett, 2001; Schulman et al., 2015a). However, these methods rarely exploit the available trajectories efficiently, since each batch is discarded after just one gradient update. On the other hand, *off–policy* methods maintain a behavioral policy, used to explore the environment and to collect samples, and a target policy which is optimized. The concept of off–policy learning is rooted in value–based RL (Watkins and Dayan, 1992; Peng and Williams, 1994; Munos et al., 2016) and was first adapted to PG in Degris et al. (2012), using an actor–critic architecture.

While on–policy algorithms are, by their nature, *on–line*, as they need to be fed with fresh samples whenever the policy is updated, off–policy methods can benefit from mixing on–line and *off–line* optimization. This can be done by alternately sampling trajectories and performing optimization epochs with the collected data. A prime example of this alternating procedure is Proximal Policy Optimization (PPO, Schulman et al., 2017), which has displayed remarkable performance in continuous control tasks. Off–line optimization, however, introduces further sources of approximation, as the gradient w.r.t. the target policy needs to be estimated (off–policy) with samples collected with a behavioral policy. A common choice is to adopt an *importance sampling* (IS, Owen, 2013; Hesterberg, 1988) estimator in which each sample is weighted proportionally to the likelihood of being generated by the target policy. However, direct optimization of this utility function is impractical since it likely displays a wide variance (Owen, 2013). Intuitively, the variance increases proportionally to the distance between the behavioral and the target policy; thus, the estimate is reliable as long as the two policies are close enough.

In this paper, we extend *Policy Optimization via Importance Sampling* (POIS) presented in (Metelli et al., 2018) from the theoretical, algorithmic, and experimental viewpoint. POIS is a model–free, actor–only, policy optimization algorithm that mixes on–line and off–line optimization to efficiently leverage the information contained in the collected trajectories. It explicitly accounts for the uncertainty introduced by the importance weighting procedure by optimizing a surrogate objective function that captures the trade–off between the estimated performance improvement and the uncertainty injected by the importance sampling. However, this uncertainty remains a crucial challenge when performing off–policy optimization. The main contributions of this paper over Metelli et al. (2018) are essentially directed to address this latter issue and can be summarized as follows:

1. We introduce the Multiple Importance Sampling technique (MIS, Veach and Guibas, 1995; Owen, 2013) in the POIS framework. MIS allows exploiting trajectories collected with multiple behavioral policies, as opposed to the simple IS in which all the trajectories come from a single behavioral policy. Thus, MIS can bring a significant benefit in terms of sample complexity as, compared to IS, it allows using a larger number of

samples in estimating the performance while collecting the same number of trajectories (Section 3.2).

2. We adapt action–based POIS (A-POIS) to use Per–Decision Importance Sampling (PDIS, Precup et al., 2000), a technique that allows reducing the variance of IS while preserving the unbiasedness of the estimator. PDIS exploits the fact that a reward collected at time t does not depend on the actions and states visited after t to define an importance weight for each trajectory prefix, instead of a single one for the whole trajectory, as in vanilla IS (Section 5.3).[1]

For both techniques, we first derive a bound on the variance of the estimator, then we apply it to derive a suitable concentration inequality that embeds the trade–off between the performance estimator and the dissimilarity between the target policy and the behavioral policy/policies. Finally, we empirically evaluate their performance, comparing the results with those presented in (Metelli et al., 2018).

The paper is organized as follows. We start in Section 2 by introducing the notation and basics about RL policy search. After revising some notions about IS and MIS (Section 3), we propose a concentration inequality, of independent interest, for the high–confidence "off–distribution" optimization of objective functions estimated via IS and MIS (Section 4). Then we show how this bound can be customized into a surrogate objective function in order to either search in the space of policies (Action–based POIS, A-POIS) or to search in the space of parameters (Parameter–based POIS, P-POIS). For the former case, we show how to adapt the algorithm to embed the PDIS technique, deriving a new objective function and algorithm (per–Decision action–based POIS, D-POIS). The resulting algorithms (in both the action–based and the parameter–based flavor) collect, at each iteration, a set of trajectories that are used to perform the off–line optimization of the surrogate objective via gradient ascent, after which a new batch of trajectories is collected using the optimized policy (Section 5). Then, in Section 6, we present a comparative discussion of related works. Finally, we provide an experimental evaluation with both linear policies and deep neural policies to illustrate the advantages and limitations of our approach compared to the state–of–the–art algorithms (Section 7) on classical control tasks (Duan et al., 2016; Todorov et al., 2012). The implementation of POIS can be found at `https://github.com/T3p/baselines`.

2. Preliminaries

In this section, we provide the background and the notation that will be employed in the following sections.

Notation Let $(\mathcal{X}, \mathscr{F})$ be a measurable space, where $\mathcal{X}$ is a set and $\mathscr{F}$ is a σ–algebra over $\mathcal{X}$. Given a probability measure P over $(\mathcal{X}, \mathscr{F})$, we denote with the corresponding lower case letter p the probability density function (p.d.f.) of P w.r.t. the Lebesgue measure, if it exists. We will assume that the probability density function of any probability measure exists, whenever needed. We denote with δ_x the Dirac measure on the given point $x \in \mathcal{X}$. With little abuse of notation, we will replace the probability measure from the expectation $\mathbb{E}_{x \sim P}$ with the corresponding density function $\mathbb{E}_{x \sim p}$, whenever clear from the context. Given

[1] This is the very same observation used in deriving G(PO)MDP (Baxter and Bartlett, 2001) from REINFORCE (Williams, 1992).

a probability measure P, with p.d.f. p, and a measurable function f, we define the $L_\alpha(P)$–norm as $\|f\|_{\alpha,P}^{\alpha} = \int_{\mathcal{X}} |f(x)|^{\alpha} p(x) \,\mathrm{d}x$ for any $\alpha \geqslant 1$, whereas the L_∞–norm is defined as $\|f\|_\infty = \sup_{x \in \mathcal{X}} f(x)$.

Markov Decision Processes A discrete–time Markov Decision Process (MDP, Puterman, 2014) is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, D)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(\cdot|s,a)$ is a Markovian transition model that assigns for each state–action pair (s,a) the probability of reaching the next state s', $\gamma \in [0,1]$ is the discount factor, $R(s,a) \in [-R_{\max}, R_{\max}]$ assigns the expected reward for performing action a in state s which is uniformly bounded by $R_{\max} < +\infty$ and D is the distribution of the initial state. The behavior of an agent is described by a policy $\pi(\cdot|s)$ that assigns for each state s the probability of performing action a. A trajectory $\tau \in \mathcal{T}$ is a sequence of state–action pairs $\tau = (s_{\tau,0}, a_{\tau,0}, \ldots, s_{\tau,H-1}, a_{\tau,H-1}, s_{\tau,H})$, where H is the actual trajectory horizon. The performance of an agent is evaluated in terms of the *expected return*, i.e., the expected discounted sum of the rewards collected along the trajectory: $\mathbb{E}_\tau [R(\tau)]$, where $R(\tau) = \sum_{t=0}^{H-1} \gamma^t R(s_{\tau,t}, a_{\tau,t})$ is the trajectory return.[2]

Policy Search We focus on the case in which the policy belongs to a parametric policy space $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$. In *parameter–based* approaches, the agent is equipped with a *hyperpolicy* ν used to sample the policy parameters at the beginning of each episode. The hyperpolicy belongs itself to a parametric hyperpolicy space $\mathcal{N}_{\mathcal{P}} = \{\nu_{\boldsymbol{\rho}} : \boldsymbol{\rho} \in \mathcal{P} \subseteq \mathbb{R}^r\}$. The expected return can be expressed, in the parameter–based case, as a double expectation: one over the policy parameter space Θ and one over the trajectory space $\mathcal{T}$:

$$J_{\mathcal{M}}(\boldsymbol{\rho}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta} \sim \nu_{\boldsymbol{\rho}} \\ \tau \sim p(\cdot|\boldsymbol{\theta})}} [R(\tau)] = \int_\Theta \int_{\mathcal{T}} \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}) p(\tau|\boldsymbol{\theta}) R(\tau) \,\mathrm{d}\tau \,\mathrm{d}\boldsymbol{\theta}, \tag{1}$$

where $p(\tau|\boldsymbol{\theta}) = D(s_{\tau,0}) \prod_{t=0}^{H-1} \pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t}) P(s_{\tau,t+1}|s_{\tau,t}, a_{\tau,t})$ is the trajectory density function. The goal[3] of a parameter–based learning agent is to determine the hyperparameters $\boldsymbol{\rho}^*$ so as to maximize $J_{\mathcal{M}}(\boldsymbol{\rho})$. If $\nu_{\boldsymbol{\rho}}$ is stochastic and differentiable, the hyperparameters can be learned according to the gradient ascent update: $\boldsymbol{\rho}' = \boldsymbol{\rho} + \alpha \nabla_{\boldsymbol{\rho}} J_{\mathcal{M}}(\boldsymbol{\rho})$, where $\alpha > 0$ is the step size and:

$$\nabla_{\boldsymbol{\rho}} J_{\mathcal{M}}(\boldsymbol{\rho}) = \mathop{\mathbb{E}}_{\substack{\boldsymbol{\theta} \sim \nu_{\boldsymbol{\rho}} \\ \tau \sim p(\cdot|\boldsymbol{\theta})}} [\nabla_{\boldsymbol{\rho}} \log \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}) R(\tau)] = \int_\Theta \int_{\mathcal{T}} \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}) p(\tau|\boldsymbol{\theta}) \nabla_{\boldsymbol{\rho}} \log \nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}) R(\tau) \,\mathrm{d}\tau \,\mathrm{d}\boldsymbol{\theta}.$$

Since the stochasticity of the hyperpolicy is a sufficient source of exploration, deterministic action policies of the kind $\pi_{\boldsymbol{\theta}}(a|s) = \delta_{u_{\boldsymbol{\theta}}(s)}(a)$ are typically considered, where $u_{\boldsymbol{\theta}}$ is a deterministic mapping from $\mathcal{S}$ to $\mathcal{A}$. In what we call the *action–based* case, on the contrary, the hyperpolicy $\nu_{\boldsymbol{\rho}}$ is a deterministic distribution $\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \delta_{g(\boldsymbol{\rho})}(\boldsymbol{\theta})$, where $g(\boldsymbol{\rho})$ is a deterministic

[2] Provided $H \geqslant \frac{1}{1-\gamma} \log \frac{R_{\max}}{\epsilon(1-\gamma)}$, the expected return is ϵ–close to the infinite–horizon case (Kearns and Singh, 2002).

[3] Policy optimization solves a different (typically easier) problem than classic RL, since the set of possible policies is restricted to Π_Θ. The parameter–based approach further modifies the objective function by searching for optimal hyperparameters instead of directly for policy parameters. The possibility of recovering an optimal policy for the original, unconstrained problem depends on the nature of Π_Θ and $\mathcal{N}_{\mathcal{P}}$. The bias on the performance of the learned policy induced by parametrization is discussed, e.g., by Agarwal et al. (2019).

mapping from $\mathcal{P}$ to Θ. For this reason, the dependence on $\boldsymbol{\rho}$ is typically not represented and the expected return expression simplifies into a single expectation over the trajectory space $\mathcal{T}$:[4]

$$J_{\mathcal{M}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} [R(\tau)] = \int_{\mathcal{T}} p(\tau|\boldsymbol{\theta}) R(\tau) \, \mathrm{d}\tau. \tag{2}$$

An action–based learning agent aims to find the policy parameters $\boldsymbol{\theta}^*$ that maximize $J_{\mathcal{M}}(\boldsymbol{\theta})$. In this case, we need to enforce exploration at the action level by means of the stochasticity of $\pi_{\boldsymbol{\theta}}$. For stochastic and differentiable policies, learning can be performed via gradient ascent $\boldsymbol{\theta}' = \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J_D(\boldsymbol{\theta})$, where:

$$\nabla_{\boldsymbol{\theta}} J_{\mathcal{M}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} [\nabla_{\boldsymbol{\theta}} \log p(\tau|\boldsymbol{\theta}) R(\tau)] = \int_{\mathcal{T}} p(\tau|\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} \log p(\tau|\boldsymbol{\theta}) R(\tau) \, \mathrm{d}\tau.$$

3. Evaluation via Importance Sampling

In off–policy evaluation (Thomas et al., 2015b; Thomas and Brunskill, 2016), we aim to estimate the performance of a target policy π_T (or hyperpolicy ν_T) given episodes collected with a set of J behavioral policies $\{\pi_{B_j}\}_{j=1}^J$ (or hyperpolicies $\{\nu_{B_j}\}_{j=1}^J$). More generally, we face the problem of estimating the expected value of a deterministic function f of random variable x taking values in $\mathcal{X}$ under a target distribution P, having at our disposal data sets of samples collected with J behavioral distributions $Q_{1:J} = \{Q_j\}_{j=1}^J$.

3.1. Importance Sampling

When the available samples are drawn from a single distribution Q, i.e., $J = 1$, the *importance sampling* estimator (IS, Cochran, 2007; Owen, 2013) corrects the distribution with the importance weight (or Radon–Nikodym derivative or likelihood ratio) defined as $w_{P/Q}(x) = \frac{p(x)}{q(x)}$ and leading to the estimator:

$$\widehat{\mu}_{P/Q} = \frac{1}{N} \sum_{i=1}^{N} \frac{p(x_i)}{q(x_i)} f(x_i) = \frac{1}{N} \sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i), \tag{3}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_N)^T$ are sampled from Q independently and we assume $q(x) > 0$ whenever $f(x)p(x) \neq 0$. This estimator is unbiased, i.e., $\mathbb{E}_{\mathbf{x} \sim Q}[\widehat{\mu}_{P/Q}] = \mathbb{E}_{x \sim P}[f(x)]$, but it may exhibit an undesirable behavior due to the variability of the importance weights, showing, in some cases, infinite variance. Intuitively, the magnitude of the importance weights provides an indication of how much the probability measures P and Q are dissimilar. This notion can be formalized by the Rényi divergence (Rényi, 1961; Van Erven and Harremos, 2014), an information–theoretic dissimilarity index between probability measures.

Remark 1 (Rényi divergence) Let P and Q be two probability measures on a measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q$ (P is absolutely continuous w.r.t. Q) and Q is σ–finite. Let P and Q admit p and q as Lebesgue probability density functions (p.d.f.),

[4] For notational convenience, and with little abuse, we keep the conditioning on $\boldsymbol{\theta}$ in $p(\cdot|\boldsymbol{\theta})$, although in the action–based case $\boldsymbol{\theta}$ is no longer a random variable.

respectively. The α–Rényi divergence between P and Q is defined as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha-1}\log\int_{\mathcal{X}}\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right)^{\alpha}\mathrm{d}Q = \frac{1}{\alpha-1}\log\int_{\mathcal{X}} q(x)\left(\frac{p(x)}{q(x)}\right)^{\alpha}\mathrm{d}x, \tag{4}$$

where $\mathrm{d}P/\mathrm{d}Q$ is the Radon–Nikodym derivative of P w.r.t. Q and $\alpha \in [0,\infty]$. Some remarkable cases, defined as limits, are: $\alpha = 1$ when $D_1(P\|Q) = D_{\mathrm{KL}}(P\|Q)$ and $\alpha = \infty$ yielding $D_\infty(P\|Q) = \log \operatorname{ess\,sup}_{\mathcal{X}}\left\{\frac{\mathrm{d}P}{\mathrm{d}Q}\right\}$.[5] Importing the notation from Cortes et al. (2010), we denote the exponentiated α–Rényi divergence as $d_\alpha(P\|Q) = \exp\{D_\alpha(P\|Q)\}$. With little abuse of notation, we will replace $D_\alpha(P\|Q)$ with $D_\alpha(p\|q)$ whenever possible within the context. When P and Q are (multivariate) Gaussian distributions, i.e., $P \sim \mathcal{N}(\boldsymbol{\mu}_P, \boldsymbol{\Sigma}_P)$ and $Q \sim \mathcal{N}(\boldsymbol{\mu}_Q, \boldsymbol{\Sigma}_Q)$, the Rényi divergence admits a closed–form for $\alpha \in [0,\infty)$ (Burbea, 1984):

$$D_\alpha(P\|Q) = \frac{\alpha}{2}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q)^T\boldsymbol{\Sigma}_\alpha^{-1}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q) - \frac{1}{2(\alpha-1)}\log\frac{\det(\boldsymbol{\Sigma}_\alpha)}{\det(\boldsymbol{\Sigma}_P)^{1-\alpha}\det(\boldsymbol{\Sigma}_Q)^{\alpha}}, \tag{5}$$

where $\boldsymbol{\Sigma}_\alpha = \alpha\boldsymbol{\Sigma}_Q + (1-\alpha)\boldsymbol{\Sigma}_P$ under the assumption that $\boldsymbol{\Sigma}_\alpha$ is positive–definite. The Rényi divergence can be computed in closed form also for several widely used distributions (Gil et al., 2013).

The Rényi divergence provides a convenient expression for the moments of the importance weights: $\mathbb{E}_{x\sim Q}\left[w_{P/Q}(x)^\alpha\right] = d_\alpha(P\|Q)^{\alpha-1}$. Moreover, we can relate the Rényi divergence with the variance and the essential supremum of the importance weights (Cortes et al., 2010):

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

Remark 2 (Self–Normalized Importance Sampling) A commonly used approach to mitigate the variance problem of the IS estimator, is to resort to the *self–normalized importance sampling* estimator (SN, Cochran, 2007):

$$\widetilde{\mu}_{P/Q} = \frac{\sum_{i=1}^N w_{P/Q}(x_i)f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)} = \sum_{i=1}^N \widetilde{w}_{P/Q}(x_i)f(x_i), \tag{6}$$

where $\widetilde{w}_{P/Q}(x) = w_{P/Q}(x)/\sum_{i=1}^N w_{P/Q}(x_i)$ is the self–normalized importance weight. Differently from $\widehat{\mu}_{P/Q}$, $\widetilde{\mu}_{P/Q}$ is biased but consistent (Owen, 2013) and it typically displays a more desirable behavior because of its smaller variance.[6] A more detailed analysis of the SN estimator can be found in Appendix D. Given the realization $x_1, x_2, \dots, x_N$ we can interpret the SN estimator as the expected value of f under an approximation of the distribution P made by N deltas, i.e., $\widetilde{p}(x) = \sum_{i=1}^N \widetilde{w}_{P/Q}(x)\delta_{x_i}(x)$. The problem of assessing the quality of the SN estimator has been extensively studied by the simulation community,

[5] ess sup is the essential supremum of a measurable function f, i.e., the smallest M such that $f(x) \leqslant M$ *almost everywhere*.

[6] Note that $\left|\widetilde{\mu}_{P/Q}\right| \leqslant \|f\|_\infty$. Therefore, its variance is always finite.

producing several diagnostic indexes to detect when the weights might display problematic behavior (Owen, 2013). The *effective sample size* (ESS) was introduced by Kong (1992) as the number of samples drawn from P so that the variance of the Monte Carlo estimator $\widetilde{\mu}_{P/P}$ (i.e., the sample mean) is approximately equal to the variance of the SN estimator $\widetilde{\mu}_{P/Q}$ computed with N samples. Here we report the original definition and its most common estimate:

$$\begin{aligned} \mathrm{ESS}(P\|Q) &= \frac{N}{\mathbb{V}\mathrm{ar}_{x\sim Q}\left[w_{P/Q}(x)\right]+1} = \frac{N}{d_2(P\|Q)}, \\ \widehat{\mathrm{ESS}}(P\|Q) &= \frac{1}{\sum_{i=1}^{N} \widetilde{w}_{P/Q}(x_i)^2}. \end{aligned} \tag{7}$$

The ESS has an interesting interpretation: if $d_2(P\|Q)=1$, i.e., $P=Q$ almost everywhere, then ESS $=N$ since we are performing Monte Carlo estimation. Otherwise, the ESS decreases as the dissimilarity between the two distributions increases. In the literature, other ESS–like diagnostics have been proposed that also account for the nature of f (Martino et al., 2017).

3.2. Multiple Importance Sampling

The IS estimator can be extended to the case in which we have samples collected with multiple behavioral distributions Q_j, i.e., when $J>1$. In *multiple importance sampling* frameworks (MIS, Veach and Guibas, 1995; Owen, 2013) we have a set of $J \geqslant 1$ behavioral distributions $Q_{1:J}=\{Q_j\}_{j=1}^{J}$ and a data set $\mathbf{x}_j=(x_{1j},x_{2j},...,x_{N_j j})$ of N_j samples collected independently with Q_j, $j=1,2,...,J$. We denote with $N=\sum_{j=1}^{K} N_j$ the total number of samples. The resulting estimator is given by:

$$\widehat{\mu}^{\beta}_{P/Q_{1:J}} = \sum_{j=1}^{J} \frac{1}{N_j} \sum_{i=1}^{N_j} \beta_j(x_{ij}) \frac{p(x_{ij})}{q_j(x_{ij})} f(x_{ij}) = \sum_{j=1}^{J} \frac{1}{N_j} \sum_{i=1}^{N_j} \beta_j(x_{ij}) w_{P/Q_j} f(x_{ij}), \tag{8}$$

where we assume that $q_j(x)>0$ whenever $\beta_j(x)p(x)f(x)=0$ and $\beta_j(x)$ is a *partition of the unity*, i.e., a collection of weight functions for which $\beta_j(x)\geqslant 0$ for all $j=1,2,...,J$ and $\sum_{j=1}^{J}\beta_j(x)=1$ for all $x\in\mathcal{X}$. Several choices for the coefficients β_j (Owen, 2013) are possible. A straightforward, but inefficient, choice is to select $\beta_j(x)=\frac{N_j}{N}$, so as to give equal importance to all the samples. Among all the possible choices for β_j (e.g., cutoff, maximum, power heuristics, see Owen, 2013), the most studied, thanks to its desirable theoretical properties, is the *balance heuristic* (BH, Veach and Guibas, 1995), defined as follows:

$$\beta_j^{\mathrm{BH}}(x) = \frac{N_j q_j(x)}{\sum_{k=1}^{J} N_k q_k(x)}. \tag{9}$$

This particular choice has the advantage of canceling out the q_j in the estimator. In this way, the weight of a given sample x_{ij} does not depend on which component of the mixture it comes from. The resulting estimator has the following form:

$$\widehat{\mu}^{\mathrm{BH}}_{P/Q_{1:J}} = \frac{1}{N} \sum_{j=1}^{J} \sum_{i=1}^{N_j} \frac{p(x_{ij})}{\sum_{k=1}^{J} \frac{N_k}{N} q_k(x_{ij})} f(x_{ij}) = \frac{1}{N} \sum_{j=1}^{J} \sum_{i=1}^{N_j} w^{\mathrm{BH}}_{P/Q_{1:J}}(x_{ij}) f(x_{ij}), \tag{10}$$

	Behavioral distributions	Number of samples	Evaluation complexity
IS	Q	N	$\mathcal{O}(N)$
MIS with BH	$Q_{1:J} = \{Q_j\}_{j=1}^J$	$N = \sum_{j=1}^J N_j$	$\mathcal{O}(NJ)$

Table 1: Comparison between importance sampling (IS) and multiple importance sampling with balance heuristics (MIS with BH) in terms of computational complexity for the evaluation of the corresponding estimators, with the same number of samples. We assume that the evaluations of the density functions p and q_j and of the function f have complexity $\mathcal{O}(1)$.

which can be interpreted as an importance sampling estimator using the mixture of behavioral distributions with mixture weights $\frac{N_k}{N}$, i.e., $\Phi = \sum_{k=1}^K \frac{N_k}{N} Q_k$. Furthermore, this choice of coefficient functions is nearly optimal (Veach and Guibas, 1995, Theorem 1) in terms of variance of the estimator $\widehat{\mu}_{P/Q_{1:J}}$. Although the variance problem is less crucial in the MIS, compared to the IS case, it is possible to combine the MIS estimator with the self–normalization technique. This can be done in two ways: by normalizing the weights separately for each behavioral distribution:

$$\widetilde{\mu}_{P/Q_{1:J}}^{\mathrm{BH}} = \frac{1}{J} \sum_{j=1}^{J} \sum_{i=1}^{N_j} \frac{w_{P/Q_{1:J}}^{\mathrm{BH}}(x_{ij})}{\sum_{k=1}^{N_j} w_{P/Q_{1:J}}^{\mathrm{BH}}(x_{kj})} f(x_{ij}), \tag{11}$$

or by normalizing each weight over all available samples:

$$\widetilde{\widetilde{\mu}}_{P/Q_{1:J}}^{\mathrm{BH}} = \sum_{j=1}^{J} \sum_{i=1}^{N_j} \frac{w_{P/Q_{1:J}}^{\mathrm{BH}}(x_{ij})}{\sum_{h=1}^{J} \sum_{k=1}^{N_h} w_{P/Q_{1:J}}^{\mathrm{BH}}(x_{kh})} f(x_{ij}). \tag{12}$$

Both reduce to the classic SN when $J = 1$. However, the first normalization at Equation (11) degenerates under unit batch sizes, setting all the normalized weights to one. For this reason, we will adopt the second version at Equation (12) in our experiments.

In the policy optimization framework, the major advantage of the MIS estimation, over the standard IS, is the higher sample–efficiency. Indeed, using MIS we can reuse the trajectories generated by all past policies $\{\pi_{B_j}\}_{j=1}^J$ to estimate the performance of the target policy π_T. Differently, with the IS estimator we just reuse the trajectories generated by a single policy π_B, usually the last one. This advantage comes at the cost of higher computational complexity, as we need to evaluate the density function induced by each policy for all the collected trajectories (Table 1).

4. Optimization via Importance Sampling

The off–policy optimization problem (Thomas et al., 2015a) can be formulated as finding the best target policy π_T (or hyperpolicy ν_T), i.e., the one that maximizes the expected return, having access to a set of samples collected with a set of behavioral policies $\{\pi_{B_j}\}_{j=1}^J$ (or hyperpolicies $\{\nu_B\}_{j=1}^J$). In a more abstract sense, we aim to determine the target distribution

P that maximizes $\mathbb{E}_{x\sim P}[f(x)]$ by having data sets of samples collected with J behavioral distributions $Q_{1:J} = \{Q_j\}_{j=1}^J$. In this section, we analyze the problem of defining an objective function suitable for this purpose.

The naïve approach would be to directly optimize the estimator $\widehat{\mu}^{\beta}_{P/Q_{1:J}}$ with the data sampled from $Q_1, \dots, Q_J$. This approach has a fundamental problem (even when using the BH). As shown in Section 3, the IS estimate is less reliable (i.e., displays a larger variance) for target distributions very different from the behavioral one. With enough freedom in choosing P, the optimal solution would assign as much probability mass as possible to the maximum value among $f(x_i)$. Since the IS estimator is clearly unreliable for such an extreme distribution, this kind of optimization is ill–informed and overconfident. For this reason, we adopt a risk–averse approach and we decide to optimize a statistical *lower bound* of the expected value $\mathbb{E}_{x\sim P}[f(x)]$ which holds with high confidence. We start by analyzing the behavior of the IS estimator, i.e., $J = 1$ and we provide the following result that bounds the variance of $\widehat{\mu}_{P/Q}$ in terms of the Rényi divergence.

Lemma 1 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q$. Let $\alpha \in [1, +\infty]$, $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ be i.i.d. random variables sampled from Q and $f : \mathcal{X} \to \mathbb{R}$ be a function with bounded $\frac{2\alpha}{\alpha-1}$–moment under Q ($\|f\|_{Q,\frac{2\alpha}{\alpha-1}} < +\infty$). Then, for any $N > 0$, the variance of the IS estimator $\widehat{\mu}_{P/Q}$ can be upper bounded as:*

$$\underset{\mathbf{x}\sim Q}{\mathbb{V}\mathrm{ar}}\left[\widehat{\mu}_{P/Q}\right] \leqslant \frac{1}{N}\|f\|^2_{Q,\frac{2\alpha}{\alpha-1}} d_{2\alpha}\left(P\|Q\right)^{2-\frac{1}{\alpha}}, \tag{13}$$

where we used the abbreviation $\mathbf{x} \sim Q$ for denoting $x_i \sim Q$ for all $i = 1, 2, ..., N$ all independent.

Proof We consider the following derivation:

$$\begin{aligned}
\underset{\mathbf{x}\sim Q}{\mathbb{V}\mathrm{ar}}\left[\widehat{\mu}_{P/Q}\right] &= \frac{1}{N}\underset{x_1\sim Q}{\mathbb{V}\mathrm{ar}}\left[\frac{p(x_1)}{q(x_1)}f(x_1)\right] && \text{(P.1)}\\
&\leqslant \frac{1}{N}\underset{x_1\sim Q}{\mathbb{E}}\left[\left(\frac{p(x_1)}{q(x_1)}f(x_1)\right)^2\right] && \text{(P.2)}\\
&\leqslant \frac{1}{N}\underset{x_1\sim Q}{\mathbb{E}}\left[\left|\frac{p(x_1)}{q(x_1)}\right|^{2\alpha}\right]^{\frac{1}{\alpha}}\underset{x_1\sim Q}{\mathbb{E}}\left[|f(x_1)|^{\frac{2\alpha}{\alpha-1}}\right]^{\frac{\alpha-1}{\alpha}} && \text{(P.3)}\\
&= \frac{1}{N}\|f\|^2_{Q,\frac{2\alpha}{\alpha-1}} d_{2\alpha}\left(P\|Q\right)^{2-\frac{1}{\alpha}},
\end{aligned}$$

where the line (P.1) follows form the fact that the random variables x_i are i.i.d., line (P.2) follows from bounding the variance with the second moment, and line (P.3) is derived by applying Hölder's inequality with $p = \alpha$ and $q = \frac{\alpha}{\alpha-1}$. Finally, we exploit the definition of d_α and $\|\cdot\|_{Q,p}$. ■

This result generalizes Lemma 4.1 of Metelli et al. (2018), that can be recovered by setting $\alpha = 1$ under the condition that $\|f\|_\infty < +\infty$:

$$\underset{\mathbf{x}\sim Q}{\mathbb{V}\mathrm{ar}}\left[\widehat{\mu}_{P/Q}\right] \leqslant \frac{1}{N}\|f\|^2_\infty d_2\left(P\|Q\right). \tag{14}$$

When $P = Q$ almost everywhere, we get $\mathbb{V}\mathrm{ar}_{\mathbf{x} \sim Q} \left[\widehat{\mu}_{Q/Q} \right] \leqslant \frac{1}{N} \|f\|_\infty^2$, a well–known upper bound to the variance of a Monte Carlo estimator. Recalling the definition of ESS (Equation 7) we can rewrite the previous bound as:

$$\underset{\mathbf{x} \sim Q}{\mathbb{V}\mathrm{ar}} \left[\widehat{\mu}_{P/Q} \right] \leqslant \frac{\|f\|_\infty^2}{\mathrm{ESS}(P\|Q)}. \tag{15}$$

Thus, the variance scales with ESS instead of N, justifying the definition of ESS. While $\widehat{\mu}_{P/Q}$ can have an unbounded variance even if f is bounded, the SN estimator $\widetilde{\mu}_{P/Q}$ is always bounded by $\|f\|_\infty$ and therefore it always has finite variance. Since the normalization term makes all the samples $\widetilde{w}_{P/Q}(x_i) f(x_i)$ interdependent, an exact analysis of its bias and variance is more challenging. Several works adopted approximate methods for providing an expression for its variance (Hesterberg, 1988). We propose an analysis of bias and variance of the SN estimator in Appendix D.

When considering the MIS estimator with BH, a similar bound for the variance was derived by Papini et al. (2019, Lemma 1):

Lemma 2 *Let P and $\{Q_j\}_{j=1}^J$ be probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q_j$ for $j = 1, \dots, J$. Let $\mathbf{x}_j = (x_{1j}, x_{2j}, \dots, x_{N_j j})^T$ be i.i.d. random variables sampled from Q_j for $j = 1, \dots, J$. Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J)^T$ and $\Phi = \sum_{k=1}^J \frac{N_k}{N} Q_k$ be a finite mixture. Let $\alpha \in [1, +\infty]$ and $f : \mathcal{X} \to \mathbb{R}$ be a function with bounded $\frac{2\alpha}{\alpha-1}$–moment under Φ ($\|f\|_{\Phi, \frac{2\alpha}{\alpha-1}} < +\infty$). Then, the variance of the multiple importance sampling estimator can be upper bounded as:*

$$\underset{\mathbf{x} \sim Q_{1:J}}{\mathbb{V}\mathrm{ar}} \left[\widehat{\mu}_{P/Q_{1:J}}^{\mathrm{BH}} \right] \leqslant \frac{1}{N} \|f\|_{\Phi, \frac{2\alpha}{\alpha-1}}^2 d_{2\alpha} \left(P \| \Phi \right)^{2 - \frac{1}{\alpha}}, \tag{16}$$

where we used the abbreviation $\mathbf{x} \sim Q_{1:J}$ for denoting $\mathbf{x}_j \sim Q_j$ for all $j = 1, 2, ..., J$ all independent.

Proof Consider the following derivation:

$$\begin{aligned}
\underset{\mathbf{x} \sim Q_{1:J}}{\mathbb{V}\mathrm{ar}} \left[\widehat{\mu}_{P/Q_{1:J}}^{\mathrm{BH}} \right] &= \underset{\mathbf{x} \sim Q_{1:J}}{\mathbb{V}\mathrm{ar}} \left[\frac{1}{N} \sum_{j=1}^J \sum_{i=1}^{N_j} \frac{p(x_{ij})}{\sum_{k=1}^K \frac{N_k}{N} q_k(x_{ij})} f(x_{ij}) \right] \\
&= \frac{1}{N^2} \sum_{j=1}^J \sum_{i=1}^{N_j} \underset{x_{ij} \sim Q_j}{\mathbb{V}\mathrm{ar}} \left[\frac{p(x_{ij})}{\sum_{k=1}^J \frac{N_k}{N} q_k(x_{ij})} f(x_{ij}) \right] && \text{(P.4)} \\
&\leqslant \frac{1}{N^2} \sum_{j=1}^J \sum_{i=1}^{N_j} \underset{x_{ij} \sim Q_j}{\mathbb{E}} \left[\left(\frac{p(x_{ij})}{\sum_{k=1}^J \frac{N_k}{N} q_k(x_{ij})} f(x_{ij}) \right)^2 \right] && \text{(P.5)} \\
&= \frac{1}{N} \underset{x \sim \Phi}{\mathbb{E}} \left[\left(\frac{p(x)}{\sum_{k=1}^J \frac{N_k}{N} q_k(x)} f(x) \right)^2 \right] && \text{(P.6)} \\
&\leqslant \frac{1}{N} \underset{x \sim \Phi}{\mathbb{E}} \left[\left| \frac{p(x)}{\sum_{k=1}^J \frac{N_k}{N} q_k(x)} \right|^{2\alpha} \right]^{\frac{1}{\alpha}} \underset{x \sim \Phi}{\mathbb{E}} \left[|f(x)|^{\frac{2\alpha}{\alpha-1}} \right]^{\frac{\alpha-1}{\alpha}} && \text{(P.7)}
\end{aligned}$$

$$= \frac{1}{N} \|f\|^2_{\Phi, \frac{2\alpha}{\alpha-1}} d_{2\alpha} \left(P \| \Phi\right)^{2-\frac{1}{\alpha}},$$

where line (P.4) follows from the fact that all x_{ij} are i.i.d., line (P.5) derives from bounding the variance with the second moment, line (P.6) is obtained from observing that, for a generic function g we have:

$$\frac{1}{N} \sum_{j=1}^{J} \sum_{i=1}^{N_j} \mathop{\mathbb{E}}_{x_{ij} \sim Q_j} [g(x_{ij})] = \sum_{j=1}^{J} \frac{N_j}{N} \mathop{\mathbb{E}}_{x_{1j} \sim Q_j} [g(x_{1j})] = \mathop{\mathbb{E}}_{x \sim \Phi} [g(x)].$$

Then, line (P.7) is obtained by Hölder's inequality with $p = \alpha$ and $q = \frac{\alpha}{\alpha-1}$. ■

Similarly to the single–IS case, an interesting case is obtained when setting $\alpha = 2$ and, consequently, requiring $\|f\|_\infty < +\infty$:

$$\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x} \sim Q_{1:J}} \left[\widehat{\mu}^{\mathrm{BH}}_{P/Q_{1:J}} \right] \leqslant \frac{1}{N} \|f\|^2_\infty d_2 \left(P \| \Phi\right). \tag{17}$$

While for Gaussian distributions the Rényi divergence can be computed in closed–form (Equation 5), when passing to the MIS case we need to evaluate the d_α between a Gaussian distribution and a mixture of Gaussians, which does not admit a closed form. A straightforward approach consists in exploiting the convexity of d_α w.r.t. to the second argument, when $\alpha \geqslant 1$, to obtain the loose bound:

$$d_\alpha(P \| \Phi) \leqslant \sum_{k=1}^{K} \frac{N_k}{N} d_\alpha(P \| Q_k).$$

However, this bound would be vacuous when at least one of the terms $d_\alpha(P \| Q_k)$ is infinite, while, clearly, the variance of the estimator would be finite as long as at least one of the terms $d_\alpha(P \| Q_k)$ is finite. This intuition is captured by a tighter bound that resorts to the harmonic mean of the terms $d_\alpha(P \| Q_k)$, as presented in Papini et al. (2019), which we report here using our notation.

Theorem 1 *(Papini et al., 2019, Theorem 5) Let P and $\{Q_j\}_{j=1}^J$ be probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q_j$ for $j = 1, \dots, J$. Let $\Phi = \sum_{j=1}^J \zeta_j Q_j$, with $\zeta_j \geqslant 0$ for all $j = 1, 2, ..., J$ and $\sum_{j=1}^J \zeta_j = 1$ be a finite mixture. Then, for any $\alpha \in [1, \infty]$, the exponentiated α–Rényi divergence can be bounded as:*

$$d_\alpha(P \| \Phi) \leqslant \frac{1}{\sum_{j=1}^J \frac{\zeta_j}{d_\alpha(P \| Q_j)}}. \tag{18}$$

We just need to set $\zeta_j = \frac{N_j}{N}$ in Theorem 1 to obtain the case of our interest. It is worth noting that Theorem 1 shows that the bound on the variance of the MIS estimator $\widehat{\mu}_{P/Q_{1:J}}$ with BH is never worse than the bound on the variance of the IS estimator $\widehat{\mu}_{P/Q_{j*}}$ that uses the distribution Q_{j*} which is the closest to P among the $Q_{1:J}$. Indeed, we can easily obtain the following inequality:

$$\frac{d_2(P \| \Phi)}{N} \leqslant \frac{1}{\sum_{j=1}^J \frac{N_j}{d_2(P \| Q_j)}} \leqslant \min_{j \in \{1, \dots, J\}} \frac{d_2(P \| Q_j)}{N_j}.$$

4.1. Concentration Inequality

The problem of finding a suitable concentration inequality for off–policy learning was studied by Thomas et al. (2015b) for off–line policy evaluation and subsequently by Thomas et al. (2015a) for optimization. On the one hand, fully empirical concentration inequalities, like Student–T, besides the asymptotic approximation, are not suitable in this case since the empirical variance needs to be estimated with importance sampling as well, injecting further uncertainty (Owen, 2013). On the other hand, several distribution–free inequalities, like Hoeffding, require knowing the maximum of the estimator, which might not exist for the IS estimator when $d_\infty(P\|Q) = \infty$. Constraining $d_\infty(P\|Q)$ to be finite often introduces unacceptable limitations. For instance, consider the case of univariate Gaussian distributions of the form $\mathcal{N}(\mu, \sigma^2)$, where the standard deviation σ is one of the parameters that must be learned. The constraint on $d_\infty(P\|Q)$ prevents a step that selects a target policy variance σ^2 larger than the behavioral one.[7] Even Bernstein inequalities (Bercu et al., 2015), are hardly applicable since, for instance, in the case of univariate Gaussian distributions, the importance weights display a *heavy–tail* behavior. For a detailed analysis of the properties of the IS estimator for Gaussian distributions refer to Appendix C. We believe that a reasonable trade–off should require the variance of the importance weights to be finite, which is equivalent to require $d_2(P\|Q) < \infty$, i.e., $\sigma_P < 2\sigma_Q$ for univariate Gaussians. For this reason, we resort to Chebyshev–like inequalities and we propose the following concentration bound derived from Cantelli's inequality and customized for the IS estimator.

Theorem 2 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ be i.i.d. random variables sampled from Q, and $f : \mathcal{X} \to \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $0 < \delta \leqslant 1$ and $N > 0$, with probability at least $1 - \delta$, it holds that:*

$$\mathop{\mathbb{E}}_{x\sim P}[f(x)] \geqslant \underbrace{\frac{1}{N}\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}_{\widehat{\mu}_{P/Q}} - \|f\|_\infty \sqrt{\frac{(1-\delta) d_2(P\|Q)}{\delta N}}. \tag{19}$$

Proof We start from Cantelli's inequality (Cantelli, 1929) applied on the random variable $\widehat{\mu}_{P/Q} = \frac{1}{N}\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)$:

$$\Pr\left(\widehat{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}[f(x)] \geqslant \lambda\right) \leqslant \frac{1}{1 + \frac{\lambda^2}{\mathbb{V}\mathrm{ar}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right]}}. \tag{P.8}$$

By renaming $\delta = \frac{1}{1 + \frac{\lambda^2}{\mathbb{V}\mathrm{ar}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right]}}$ and considering the complementary event, we get that with probability at least $1 - \delta$ we have:

$$\mathop{\mathbb{E}}_{x\sim P}[f(x)] \geqslant \widehat{\mu}_{P/Q} - \sqrt{\frac{1-\delta}{\delta} \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right]}. \tag{P.9}$$

[7] Although the policy variance tends to be reduced during the learning process, there might be cases in which it needs to be increased (e.g., suppose we start with a behavioral policy with small variance, it might be beneficial to increase the variance to enforce exploration). See (Ahmed et al., 2019; Papini et al., 2020) on this topic.

By replacing the variance with the bound in Lemma 1 (setting $\alpha = 1$) we get the result. ■

The bound highlights the interesting trade–off between the estimated performance and the uncertainty introduced by changing the distribution. The latter enters in the bound as the 2–Rényi divergence between the target distribution P and the behavioral distribution Q. Intuitively, we should trust the estimator $\widehat{\mu}_{P/Q}$ as long as P is not too far from Q. For the SN estimator, accounting for the bias, we are able to obtain a bound (reported in Appendix D), with a similar dependence on P as in Theorem 2, albeit with different constants. The same result is also applicable to the multiple importance sampling estimator, by just swapping $\widehat{\mu}_{P/Q}$ with $\widehat{\mu}^{\mathrm{BH}}_{P/Q_{1:J}}$ and using the variance bound from Lemma 2.

Corollary 1 *Let P and $\{Q_j\}_{j=1}^{J}$ be probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q_j$ for $j = 1, \dots, J$. Let $x_{1j}, x_{2j}, \dots, x_{N_j j}$ be i.i.d. random variables sampled from Q_j with $j = 1, 2, ..., J$, let $\Phi = \sum_{k=1}^{J} \frac{N_k}{N} Q_k$ be a finite mixture such that $d_2(P \| \Phi) < +\infty$ and $f : \mathcal{X} \to \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $0 < \delta \leqslant 1$ and $N > 0$, with probability at least $1 - \delta$, it holds that:*

$$\mathop{\mathbb{E}}_{x \sim P}[f(x)] \geqslant \underbrace{\frac{1}{N} \sum_{j=1}^{J} \sum_{i=1}^{N_j} w^{\mathrm{BH}}_{P/Q_{1:J}}(x_{ij}) f(x_{ij})}_{\widehat{\mu}^{\mathrm{BH}}_{P/Q_{1:J}}} - \|f\|_\infty \sqrt{\frac{(1-\delta) d_2(P \| \Phi)}{\delta N}}. \tag{20}$$

By renaming the constants involved in the bound of Theorem 2 as $\lambda = \|f\|_\infty \sqrt{(1-\delta)/\delta}$, we get a surrogate objective function. In the following sections, we will denote it with $\mathcal{L}$ and particularize it for the action–based and parameter–based frameworks. Bound optimization can be carried out in different ways. Section 4.2 shows why using the natural gradient could be a successful choice in case P and Q can be expressed as differentiable parametric distributions.

Remark 3 (On Importance Weight Clipping) In Section 4.1, we have seen that one of the main challenges in employing importance sampling is the undesirable heavy–tailed behavior. A common technique for overcoming this problem is *weight clipping* (or *truncation*) (Ionides, 2008). More specifically, given a clipping threshold $M < \infty$, we define the clipped weight as $\breve{\omega}_{P/Q}(x) = \min\{M, \omega_{P/Q}(x)\}$, leading to the estimator:

$$\breve{\mu}_{P/Q} = \frac{1}{N} \sum_{i=1}^{N} \breve{\omega}_{P/Q}(x_i) f(x_i) = \frac{1}{N} \sum_{i=1}^{N} \min\{M, \omega_{P/Q}(x_i)\} f(x_i). \tag{21}$$

Clearly, truncating the weights introduces a bias that, under certain conditions (e.g., $f(x) \geqslant 0$ for all $x \in \mathcal{X}$), can be neglected to derive one–sided concentration inequalities (Thomas et al., 2015b). In any case, the bias and the variance of $\breve{\mu}_{P/Q}$ can be bounded as a function of the Rényi divergence and the clipping threshold (Papini et al., 2019, see also Lemma 3):

$$\left| \mathop{\mathbb{E}}_{\mathbf{x} \sim Q}\left[\breve{\mu}_{P/Q}\right] - \mathop{\mathbb{E}}_{x \sim P}[f(x)] \right| \leqslant \|f\|_\infty \frac{d_2(P \| Q)}{M},$$

$$\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x} \sim Q}\left[\breve{\mu}_{P/Q}\right] \leqslant \|f\|_\infty^2 \frac{d_2(P \| Q)}{N}.$$

As also supported by intuition, the form of the bias and the variance suggests that the clipping threshold should be *adaptive* and a function of the number of samples N. The following result adapts Theorem 1 by Papini et al. (2019) showing that, by making the threshold M dependent on the number of samples N and on the probability δ, we are able to achieve exponential concentration, compared to the polynomial concentration of Theorem 2.

Theorem 3 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q$ and $d_2(P \| Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ be i.i.d. random variables sampled from Q, and $f : \mathcal{X} \to \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $0 < \delta \leqslant 1$ and $N > 0$, using a clipping threshold $M(N, \delta) = \sqrt{\frac{3N d_2(P\|Q)}{2 \log \frac{1}{\delta}}}$, with probability at least $1 - \delta$ it holds that:*

$$\mathop{\mathbb{E}}_{x \sim P}[f(x)] \geqslant \underbrace{\frac{1}{N} \sum_{i=1}^{N} \check{w}_{P/Q}(x_i) f(x_i)}_{\check{\mu}_{P/Q}} - \|f\|_\infty (2 + \sqrt{3}) \sqrt{\frac{2 d_2(P\|Q) \log \frac{1}{\delta}}{3N}}. \tag{22}$$

Despite the more convenient dependence on δ, we believe that weight clipping is unsuited for our purposes. First, in order to set the clipping threshold $M(N, \delta)$, it is necessary to know in advance the confidence δ. Second, and most importantly, weight clipping makes the objective function non–differentiable w.r.t. the policy/hyperpolicy parameters, due to the presence of minimum $\min\{M, \omega_{P/Q}(x)\}$, preventing gradient–based optimization. Thus, while clipping is a viable alternative in the case of off–distribution evaluation, it introduces significant challenges when it comes to off–distribution optimization. Indeed, in (Papini et al., 2019) the optimization of the clipped estimator is carried out, only approximately, by discretizing the space of hyperpolicy parameters. For these reasons, in this work, we will not deepen the study of weight clipping.

4.2. Importance Sampling and Natural Gradient

We can look at a parametric distribution $P_{\boldsymbol{\omega}}$, having $p_{\boldsymbol{\omega}}$ as a density function, as a point on a probability manifold with coordinates $\boldsymbol{\omega} \in \Omega$. If $p_{\boldsymbol{\omega}}$ is differentiable, the Fisher Information Matrix (FIM, Rao, 1992; Amari, 2012) is defined as:

$$\mathcal{F}(\boldsymbol{\omega}) = \int_{\mathcal{X}} p_{\boldsymbol{\omega}}(x) \nabla_{\boldsymbol{\omega}} \log p_{\boldsymbol{\omega}}(x) \nabla_{\boldsymbol{\omega}} \log p_{\boldsymbol{\omega}}(x)^T \, \mathrm{d}x.$$

This matrix is, up to a scale, an invariant metric (Amari, 1998) on parameter space Ω, i.e., $(\boldsymbol{\omega}' - \boldsymbol{\omega})^T \mathcal{F}(\boldsymbol{\omega})(\boldsymbol{\omega}' - \boldsymbol{\omega})$ is independent from the specific parameterization and provides a second–order approximation of the distance between $p_{\boldsymbol{\omega}}$ and $p_{\boldsymbol{\omega}'}$ on the probability manifold up to a scale factor. Given a loss function $\mathcal{L}(\boldsymbol{\omega})$, we define the natural gradient (Amari, 1998; Kakade, 2002) as $\widetilde{\nabla}_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega}) = \mathcal{F}(\boldsymbol{\omega})^{-1} \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega})$, whenever $\mathcal{F}(\boldsymbol{\omega})$ is non–singular, which represents the steepest ascent direction in the probability manifold. Thanks to the invariance property, there is a tight connection between the geometry induced by the Rényi divergence and the Fisher information metric (Amari and Cichocki, 2010).

Theorem 4 *Let $p_{\boldsymbol{\omega}}$ be a p.d.f. differentiable w.r.t. $\boldsymbol{\omega} \in \Omega$. Then, it holds that, for the Rényi divergence:*

$$D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}}) = \frac{\alpha}{2} \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right)^T \mathcal{F}(\boldsymbol{\omega}) \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right) + o(\|\boldsymbol{\omega}' - \boldsymbol{\omega}\|_2^2),$$

and for the exponentiated Rényi divergence:

$$d_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}}) = 1 + \frac{\alpha}{2} \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right)^T \mathcal{F}(\boldsymbol{\omega}) \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right) + o(\|\boldsymbol{\omega}' - \boldsymbol{\omega}\|_2^2).$$

This result provides an approximate expression for the variance of the importance weights:

$$\underset{x \sim p_{\boldsymbol{\omega}}}{\mathbb{V}\mathrm{ar}} \left[w_{\boldsymbol{\omega}'/\boldsymbol{\omega}}(x)\right] = d_2(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}}) - 1 \simeq \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right)^T \mathcal{F}(\boldsymbol{\omega}) \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right), \tag{23}$$

which can be used to justify the use of natural gradients in off–distribution optimization. Say we want to find the steepest descent update for $\boldsymbol{\omega}$ that keeps the variance of the importance weights under control:

$$\begin{aligned} \max_{\Delta\boldsymbol{\omega}} \quad & \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega})^T \Delta\boldsymbol{\omega} \\ \text{subject to} \quad & \underset{x \sim p_{\boldsymbol{\omega}}}{\mathbb{V}\mathrm{ar}} \left[w_{\boldsymbol{\omega}'/\boldsymbol{\omega}}(x)\right] \leqslant \epsilon^2, \end{aligned}$$

for some small $\epsilon > 0$. By approximating the variance with Equation (23) and solving the resulting constrained optimization problem we obtain:

$$\Delta\boldsymbol{\omega} = \frac{\epsilon}{\sqrt{\nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega})^T \mathcal{F}(\boldsymbol{\omega})^{-1} \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega})}} \mathcal{F}(\boldsymbol{\omega})^{-1} \nabla_{\boldsymbol{\omega}} \mathcal{L}(\boldsymbol{\omega}), \tag{24}$$

which is precisely the natural gradient update with adaptive step size (Amari, 1998; Matsubara et al., 2010).

5. Policy Optimization via Importance Sampling

In this section, we discuss how to customize the bound provided in Theorem 2 (and Corollary 1) for policy optimization. We start with presenting *Policy Optimization via Importance Sampling* (POIS, Metelli et al., 2018), a model–free actor–only policy search algorithm in its two flavors: *Parameter–based POIS* (P-POIS, Section 5.1), which adopts the PGPE framework, and *Action–based POIS* (A-POIS, Section 5.2), which is based on a policy gradient approach. Then, we show how to extend these algorithms to the MIS framework. Finally, for the action–based case, we introduce the PDIS, proposing a new algorithm called *per–Decision action–based POIS* (D-POIS, Section 5.3). A more detailed description of the implementation aspects is reported in Appendix G.

5.1. Parameter–based POIS

In the Parameter–based POIS (P-POIS, Figure 1) we consider a parametrized policy space $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$, with $\pi_{\boldsymbol{\theta}}$ not necessarily differentiable nor stochastic. The policy parameters $\boldsymbol{\theta}$ are sampled at the beginning of each episode from a parametric hyperpolicy $\nu_{\boldsymbol{\rho}}$ selected in a parametric space $\mathcal{N}_{\mathcal{P}} = \{\nu_{\boldsymbol{\rho}} : \boldsymbol{\rho} \in \mathcal{P} \subseteq \mathbb{R}^r\}$, which needs to be stochastic and differentiable in $\boldsymbol{\rho}$. The goal is to learn the *hyperparameters* $\boldsymbol{\rho}$ so as to maximize $J_{\mathcal{M}}(\boldsymbol{\rho})$

as in Equation (1). In this setting, the distributions Q and P of Section 4 correspond to the behavioral $\nu_{\boldsymbol{\rho}}$ and target $\nu_{\boldsymbol{\rho}'}$ hyperpolicies, while f is the trajectory return $R(\tau)$. The importance weights must take into account all sources of randomness, derived from sampling a policy parameter $\boldsymbol{\theta}$ and a trajectory τ (Zhao et al., 2013):

$$w_{\boldsymbol{\rho}'/\boldsymbol{\rho}}(\boldsymbol{\theta}) = \frac{\nu_{\boldsymbol{\rho}'}(\boldsymbol{\theta})p(\tau|\boldsymbol{\theta})}{\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta})p(\tau|\boldsymbol{\theta})} = \frac{\nu_{\boldsymbol{\rho}'}(\boldsymbol{\theta})}{\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta})}.$$

Notice that, from the uniform bound on the immediate reward, it follows that the trajectory return is bounded by $R_{\max}\frac{1-\gamma^H}{1-\gamma}$ if $\gamma < 1$ and $R_{\max}H$ if $\gamma = 1$. We can now rephrase Theorem 2 for P-POIS, getting to the surrogate objective function:

$$\mathcal{L}_{\lambda}^{\mathrm{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho}) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} w_{\boldsymbol{\rho}'/\boldsymbol{\rho}}(\boldsymbol{\theta}_i)R(\tau_i)}_{\hat{J}_{\mathcal{M}}^{\mathrm{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho})} - \lambda\sqrt{\frac{d_2\left(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}}\right)}{N}}, \tag{25}$$

where λ is a regularization parameter[8] and each trajectory τ_i is obtained by running an episode with action policy $\pi_{\boldsymbol{\theta}_i}$, and the corresponding policy parameters $\boldsymbol{\theta}_i$ are sampled independently from hyperpolicy $\nu_{\boldsymbol{\rho}}$ at the beginning of each episode $i = 1, 2, ..., N$.

When moving to the MIS framework with balance heuristic, we need to redefine the importance weight accounting for the several behavioral hyperpolicies considered, having hyperparameters $\boldsymbol{\rho}_{1:J} = \{\boldsymbol{\rho}_j\}_{j=1}^{J}$:

$$w_{\boldsymbol{\rho}'/\boldsymbol{\rho}_{1:J}}^{\mathrm{BH}}(\boldsymbol{\theta}) = \frac{\nu_{\boldsymbol{\rho}'}(\boldsymbol{\theta})p(\tau|\boldsymbol{\theta})}{\sum_{k=1}^{J}\frac{N_k}{N}\nu_{\boldsymbol{\rho}_k}(\boldsymbol{\theta})p(\tau|\boldsymbol{\theta})} = \frac{\nu_{\boldsymbol{\rho}'}(\boldsymbol{\theta})}{\sum_{k=1}^{J}\frac{N_k}{N}\nu_{\boldsymbol{\rho}_k}(\boldsymbol{\theta})}.$$

Therefore, by employing Corollary 1 together with the bound on the Rényi divergence (Theorem 1), we are able to formulate the new objective function:

$$\mathcal{L}_{\lambda}^{\mathrm{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho}_{1:J}) = \underbrace{\frac{1}{N}\sum_{j=1}^{J}\sum_{i=1}^{N_j} w_{\boldsymbol{\rho}'/\boldsymbol{\rho}_{1:J}}^{\mathrm{BH}}(\boldsymbol{\theta}_{ij})R(\tau_{ij})}_{\hat{J}_{\mathcal{M}}^{\mathrm{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho}_{1:J})} - \frac{\lambda}{\sqrt{\sum_{j=1}^{J}\frac{N_j}{d_2\left(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}_j}\right)}}}, \tag{26}$$

where each $\boldsymbol{\theta}_{ij}$ is sampled independently from $\nu_{\boldsymbol{\rho}_j}$ and the corresponding trajectory τ_j is obtained by running policy $\pi_{\boldsymbol{\theta}_{ij}}$ in the environment with $i = 1, 2, ..., N_j$ and $j = 1, 2, ..., J$. Clearly, the objective function in Equation (26) reduces to Equation (25) when setting $J = 1$, i.e., when considering a single behavioral hyperpolicy.

To derive a practical algorithm, we use as behavioral hyperpolicies the J most recent hyperpolicies and we denote them with $\boldsymbol{\rho}_{1:J}$. At each *on–line iteration* $h = 1, 2, ..., M_{\text{on-line}}$, we sample N_J parameters $\{\boldsymbol{\theta}_i^h\}_{i=1}^{N_J}$ independently from $\nu_{\boldsymbol{\rho}_0^h}$. For each of the $\boldsymbol{\theta}_i^h$, we collect a single trajectory τ_i^h by running policy $\pi_{\boldsymbol{\theta}_i^h}$ in the environment and we observe its return $R(\tau_i^h)$. We now employ this return and all the ones previously collected to optimize the objective function $\mathcal{L}_{\lambda}^{\mathrm{P-POIS}}$ off–line. In particular, for each *off–line iteration* $k = 1, 2, ..., M_{\text{off-line}}$, we

[8] Formally, from Theorem 2, $\lambda = R_{\max}\frac{1-\gamma^H}{1-\gamma}\sqrt{\frac{1-\delta}{\delta}}$ for $\gamma < 1$ and $\lambda = R_{\max}H\sqrt{\frac{1-\delta}{\delta}}$ for $\gamma = 1$.

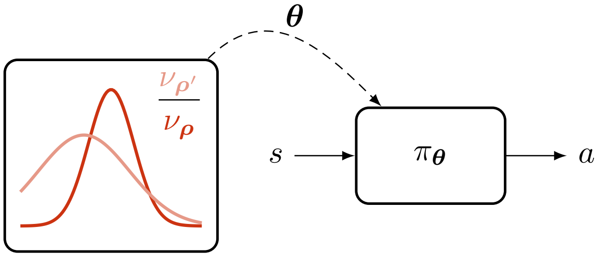

$\boldsymbol{\theta}$
$\frac{\nu_{\boldsymbol{\rho}'}}{\nu_{\boldsymbol{\rho}}}$
s
$\pi_{\boldsymbol{\theta}}$
a

Figure 1: Graphical representation of P-POIS.

compute the gradient $\nabla_{\boldsymbol{\rho}_k^h} \mathcal{L}_\lambda^{\text{P-POIS}}(\boldsymbol{\rho}_k^h/\boldsymbol{\rho}_{1:J}^h)$ of the objective function and we determine a step size α_k using a *line search* procedure (see Appendix G.1). We employ them to update the hyperpolicy parameters, via gradient ascent:

$$\boldsymbol{\rho}_{k+1}^h = \boldsymbol{\rho}_k^h + \alpha_k \nabla_{\boldsymbol{\rho}_k^h} \mathcal{L}_\lambda^{\text{P-POIS}}(\boldsymbol{\rho}_k^h/\boldsymbol{\rho}_{1:J}^h).$$

Finally, when the off–line optimization is performed, we update the set of behavioral hyperpolicies by removing the oldest parametrization $\boldsymbol{\rho}_0^{h-J}$ and inserting the most recent $\boldsymbol{\rho}_0^{h+1}$. Clearly, the removal of the oldest one needs to be performed only if we have performed at least J on–line iterations, i.e., if $h \geqslant J$. Refer to Algorithm 1 for the complete pseudo–code of P-POIS.

Remark 4 (How to choose the hyperpolicy model?) The choice of the hyperpolicy model influences the computation of the objective function. Often a Gaussian hyperpolicy $\nu_{\boldsymbol{\rho}}$ with diagonal covariance matrix is used, i.e., $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\rho}}, \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\rho}}^2))$ with hyperparameters $\boldsymbol{\rho}$. The policy is typically chosen as deterministic: $\pi_{\boldsymbol{\theta}}(a|s) = \delta_{u_{\boldsymbol{\theta}}(s)}(a)$, where $u_{\boldsymbol{\theta}}$ is a deterministic function of the state s (e.g., Sehnke et al., 2010; Grüttner et al., 2010). This particular setting has an advantage over the action–based setting (Section 5.2), since the distribution of the importance weights is entirely known, being the ratio between two Gaussians, and the Rényi divergence $d_2(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}})$ can be computed exactly (Burbea, 1984, , see Equation 5). The parameter–based approach has another key advantage. Indeed the FIM can be computed exactly, and it is diagonal in the case of a Gaussian hyperpolicy with a diagonal covariance matrix:

$$\mathcal{F}(\boldsymbol{\rho}) = \left(\begin{array}{c|c} \operatorname{diag}(\boldsymbol{\sigma}_{\boldsymbol{\rho}})^{-2} & \mathbf{0} \\ \hline \mathbf{0} & 2\mathbf{I} \end{array}\right).$$

The FIM is block–diagonal in the more general case of a Gaussian hyperpolicy, as observed in Miyamae et al. (2010). This makes the natural gradient much more enticing for P-POIS. The natural gradient can be obtained by simply premultiplying the gradient of the objective function by the inverse of the FIM: $\mathcal{F}(\boldsymbol{\rho}_k^h)^{-1} \nabla_{\boldsymbol{\rho}_k^h} \mathcal{L}_\lambda^{\text{P-POIS}}(\boldsymbol{\rho}_k^h/\boldsymbol{\rho}_{1:J}^h)$.

5.2. Action–based POIS

In Action–based POIS (A-POIS, Figure 2) we search for a policy that maximizes the performance index $J_{\mathcal{M}}(\boldsymbol{\theta})$ within a parametric space $\Pi_\Theta = \{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p\}$ of stochastic differentiable policies. In this context, the behavioral (resp. target) distribution Q (resp. P)

Algorithm 1 Parameter–based POIS.

```
Input:  $J$ number of behavioral hyperpolicies
        $N_J$ number of samples to collect for each hyperpolicy
        $M_{\text{on-line}}$ maximum number of on–line iterations
        $M_{\text{off-line}}$ maximum number of off–line iterations
        $\lambda \geqslant 0$ regularization parameter
Output: $\boldsymbol{\rho}_0^{M_{\text{on-line}}}$ final hyperpolicy parametrization
Initialize the behavioral hyperpolicy $\boldsymbol{\rho}_0^0$ arbitrarily
Initialize the behavioral hyperpolicy set $\boldsymbol{\rho}_{1:J}^0 = \{\boldsymbol{\rho}_0^0\}$
for $h = 0, 1, ..., M_{\text{on-line}} - 1$ do                                On–line optimization
    Sample $N_J$ policy parameters $\{\boldsymbol{\theta}_i^h\}_{i=1}^{N_J}$ independently from $\nu_{\boldsymbol{\rho}_0^h}$
    Sample $N_J$ trajectories $\{\tau_i^h\}_{i=1}^{N_J}$ independently with each $\{\pi_{\boldsymbol{\theta}_i^h}\}_{i=1}^{N_J}$
    for $k = 0, 1, ..., M_{\text{off-line}} - 1$ do                           Off–line optimization
        Compute the objective function gradient $\nabla_{\boldsymbol{\rho}_k^h} \mathcal{L}_\lambda^{\text{P-POIS}}(\boldsymbol{\rho}_k^h / \boldsymbol{\rho}_{1:J}^h)$
        Find the step size $\alpha_k^h$ using line search
        Update the hyperpolicy parameters $\boldsymbol{\rho}_{k+1}^h = \boldsymbol{\rho}_k^h + \alpha_k^h \nabla_{\boldsymbol{\rho}_k^h} \mathcal{L}_\lambda^{\text{P-POIS}}(\boldsymbol{\rho}_k^h / \boldsymbol{\rho}_{1:J}^h)$
    end for
    Update the last behavioral hyperpolicy $\boldsymbol{\rho}_0^{h+1} = \boldsymbol{\rho}_{M_{\text{off-line}}}^h$
    Update the behavioral hyperpolicy set $\boldsymbol{\rho}_{1:J}^{h+1} = \begin{cases} \left(\boldsymbol{\rho}_{1:J}^h \setminus \{\boldsymbol{\rho}_0^{h-J}\}\right) \cup \{\boldsymbol{\rho}_0^{h+1}\} & \text{if } h \geqslant J \\ \boldsymbol{\rho}_{1:J}^h \cup \{\boldsymbol{\rho}_0^{h+1}\} & \text{otherwise} \end{cases}$
end for
```

becomes the distribution over trajectories $p(\cdot|\boldsymbol{\theta})$ (resp. $p(\cdot|\boldsymbol{\theta}'))$ induced by the behavioral policy $\pi_{\boldsymbol{\theta}}$ (resp. target policy $\pi_{\boldsymbol{\theta}'}$) and f is again the trajectory return $R(\tau)$. The corresponding importance weight is defined in terms of trajectory density functions, and reduces to a product of policy ratios:

$$w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) = \frac{p(\tau|\boldsymbol{\theta}')}{p(\tau|\boldsymbol{\theta})} = \prod_{t=0}^{H-1} \frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t}|s_{\tau,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})}.$$

The Rényi divergence has to be computed between the distributions over trajectories induced by the policies, leading to the surrogate objective function:

$$\mathcal{L}_\lambda^{\text{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \underbrace{\frac{1}{N}\sum_{i=1}^{N} w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i) R(\tau_i)}_{\widehat{J}_{\mathcal{M}}^{\text{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})} - \lambda \sqrt{\frac{d_2\left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta})\right)}{N}}, \tag{27}$$

Differently from P-POIS, the surrogate objective function cannot be directly optimized via gradient ascent since computing $d_2\left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta})\right)$ requires the approximation of an integral over the trajectory space, even for well–known policy models (like Gaussian policies). Furthermore, for stochastic environments, we need to know the functional form of the transition model P:

$$d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) = \int_{\mathcal{T}} p(\tau|\boldsymbol{\theta}) \left(\frac{p(\tau|\boldsymbol{\theta}')}{p(\tau|\boldsymbol{\theta})}\right)^2 \mathrm{d}\tau = \int_{\mathcal{T}} p(\tau|\boldsymbol{\theta}) \left(\prod_{t=0}^{H-1} \frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t}|s_{\tau,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})}\right)^2 \mathrm{d}\tau. \quad (28)$$

Nevertheless, we can upper bound $d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)$ with the Rényi divergence between the policies, as provided by the following result.

Proposition 1 *Let $p(\cdot|\boldsymbol{\theta})$ and $p(\cdot|\boldsymbol{\theta}')$ be the behavioral and target trajectory probability density functions. If $p(\cdot|\boldsymbol{\theta}') \ll p(\cdot|\boldsymbol{\theta})$ and $H < +\infty$, then, for any $\alpha \in [0, +\infty]$ it holds that:*

$$d_\alpha\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) \leqslant \sup_{s\in\mathcal{S}} \left\{d_\alpha\left(\pi_{\boldsymbol{\theta}'}(\cdot|s)\|\pi_{\boldsymbol{\theta}}(\cdot|s)\right)\right\}^H .$$

However, this bound, besides being hard to compute due to the presence of the supremum, is extremely conservative since the Rényi divergence is raised to the horizon H. For these reasons, in practice, we resort to *Rényi divergence estimators* that will be presented and discussed in Remark 6.

The multiple importance sampling extension is straightforward, provided that we redefine the importance weight according to the balance heuristic, considering the set of behavioral policies induced by the corresponding parameters $\boldsymbol{\theta}_{1:J} = \{\boldsymbol{\theta}_j\}_{j=1}^J$:

$$w^{\mathrm{BH}}_{\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J}}(\tau) = \frac{p(\tau|\boldsymbol{\theta}')}{\sum_{k=1}^J \frac{N_k}{N} p(\tau|\boldsymbol{\theta}_j)} = \frac{\prod_{t=0}^{H-1} \pi_{\boldsymbol{\theta}'}(a_{\tau,t}|s_{\tau,t})}{\sum_{k=1}^J \prod_{t=0}^{H-1} \frac{N_k}{N} \pi_{\boldsymbol{\theta}_j}(a_{\tau,t}|s_{\tau,t})}.$$

Given the importance weights, we can apply Corollary 1 and Theorem 1 in order to define the surrogate objective function:

$$\mathcal{L}_\lambda^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J}) = \underbrace{\frac{1}{N}\sum_{j=1}^{J}\sum_{i=1}^{N_j} w^{\mathrm{BH}}_{\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J}}(\tau_{ij}) R(\tau_{ij})}_{\widehat{J}_{\mathcal{M}}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J})} - \frac{\lambda}{\sqrt{\sum_{j=1}^J \frac{N_j}{d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta}_j)\right)}}}, \quad (29)$$

where each τ_{ij} is obtained by running policy $\pi_{\boldsymbol{\theta}_j}$ in the environment with $i = 1, 2, ..., N_j$ and $j = 1, 2, ..., J$.

The learning process proceeds in a way similar to P-POIS. We consider the J most recent policies $\boldsymbol{\theta}_{1:J}$ as behavioral policies. At each *on–line iteration* $h = 1, 2, ..., M_{\text{on-line}}$, we collect N_J trajectories $\{\theta_i^h\}_{i=1}^{N_J}$ independently from $\pi_{\boldsymbol{\theta}_0^h}$ and we observe their return $R(\tau_i^h)$. These trajectories are then used to perform off–line optimization of the objective function $\mathcal{L}_\lambda^{\mathrm{A-POIS}}$. More specifically, for each *off–line iteration* $k = 1, 2, ..., M_{\text{off-line}}$, we compute the gradient $\nabla_{\boldsymbol{\theta}_k^h} \mathcal{L}_\lambda^{\mathrm{A-POIS}}(\boldsymbol{\theta}_k^h/\boldsymbol{\theta}_{1:J}^h)$ of the objective function and we determine a step size using a *line search* procedure (see Appendix G.1). The policy parametrization is then updated via gradient ascent:

$$\boldsymbol{\theta}_{k+1}^h = \boldsymbol{\theta}_k^h + \alpha_k \nabla_{\boldsymbol{\theta}_k^h} \mathcal{L}_\lambda^{\mathrm{A-POIS}}(\boldsymbol{\theta}_k^h/\boldsymbol{\theta}_{1:J}^h).$$

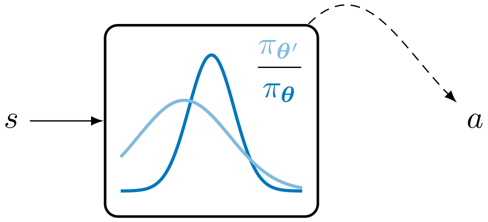

s
$\frac{\pi_{\boldsymbol{\theta}'}}{\pi_{\boldsymbol{\theta}}}$
a

Figure 2: Graphical representation of A-POIS.

Finally, we update the set of behavioral policies by inserting the new parametrization $\boldsymbol{\theta}_0^{h+1}$ and, if $h \geqslant J$, removing the oldest one $\boldsymbol{\theta}_0^{h-J}$. Refer to Algorithm 2 for the complete pseudo–code of A-POIS.

Remark 5 (How to choose the policy model?) Typically, we consider a policy $\pi_{\boldsymbol{\theta}}(\cdot|s)$ defined as a Gaussian distribution over actions whose mean depends on the state and whose covariance is state–independent and diagonal, i.e., $a \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}(s), \mathrm{diag}(\sigma^2_{\boldsymbol{\theta}}))$, where $\boldsymbol{\theta}$ are the parameters we need to optimize. However, even in this case, we cannot exactly compute the exponentiated Rényi divergence between the trajectory probability distributions (Equation 28). Furthermore, w.r.t. P-POIS, the usage of natural gradient becomes less appealing for A-POIS. Indeed, the FIM needs to be estimated off–policy from samples, possibly injecting further uncertainty and betraying its original goal. For instance, in the single–IS setting, we can employ the estimator:

$$\widehat{\mathcal{F}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i) \left(\sum_{t=0}^{H-1} \nabla_{\boldsymbol{\theta}'} \log \pi_{\boldsymbol{\theta}'}(a_{\tau_i,t}|s_{\tau_i,t}) \right)^T \left(\sum_{t=0}^{H-1} \nabla_{\boldsymbol{\theta}'} \log \pi_{\boldsymbol{\theta}'}(a_{\tau_i,t}|s_{\tau_i,t}) \right).$$

The SN estimator is obtained by replacing $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i)$ with $\widetilde{w}_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i)$. These estimators become very unreliable when $\boldsymbol{\theta}'$ is far from $\boldsymbol{\theta}$, making them difficult to use in practice.

Remark 6 (Estimating the Rényi Divergence) Since the exponentiated Rényi divergence $d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)$ between distributions over trajectories, as in Equation (28), cannot be computed exactly in A-POIS, we address the problem of how to estimate it. For brevity, we denote with $\mathfrak{D}_\alpha = \sup_{s\in\mathcal{S}} \left\{ d_\alpha\left(\pi_{\boldsymbol{\theta}'}(\cdot|s)\|\pi_{\boldsymbol{\theta}}(\cdot|s)\right)\right\}^{\alpha-1}$ for $\alpha \in [0, +\infty]$.[9] The simplest and most natural estimator can be obtained by computing the second sample moment of the importance weights, i.e., rephrasing Equation (28) in a sample–based version:

$$\widehat{d}_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) = \frac{1}{N}\sum_{i=1}^{N} w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i)^2 = \frac{1}{N}\sum_{i=1}^{N}\prod_{t=0}^{H-1}\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau_i,t}|s_{\tau_i,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau_i,t}|s_{\tau_i,t})}\right)^2. \tag{30}$$

This estimator is clearly unbiased, but it tends to display high variance. In particular, it is affected by a very undesirable property: when all the importance weights $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau)$ are close to zero we get a significant underestimation of the divergence. This is a symptom of the fact that the two distributions, $p(\cdot|\boldsymbol{\theta}')$ and $p(\cdot|\boldsymbol{\theta})$, are quite far away. Thus, their divergence should be large, but we estimate a value close to zero.

[9] $\mathfrak{D}_\alpha$ represents the supremum over the state space $\mathcal{S}$ of the α–moment of the policy ratio $\frac{\pi_{\boldsymbol{\theta}'}(\cdot|s)}{\pi_{\boldsymbol{\theta}}(\cdot|s)}$.

Algorithm 2 Action–based POIS.

```
Input:  $J$ number of behavioral policies
        $N_J$ number of samples to collect for each policy
        $M_{\text{on-line}}$ maximum number of on–line iterations
        $M_{\text{off-line}}$ maximum number of off–line iterations
        $\lambda \geqslant 0$ regularization parameter
Output: $\boldsymbol{\theta}_0^{M_{\text{on-line}}}$ final policy parametrization
Initialize the behavioral policy $\boldsymbol{\theta}_0^0$ arbitrarily
Initialize the behavioral policy set $\boldsymbol{\theta}_{1:J}^0 = \{\boldsymbol{\theta}_0^0\}$
for $h = 0, 1, ..., M_{\text{on-line}} - 1$ do                                  On–line optimization
    Sample $N_J$ trajectories $\{\tau_i^h\}_{i=1}^{N_J}$ independently from $\pi_{\boldsymbol{\theta}_0^h}$
    for $k = 0, 1, ..., M_{\text{off-line}} - 1$ do                             Off–line optimization
        Compute the objective function gradient $\nabla_{\boldsymbol{\theta}_k^h} \mathcal{L}_\lambda^{\text{A-POIS}}(\boldsymbol{\theta}_k^h / \boldsymbol{\theta}_{1:J}^h)$
        Find the step size $\alpha_k^h$ using line search
        Update the hyperpolicy parameters $\boldsymbol{\theta}_{k+1}^h = \boldsymbol{\theta}_k^h + \alpha_k \nabla_{\boldsymbol{\theta}_k^h} \mathcal{L}_\lambda^{\text{A-POIS}}(\boldsymbol{\theta}_k^h / \boldsymbol{\theta}_{1:J}^h)$
    end for
    Update the last behavioral policy $\boldsymbol{\theta}_0^{h+1} = \boldsymbol{\theta}_{M_{\text{off-line}}}^h$
    Update the behavioral policy set $\boldsymbol{\theta}_{1:J}^{h+1} = \begin{cases} \left(\boldsymbol{\theta}_{1:J}^h \backslash \{\boldsymbol{\theta}_0^{h-J}\}\right) \cup \{\boldsymbol{\theta}_0^{h+1}\} & \text{if } h \geqslant J \\ \boldsymbol{\theta}_{1:J}^h \cup \{\boldsymbol{\theta}_0^{h+1}\} & \text{otherwise} \end{cases}$
end for
```

We can mitigate this problem by incorporating the fact that the mean of the importance weights is known to be 1. This observation leads to the estimator:

$$\check{d}_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) = 1 + \frac{1}{N}\sum_{i=1}^{N}\left(w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i) - 1\right)^2. \tag{31}$$

Like the previous one, this estimator remains unbiased, but its minimum value is now 1 (notice that the divergence is never lower than 1). Indeed, when all the importance weights are close to zero, the estimated value is close to 2, instead of zero.[10] For both these estimators, the variance is proportional to the fourth moment of the importance weight distribution, i.e., $\mathcal{O}\left(\frac{1}{N}\mathfrak{D}_4^H\right)$. Recalling the spirit behind the concentration inequality of Theorem 2, this fact is undesirable since the 4–Rényi divergence of the importance weights might not exist. This implies that the variance of the estimators $\widehat{d}_2$ and $\check{d}_2$ might be infinite, even when the true d_2 is finite.

To overcome this problem, we can sacrifice unbiasedness and observe that, under certain policy models (like Gaussians) we can exactly compute the Rényi divergence at single trajectory steps, with a possible benefit in terms of injected uncertainty. Therefore, we

[10] Nevertheless, in these cases, 2 can be a crude underestimation of the true value of the divergence.

Estimator		Min Value	Max Value	Bias	Variance
$\widehat{d}_2$	Equation (30)	0	$\mathfrak{D}_\infty$	0	$\mathcal{O}\left(\frac{1}{N}\mathfrak{D}_4^H\right)$
$\check{d}_2$	Equation (31)	1	$\mathfrak{D}_\infty$	0	$\mathcal{O}\left(\frac{1}{N}\mathfrak{D}_4^H\right)$
$\widetilde{d}_2$	Equation (32)	1	$\mathfrak{D}_2$	$\mathcal{O}\left(\mathfrak{D}_2^H\right)$	$\mathcal{O}\left(\frac{1}{N}\mathfrak{D}_2^{2H}\right)$

Table 2: Comparison of the three estimators for the exponentiated 2–Rényi divergence in terms of minimum value, maximum value, bias, and variance of the estimators. For brevity, we denote with $\mathfrak{D}_\alpha = \sup_{s \in \mathcal{S}} \left\{ d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s) \right) \right\}^{\alpha-1}$ for $\alpha \in [0, +\infty]$. Recall that, from Jensen inequality, $\mathfrak{D}_2^2 \leqslant \mathfrak{D}_4$.

propose the following estimator that computes the product of the (exact) Rényi divergences for the trajectory steps:

$$\widetilde{d}_2 \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right) = \frac{1}{N} \sum_{i=1}^{N} \prod_{t=0}^{H-1} d_2 \left(\pi_{\boldsymbol{\theta}'}(\cdot|s_{\tau_i,t}) \| \pi_{\boldsymbol{\theta}}(\cdot|s_{\tau_i,t}) \right). \tag{32}$$

The main advantage of this estimator is that its variance is proportional to the second moment of the importance weight distribution, i.e., $\mathcal{O}\left(\frac{1}{N}\mathfrak{D}_2^{2H}\right)$. This comes at the price of a bias term that is proportional to the same quantity as well. Nevertheless, the biased estimator $\widetilde{d}_2$ should be preferred over $\widehat{d}_2$ and $\check{d}_2$ as long as $\left(1 + \frac{1}{N}\right)\mathfrak{D}_2^{2H} \lesssim \frac{1}{N}\mathfrak{D}_4^H$. Table 2 compares the properties of the three estimators; the complete analysis of these estimators can be found in Appendix E.

5.3. per–Decision action–based POIS

In the previous section, we introduced A-POIS by defining a unique importance weight $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau)$ for a whole trajectory τ. However, we can refine the estimator $\widehat{J}_{\mathcal{M}}^{\text{A}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ by observing that, given a time step $t \in \{0, 1, ..., H-1\}$, the corresponding reward $R(s_{\tau,t}, a_{\tau,t})$ does not depend on actions and states visited after t. Thus, to reweigh the reward at time t, we can limit the importance weight to consider the products of policy ratios *up to t*. This is the rationale behind the introduction of *Per–Decision Importance Sampling* (PDIS, Precup et al., 2000). Let us define the probability density functions of the trajectory prefixes up to step $t \in \{0, 1, ..., H-1\}$ as:

$$p(\tau|\boldsymbol{\theta}, t) = D(s_{\tau,0}) \prod_{t'=0}^{t} \pi_{\boldsymbol{\theta}}(a_{\tau,t'}|s_{\tau,t'}) P(s_{\tau,t'+1}|s_{\tau,t'}, a_{\tau,t'}). \tag{33}$$

Now, we introduce the per–decision importance weight, defined for each time step as:

$$w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t) = \frac{p(\tau|\boldsymbol{\theta}', t)}{p(\tau|\boldsymbol{\theta}, t)} = \prod_{t'=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t'}|s_{\tau,t'})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t'}|s_{\tau,t'})}, \quad t \in \{0, 1, ..., H-1\}. \tag{34}$$

By using the weights defined in Equation (34), we can provide the following estimator for the expected return:

$$\widehat{J}_{\mathcal{M}}^{\text{D}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{H-1} \gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) R(s_{\tau_i,t}, a_{\tau_i,t}). \tag{35}$$

PDIS preserves the unbiasedness of the estimator, indeed $\mathbb{E}_{\tau \sim p(\cdot|\boldsymbol{\theta})}[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t) R(s_{\tau,t}, a_{\tau,t})] = \mathbb{E}_{\tau \sim p(\cdot|\boldsymbol{\theta})}[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) R(s_{\tau,t}, a_{\tau,t})]$ for all $t \in \{0, 1, ..., H-1\}$. It is worth noting that the importance weight employed in A-POIS is obtained by setting $t = H-1$ in Equation (34), i.e., $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) = w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, H-1)$. Intuitively, by considering a product made up of fewer policy ratios, we might gain an advantage in terms of injected uncertainty. We now provide a bound for the variance of the estimator $\widehat{J}_{\mathcal{M}}^{\text{D}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})$.

Theorem 5 *Let $\widehat{J}_{\mathcal{M}}^{\text{D}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ be the PDIS estimator of the expected return $J(\boldsymbol{\theta}')$ computed with N i.i.d. trajectories $\boldsymbol{\tau} = (\tau_1, \tau_2, \dots, \tau_N)$ collected running $\pi_{\boldsymbol{\theta}}$, as defined in Equation (35). If $p(\tau|\boldsymbol{\theta}', t) \ll p(\tau|\boldsymbol{\theta}, t)$ for all $t \in \{0, 1, ..., H-1\}$, then, the variance of $\widehat{J}_{\mathcal{M}}^{\text{D}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ can be upper bounded as:*

$$\underset{\boldsymbol{\tau} \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\text{ar}} \left[\widehat{J}_{\mathcal{M}}^{\text{D}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) \right] \leqslant \frac{R_{\max}^2}{N} \sum_{t=0}^{H-1} c_t d_2 \left(p(\cdot|\boldsymbol{\theta}', t) \| p(\cdot|\boldsymbol{\theta}, t) \right), \tag{36}$$

where c_t is defined as:

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

Proof

$$\underset{\boldsymbol{\tau} \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\text{ar}} \left[\widehat{J}_{\mathcal{M}}^{\text{D}-\text{POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) \right] = \frac{1}{N} \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\text{ar}} \left[\sum_{t=0}^{H-1} \gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t) R(s_{\tau_1,t}, a_{\tau_1,t}) \right] \tag{P.10}$$

$$\leqslant \frac{1}{N} \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}} \left[\left(\sum_{t=0}^{H-1} \gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t) R(s_{\tau_1,t}, a_{\tau_1,t}) \right)^2 \right] \tag{P.11}$$

$$\leqslant \frac{R_{\max}^2}{N} \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}} \left[\left(\sum_{t=0}^{H-1} \gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t) \right)^2 \right] \tag{P.12}$$

$$= \frac{R_{\max}^2}{N} \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}} \left[\sum_{t=0}^{H-1} \gamma^{2t} w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t)^2 + 2 \sum_{t=0}^{H-2} \sum_{t'=t+1}^{H-1} \gamma^{t+t'} w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t) w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t') \right]$$

$$= \frac{R_{\max}^2}{N} \left\{ \sum_{t=0}^{H-1} \gamma^{2t} \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t)^2 \right] + 2 \sum_{t=0}^{H-2} \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t)^2 \right] \sum_{t'=t+1}^{H-1} \gamma^{t+t'} \right\} \tag{P.13}$$

$$= \frac{R_{\max}^2}{N} \sum_{t=0}^{H-1} \left(\gamma^{2t} + \frac{2\gamma^t (\gamma^{t+1} - \gamma^H)}{1-\gamma} \right) \underset{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t)^2 \right] \tag{P.14}$$

$$= \frac{R_{\max}^2}{N} \sum_{t=0}^{H-1} \frac{\gamma^t \left(\gamma^t + \gamma^{t+1} - 2\gamma^H\right)}{1-\gamma} d_2\left(p(\cdot|\boldsymbol{\theta}', t) \| p(\cdot|\boldsymbol{\theta}, t)\right), \tag{P.15}$$

where line (P.10) follows from the fact that the trajectories τ_i are i.i.d., line (P.11) is obtained by bounding the variance with the second moment, line (P.12) derives from the fact that the immediate reward is uniformly bounded, line (P.13) is obtained by observing that $\mathbb{E}_{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t) w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t')\right] = \mathbb{E}_{\tau_1 \sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_1, t)^2\right]$ as $t' > t$, line (P.14) follows from the properties of the geometric sum and finally line (P.15) is obtained from the definition of d_2 and $p(\cdot|\boldsymbol{\theta}, t)$. By taking the limit we get the expression for $\gamma = 1$:

$$\lim_{\gamma \to 1} \frac{\gamma^t \left(\gamma^t + \gamma^{t+1} - 2\gamma^H\right)}{1-\gamma} = 2H - 2t - 1. \tag{P.16}$$

■

Using the estimator defined in Equation (35) and the bound on the variance given in Theorem 5, we can define the objective function for the new *per–Decision action–based POIS* (D-POIS):

$$\begin{aligned} \mathcal{L}_{\lambda}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = & \underbrace{\frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{H-1} \gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i, t) R(s_{\tau_i,t}, a_{\tau_i,t})}_{\hat{J}_{\mathcal{M}}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})} \\ & - \lambda \sqrt{\frac{1}{N} \sum_{t=0}^{H-1} c_t d_2\left(p(\cdot|\boldsymbol{\theta}', t) \| p(\cdot|\boldsymbol{\theta}, t)\right)}, \end{aligned} \tag{37}$$

where $\lambda = R_{\max}\sqrt{\frac{1-\delta}{\delta}}$ is the regularization parameter and c_t is defined in Theorem 5. As in the action–based setting, we make use of an estimator for the exponentiated Rényi divergence, as a direct computation of $d_2\left(p(\cdot|\boldsymbol{\theta}', t) \| p(\cdot|\boldsymbol{\theta}, t)\right)$ is also not possible in this setting as it requires to compute an integral over the space of trajectories:

$$d_2\left(p(\cdot|\boldsymbol{\theta}', t) \| p(\cdot|\boldsymbol{\theta}, t)\right) = \int_{\mathcal{T}} p(\tau|\boldsymbol{\theta}, t) \left(\frac{p(\tau|\boldsymbol{\theta}', t)}{p(\tau|\boldsymbol{\theta}, t)}\right)^2 \mathrm{d}\tau = \int_{\mathcal{T}} p(\tau|\boldsymbol{\theta}, t) \left(\prod_{t'=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t'}|s_{\tau,t'})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t'}|s_{\tau,t'})}\right)^2 \mathrm{d}\tau.$$

Since we need to estimate this term for each timestep t, we can use the estimators presented in Remark 6 simply by limiting the product to time t instead of $H-1$.

Similarly to A-POIS, we can derive a multiple importance sampling extension based on the balance heuristic for each time step $t \in \{0, \ldots, H-1\}$. Let $\boldsymbol{\theta}_{1:J} = \{\boldsymbol{\theta}_j\}_{j=1}^J$ be the set of behavioral policy parameters, we have:

$$w_{\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J}}^{\mathrm{BH}}(\tau, t) = \frac{p(\tau|\boldsymbol{\theta}', t)}{\sum_{k=1}^J \frac{N_k}{N} p(\tau|\boldsymbol{\theta}_j, t)} = \frac{\prod_{t'=0}^t \pi_{\boldsymbol{\theta}'}(a_{\tau,t'}|s_{\tau,t'})}{\sum_{k=1}^J \prod_{t'=0}^t \frac{N_k}{N} \pi_{\boldsymbol{\theta}_j}(a_{\tau,t'}|s_{\tau,t'})}.$$

Consequently, by applying Corollary 1 we obtain the objective function:

$$\mathcal{L}_{\lambda}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J}) = \underbrace{\frac{1}{N}\sum_{j=1}^{J}\sum_{i=1}^{N}\sum_{t=0}^{H-1}\gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J}}^{\mathrm{BH}}(\tau_{ij},t)R(s_{\tau_{ij},t},a_{\tau_{ij},t})}_{\widehat{J}_{\mathcal{M}}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}_{1:J})} \\ - \lambda\sqrt{\frac{1}{N}\sum_{t=0}^{H-1}\frac{c_t}{\sum_{j=1}^{J}\frac{N_j}{d_2\left(p(\cdot|\boldsymbol{\theta}',t)\|p(\cdot|\boldsymbol{\theta}_j,t)\right)}}}, \tag{38}$$

where each τ_{ij} is obtained by running policy $\pi_{\boldsymbol{\theta}_j}$ in the environment with $i = 1, 2, ..., N_j$ and $j = 1, 2, ..., J$. The learning process is analogous to that of A-POIS with the only foresight to employ objective function at Equation (38).

Remark 7 (Analysis of the Variance of D-POIS) We now discuss in more detail the intuition behind the possible uncertainty reduction granted by the PDIS. We will prove that the reduction in variance actually holds for the importance weights (Proposition 2) but not, in general, for the expected return estimator (Fact 1).

Proposition 2 *Let $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau)$ be the importance weight of trajectory τ and $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau,t)$ be the per–decision importance weight of trajectory τ up to time $t \in \{0,1,...,H-1\}$. Then, it holds that:*

$$\mathop{\mathbb{V}\mathrm{ar}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau,t)\right] \leqslant \mathop{\mathbb{V}\mathrm{ar}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau)\right].$$

While we are able to prove the lower variance of the PDIS weights compared to IS, we are unable to do the same for the variance of the expected return estimator itself. We provide in the following a counter–example where the variance of the PDIS estimator $\widehat{J}_{\mathcal{M}}^{\mathrm{D-POIS}}$ is greater than the variance of the vanilla IS estimator $\widehat{J}_{\mathcal{M}}^{\mathrm{A-POIS}}$:

Fact 1 *Let $\widehat{J}_{\mathcal{M}}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ and $\widehat{J}_{\mathcal{M}}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})$ the estimators of the expected return using IS and PDIS respectively. Then, there exists an MDP $\mathcal{M}$ and a pair of behavioral and target policies $(\pi_{\boldsymbol{\theta}},\pi_{\boldsymbol{\theta}'})$ such that:*

$$\mathop{\mathbb{V}\mathrm{ar}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] > \mathop{\mathbb{V}\mathrm{ar}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right].$$

Proof Consider an MDP $\mathcal{M}$ with three states $\mathcal{S} = \{s_1, s_2, s_3\}$, where s_3 is an absorbing state. In state s_1, only one action is available, a, which transitions deterministically to s_2 and provides a reward of $R(s_1, a) = 1$. In state s_2, there are two actions available, a_1 and a_2, both of which transition deterministically to s_3 (thus ending the episode). The rewards are $R(s_2, a_1) = 0$ and $R(s_2, a_2) = -1$. The episode starts in state s_1, and the discount factor is $\gamma = 1$. The behavioral policy is uniform over the actions, i.e., $\pi(a_1|s_1) = \pi(a_2|s_1) = 1/2$. The target policy assigns probability $q \in [0, 1]$ to action a_1, i.e., $\pi'(a_1|s_1) = q$ and $\pi'(a_2|s_1) = 1-q$ (Figure 3). We are going to find a proper value of q such that the claim holds. Without loss of generality, we consider trajectories of length 2. There are two possible trajectories in this environment, $\tau_1 = (s_1, a, s_2, a_1, s_3)$ and $\tau_2 = (s_1, a, s_2, a_2, s_3)$, which differ only in the action

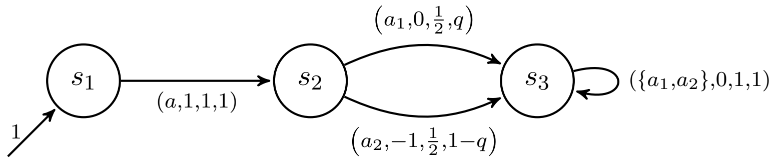

s_1
s_2
s_3
1
$(a,1,1,1)$
$\left(a_1,0,\frac{1}{2},q\right)$
$\left(a_2,-1,\frac{1}{2},1-q\right)$
$(\{a_1,a_2\},0,1,1)$

Figure 3: The MDP considered in the proof. Each arrow connecting two states s and s' is labeled with a 4–tuple $(a,\ R(s,a),\ P(s'|s,a),\ \pi(a|s))$.

taken in state s_2, and they both have probability 1/2 under the behavioral policy. Since both estimators are unbiased, we compare the second moments:

$$\mathop{\mathbb{E}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[\left(\widehat{J}_{\mathcal{M}}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right)^2\right] = p(\tau_1)\left(\frac{\pi'(a|s_1)\pi'(a_1|s_2)}{\pi(a|s_1)\pi(a_1|s_2)}R(\tau_1)\right)^2 + p(\tau_2)\left(\frac{\pi'(a|s_1)\pi'(a_2|s_2)}{\pi(a|s_1)\pi(a_2|s_2)}R(\tau_2)\right)^2 = 2q^2.$$

$$\mathop{\mathbb{E}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[\left(\widehat{J}_{\mathcal{M}}^{\mathrm{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right)^2\right] = p(\tau_1)\left(\frac{\pi'(a|s_1)}{\pi(a|s_1)}R(s_1,a) + \frac{\pi'(a|s_1)\pi'(a_1|s_2)}{\pi(a|s_1)\pi(a_1|s_2)}R(s_2,a_1)\right)^2 + p(\tau_2)\left(\frac{\pi'(a|s_1)}{\pi(a|s_1)}R(s_1,a) + \frac{\pi'(a|s_1)\pi'(a_2|s_2)}{\pi(a|s_1)\pi(a_2|s_2)}R(s_2,a_2)\right)^2 = 2q^2-2q+1.$$

Finally, we find a value of q to satisfy the claim:

$$2q^2 < 2q^2 - 2q + 1 \implies q < \frac{1}{2}.$$

■

This fact is explained by considering that we are not only lowering the variance of the weights in the per–decision setting, but we are also considering the immediate reward instead of the episode return. The latter can be highly correlated with the importance weight, leading to a larger variance. Therefore, the actual benefit of PDIS over IS is, in general, task–dependent. Similar analyses have recently been proposed highlighting these features of the PDIS estimator (Rowland et al., 2020; Liu et al., 2019a).

Remark 8 (Asymptotic Analysis for the Variance of A-POIS and D-POIS) We now discuss how the penalty term in the objective function optimized by A-POIS and D-POIS changes as the horizon H of the task and the discount factor γ change. To simplify the analysis, we will resort to the upper bound on the exponentiated Rényi divergence $d_2\left(p(\cdot|\boldsymbol{\theta}',t)\|p(\cdot|\boldsymbol{\theta},t)\right) \leqslant \mathfrak{D}_2^t$. We are interested in analyzing the growth rate of the variance term as a function of the horizon H of the task. Table 3 reports the asymptotic approximation of the variance of A-POIS and D-POIS estimators when $H \to +\infty$ for (a) $\gamma < 1$ and (b) $\gamma = 1$. In the discounted case ($\gamma < 1$), we notice that the variance of A-POIS is finite, independent

	A-POIS	D-POIS
$\mathfrak{D}_2 = 1$	$\frac{1}{(1-\gamma)^2}$	
$1 < \mathfrak{D}_2 < \gamma^{-2}$	$\frac{\mathfrak{D}_2^H}{(1-\gamma)^2}$	$\frac{1+\gamma}{(1-\gamma)(1-\mathfrak{D}_2\gamma^2)}$
$\mathfrak{D}_2 = \gamma^{-2}$		$\frac{1+\gamma}{1-\gamma}H$
$\mathfrak{D}_2 > \gamma^{-2}$		$\frac{(\gamma^2\mathfrak{D}_2)^H(\mathfrak{D}_2\gamma+1)}{(\mathfrak{D}_2\gamma^2-1)(\mathfrak{D}_2\gamma-1)}$

(a) $\gamma < 1$

	A-POIS	D-POIS
$\mathfrak{D}_2 = 1$	H^2	
$\mathfrak{D}_2 > 1$	$H^2\mathfrak{D}_2^H$	$\mathfrak{D}_2^H \frac{\mathfrak{D}_2+1}{(\mathfrak{D}_2-1)^2}$

(b) $\gamma = 1$

Table 3: Asymptotic growth rate of the variance upper bound for A-POIS and D-POIS. We omitted the factor $\frac{R_{\max}}{N}$, which is common to all cases, for clarity.

from H only when $\mathfrak{D}_2 = 1$, i.e., in the on–policy setting, while it grows exponentially in H for $\mathfrak{D}_2 > 1$. Instead, the variance of D-POIS is finite as long as $\mathfrak{D}_2 < \frac{1}{\gamma^2}$. Intuitively, in the per–decision weighting scheme, the importance weights are discounted by γ and this allows keeping the variance finite even in the off–policy setting, provided that $\mathfrak{D}_2$ is small enough. When $\mathfrak{D}_2 = \frac{1}{\gamma^2}$ we have a linear growth rate in H, which becomes exponential as $\mathfrak{D}_2 > \frac{1}{\gamma^2}$. In the undiscounted setting ($\gamma = 1$), we do not experience a significant advantage of the per–decision weights, as the discounting effect on the importance weights disappears. Indeed, for both A-POIS and D-POIS the variance becomes exponential in H when $\mathfrak{D}_2 > 1$. The complete analysis is available in Appendix F.

Remark 9 (Risk–Averse vs Risk–Seeking Objectives) The perspective that we have adopted in this paper is to employ off–distribution techniques in order to estimate the performance of target distributions and, consequently, being able to perform multiple gradient steps using the same data, possibly collected with multiple behavioral distributions. Specifically, the objective functions we optimize are *risk–averse*, penalizing distributions that are far from the behavioral ones. This is justified by the fact that, as we move away from the behavioral distribution, we likely experience larger uncertainty. As a consequence, our approach can be defined as "pessimistic" and might lead to an over–conservative behavior, preventing the exploration of certain regions of the parameter space. Nevertheless, we can prove that, under the same learning rate schedule, our off–distribution optimization "moves" in the parameter space at least as the (on–policy) policy gradient methods. Limiting the reasoning for simplicity to the action–based setting[11] and assuming a single behavioral distribution (i.e., $J = 1$) we have that when $\boldsymbol{\theta}' = \boldsymbol{\theta}$ (i.e., in the first step of the off–policy optimization), we take a step identical to the standard policy gradient. Specifically, for A-POIS we take a step equivalent to REINFORCE (Williams, 1992):

$$\nabla_{\boldsymbol{\theta}'}\mathcal{L}_{\lambda}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'=\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N}\nabla_{\boldsymbol{\theta}}\log p(\tau_i|\boldsymbol{\theta})R(\tau_i) = \widehat{\nabla}_{\boldsymbol{\theta}}^{\mathrm{REINFORCE}} J_{\mathcal{M}}(\boldsymbol{\theta}),$$

[11] The same rationale holds for the parameter–based setting.

while for the D-POIS case, we reduce to G(PO)MDP (Baxter and Bartlett, 2001):

$$\nabla_{\boldsymbol{\theta}'} \mathcal{L}_{\lambda}^{\mathrm{P-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}'=\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{H-1} \gamma^t \nabla_{\boldsymbol{\theta}} \log p(\tau_i|\boldsymbol{\theta}, t) R(s_{\tau_i,t}, a_{\tau_i,t}) = \widehat{\nabla}_{\boldsymbol{\theta}}^{\mathrm{G(PO)MDP}} J_{\mathcal{M}}(\boldsymbol{\theta}).$$

This effect is justified by the fact that the gradient of the Rényi divergence, and consequently the gradient of the penalization, is zero when $\boldsymbol{\theta}' = \boldsymbol{\theta}$ (see the proof of Theorem 4).

A perspective more focused on exploration could employ a *risk–seeking* objective, in which distributions that are far from the behavioral ones are rewarded. In principle, obtaining such an "optimistic" objective just amounts to switch the sign of the penalization and make it a bonus. This idea is at the basis of OPTIMIST (Papini et al., 2019). However, the naïve switch of sign is typically unsatisfactory both theoretically and empirically. If we allow the weights to get any value, they likely degenerate towards an infinite Rényi divergence. For this reason, in OPTIMIST weight truncation (coming with additional challenges in the optimization phase) is employed to limit the values of the importance weights.

To make the most of the data, we should consider the long–term advantages of exploration while retaining a conservative approach. This problem has been addressed with a meta–gradient technique by Papini et al. (2020) in the narrower scope of on–policy policy gradient with Gaussian policies. The proposed algorithms are very conservative since they are motivated by safety constraints. The development of conservative exploration strategies more oriented towards sample efficiency is an interesting research direction.

6. Related Works

Policy optimization algorithms can be classified according to different dimensions (see also Table 8). Online PG methods are likely the most popular policy search approaches: starting from the traditional algorithms based on stochastic policy gradient (Sutton et al., 2000), like REINFORCE (Williams, 1992) and G(PO)MDP (Baxter and Bartlett, 2001), moving toward more modern methods, such as Trust Region Policy Optimization (TRPO, Schulman et al., 2015a) and its extensions (e.g., Schulman et al., 2017). It is by now established, in the policy–based RL community, that effective algorithms, either on–policy or off–policy, should account for the variance of the gradient estimate. Early attempts, in the class of action–based algorithms, are the usage of a baseline to reduce the estimated gradient variance without introducing bias (Baxter and Bartlett, 2001; Peters and Schaal, 2008b). A similar rationale underlies actor–critic architectures (Konda and Tsitsiklis, 2000; Sutton et al., 2000; Peters and Schaal, 2008a), in which an estimate of the value function is used to reduce uncertainty. Baselines are typically constant (REINFORCE), time–dependent (G(PO)MDP), or state–dependent (actor–critic), but these approaches have recently been extended to take action–dependent baselines (Tucker et al., 2018; Wu et al., 2018) into account. Another line of work tries to reduce the gradient–estimation variance by exploiting the correlation between consecutive estimates (Papini et al., 2018; Xu et al., 2019a; Shen et al., 2019; Xu et al., 2019b). Off–policy optimization has been also used in conjunction with deterministic policies in Deterministic Policy Gradient (DPG, Silver et al., 2014), where data are collected with a noisy version of the target policy. This allows decoupling exploration from gradient estimation, freeing the latter from an unnecessary source of variance. More recently, an efficient version of DPG coupled with a deep neural network to represent the policy has been proposed, named Deep Deterministic Policy Gradient (DDPG, Lillicrap et al., 2015). Expected Policy Gradients (Ciosek and Whiteson, 2018) apply the same variance–reduction technique to stochastic policies by employing tractable critics. In the parameter–based framework, even though the original formulation (Sehnke et al., 2008) introduces an on–line algorithm, an extension has been proposed to efficiently reuse the trajectories in an off–line scenario (Zhao et al., 2013). Furthermore, PGPE–like approaches allow overcoming several limitations of classical PG, such as the need for a stochastic policy and the high variance of the gradient estimates. Even though parameter–based algorithms are, by their nature, affected by less variance than action–based ones, it is possible to derive baselines similar to those of the action–based case (Zhao et al., 2011).

A first dichotomy in the policy optimization landscape comes when considering the minimal unit used to compute the gradient. *Episode–based* (or episodic) approaches (e.g., Williams, 1992; Baxter and Bartlett, 2001) estimate the gradient by averaging the gradients of each episode which, in some cases, needs to have a finite horizon. On the contrary, *step–based* approaches (e.g., Schulman et al., 2015a, 2017; Lillicrap et al., 2015), derived from the Policy Gradient Theorem (Sutton et al., 2000), can estimate the gradient by averaging over timesteps. The latter requires a function approximator (a critic) to estimate the Q–function, or directly the advantage function (Schulman et al., 2015b). When coming to the on/off–policy dichotomy, the previous distinction has a significant impact. Indeed, episode–based approaches need to perform importance sampling on trajectories, thus the importance weights are the products of policy ratios for all executed actions within a trajectory. The longer the horizon, the more the variance injected by the single policy ratios is amplified. Instead, step–based algorithms need just a single density ratio per sample, which helps to keep the value of the importance weights closer to one. On the other hand, these step–based weights must also account for the mismatch between the state–occupancy measures induced by the two policies, i.e., $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(s,a) = \frac{d_D^{\boldsymbol{\theta}'}(s)\pi_{\boldsymbol{\theta}'}(a|s)}{d_D^{\boldsymbol{\theta}}(s)\pi_{\boldsymbol{\theta}}(a|s)}$, where $d_D^{\boldsymbol{\theta}}$ denotes the (discounted) probability of ending up in state s under policy $\pi_{\boldsymbol{\theta}}$.[12] Unfortunately, unlike policy and trajectory–probability ratios, state–occupancy ratios cannot be computed in closed form. Simply ignoring them, as sometimes done in the policy gradient literature (e.g., Degris et al., 2012; Silver et al., 2014), can result in poor performance (Liu et al., 2019b). Recent work provides ways to estimate these state occupancy ratios (Hallak and Mannor, 2017; Liu et al., 2018; Gelada and Bellemare, 2019; Liu et al., 2019b), possibly paving the way for a step–based version of POIS. However, computing the Rényi distance between state–action distributions is another difficult problem. Furthermore, these step–based methods typically employ a critic, which prevents a complete analysis of the uncertainty, as the bias/variance injected by the critic is hard to compute (Konda and Tsitsiklis, 2000).

Preventing uncontrolled updates in the policy parameter space is at the core of natural gradient approaches (Amari, 1998) applied effectively both on PG methods (Kakade and Langford, 2002; Peters and Schaal, 2008a; Wierstra et al., 2008) and on PGPE methods (Miyamae et al., 2010). More recently, this idea has been exploited by TRPO (Schulman et al., 2015a), which optimizes a surrogate objective function via (approximate) natural gradient, derived from safe RL (Kakade and Langford, 2002; Pirotta et al., 2013), subject to

[12] More formally, $d_D^{\boldsymbol{\theta}}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s|s_0 \sim D, a_h \sim \pi_{\boldsymbol{\theta}}(\cdot|s_h), s_{h+1} \sim P(\cdot|s_h, a_h)$ for all $h < t)$.

a constraint on the Kullback–Leibler divergence between the behavioral and target policy.[13] The actual algorithm employs several approximations, some of which can be removed (Pajarinen et al., 2019). According to recent works (Neu et al., 2017; Shani et al., 2020), TRPO should be understood as an approximate version of mirror descent rather than a conservative method. Like TRPO, PPO truncates the importance weights to discourage the optimization process from going too far. Although TRPO and PPO, together with DDPG, represent the state–of–the–art policy optimization methods in RL for continuous control, they do not explicitly encode in their objective function the uncertainty injected by the importance sampling procedure. Indeed, the step size in TRPO and the truncation range ϵ in PPO are just hyperparameters and have a limited statistical meaning. On the contrary, other actor–critic architectures have been proposed including also experience replay methods, like Wang et al. (2016), in which the importance weights are truncated, but the method is able to account for the injected bias. The authors propose to keep a running mean of the best policies seen so far to avoid a hard constraint on the policy dissimilarity. A more theoretically grounded analysis has been provided for policy selection (Doroudi et al., 2017), model–free (Thomas et al., 2015b) and model–based (Thomas and Brunskill, 2016) policy evaluation (also accounting for samples collected with multiple behavioral policies), and combined with options (Guo et al., 2017). Subsequently, in Thomas et al. (2015a), these methods have been extended for policy improvement, deriving a suitable concentration inequality for the case of truncated importance weights. Unfortunately, these methods are hardly scalable to complex control tasks. The recent Policy–on Policy–off Policy Optimization (P3O, Fakoor et al., 2019) algorithm is another way to interleave on–policy and off–policy gradient updates via importance sampling. Unlike POIS, the two kinds of gradients are combined into a single update. In addition, a KL penalty is used to control the deviation of the target policy from the behavioral, while a 2–Rényi divergence would be more appropriate.

Unlike these methods, POIS directly models the uncertainty due to the importance sampling procedure. The bound in Theorem 2 introduces the unique hyperparameter δ, which has a precise statistical meaning as a confidence level. The optimal value of δ (like the step size in TRPO and ϵ in PPO) is task–dependent and may vary during the learning procedure. Furthermore, the use of PDIS, apart from the variance reduction benefit, allows performing importance weighting on trajectory prefixes and, therefore, assigning partial credit to valuable subtrajectories.

7. Experimental Evaluation

In this section, we present the experimental evaluation of POIS in its different flavors (parameter–based, action–based, action–based per–decision). We first provide a set of empirical comparisons on classical continuous control tasks with linearly parametrized policies (Section 7.1); we then show how POIS can be adopted for learning deep neural policies (Section 7.2). We also study the effects of employing per–decision importance weights and multiple importance weights (Section 7.3). In all experiments, for A-POIS and

[13] Note that this regularization term appears in the performance improvement bound, which contains exact quantities only. Thus, it does not actually account for the uncertainty derived from the importance sampling.

D-POIS we used the IS estimator, while for P-POIS we employed the SN estimator. All experimental details are provided in Appendix H.[14]

7.1. Linear Policies

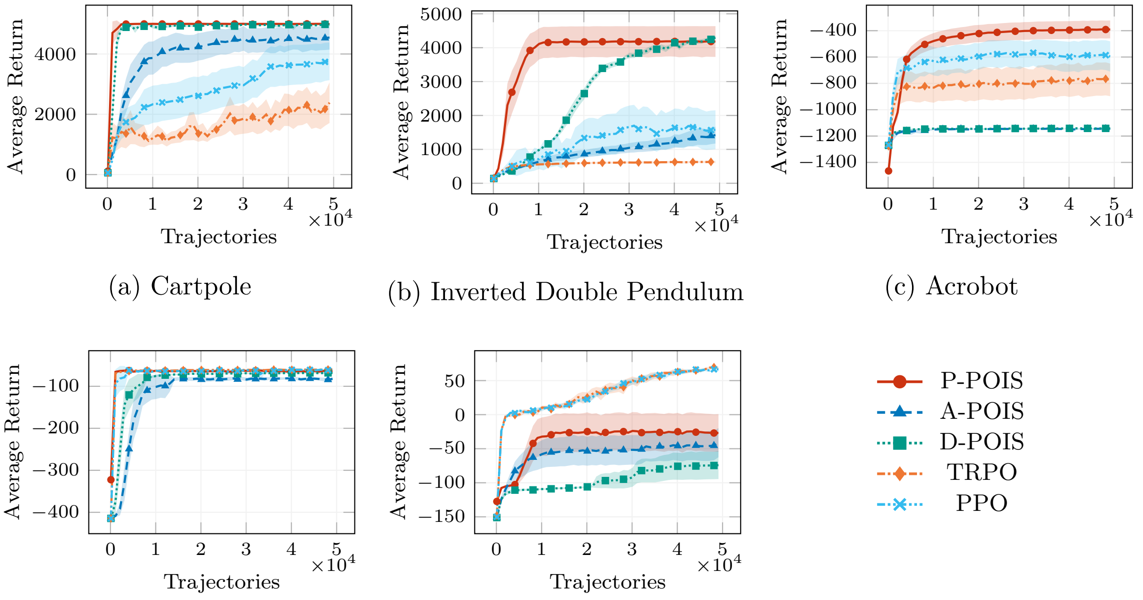
Linearly parametrized Gaussian policies proved their ability to scale to complex control tasks (Rajeswaran et al., 2017). In this section, we compare the learning performance of A-POIS, D-POIS, and P-POIS against TRPO (Schulman et al., 2015a) and PPO (Schulman et al., 2017) on some classical continuous control benchmarks (Duan et al., 2016). The hyperparameters of the individual algorithms are reported in Table 4. In the Cartpole environment, as we can see from Figure 4, all the POIS variants outperform significantly the performance of TRPO and PPO, showing not only the convergence to the optimum but also a faster convergence speed. This is particularly true of P-POIS and D-POIS, where the optimum is reached in very few iterations. Indeed, in this case, we can appreciate the benefit of the PDIS technique over the simple IS. For the Inverted Double Pendulum environment, we have a less consistent behavior: A-POIS has similar performance w.r.t. TRPO and PPO, while P-POIS learns the optimal policy at a remarkable pace; D-POIS stays somewhere in the middle, still reaching the optimum but at a slower rate. In the acrobot task, we see how, once again, P-POIS outperforms both TRPO and PPO, while A-POIS and D-POIS get stuck in what could be a local optimum. The mountain–car environment shows a very similar behavior among all the benchmarked algorithms, with A-POIS and D-POIS having a slightly slower convergence speed. Lastly, the inverted–pendulum setting is the only environment in which no version of POIS can keep up with the TRPO and PPO baselines. Overall, POIS displays a performance comparable with TRPO and PPO across the tasks. In particular, P-POIS displays better performance w.r.t. A-POIS. Furthermore, apart from the peculiar case of the inverted pendulum, D-POIS performs at least as good as A-POIS, empirically supporting the intuition that the PDIS technique helps to reduce the variance of the performance estimation and, thus, allows faster learning.

In Figure 5 we show, for several metrics, the behavior of A-POIS when changing the δ parameter in the Cartpole environment. We can see that when δ is small (e.g., 0.2), the Effective Sample Size (ESS) remains large and, consequently, the variance of the importance weights ($\mathbb{V}\mathrm{ar}[w]$) is small. This means that the penalty term in the objective function discourages the optimization process from selecting policies that are far from the behavioral policy. As a consequence, the displayed behavior is very conservative, preventing the policy from reaching the optimum. On the contrary, when δ approaches 1, the ESS is smaller and the variance of the weights tends to increase significantly. Again, the performance remains suboptimal as the penalty term in the objective function is too light. The best behavior is obtained with an intermediate value of δ, specifically 0.4.

[14] For all experiments, we plot the *confidence intervals* among the performances of the individual runs. Albeit common in the RL literature (Henderson et al., 2018), this kind of visualization does not fully capture the inherent variability of the algorithm's performance over different random seeds. In Appendix H.6, we report *tolerance intervals* for the experiments with linear policies and MIS, while in Appendix H.7, we plot the individual runs for the experiments with deep neural policies.

Task	P-POIS (δ)	A-POIS (δ)	D-POIS (δ)	TRPO (step size)	PPO (step size)
Cartpole	0.4	0.4	0.99	0.1	0.01
Inverted Double Pendulum	0.1	0.1	0.4	0.1	1
Acrobot	0.2	0.7	0.7	1	1
Mountain Car	1	0.9	0.9	0.01	1
Inverted Pendulum	0.8	0.9	0.9999	0.01	0.01

Table 4: Hyperparameter value of the individual algorithms employed in the experiments shown in Figure 4. For all versions of POIS we report the value of δ, while for TRPO (Schulman et al., 2015a) and PPO (Schulman et al., 2017) the value of the step size.



(a) Cartpole

(b) Inverted Double Pendulum

(c) Acrobot

(d) Mountain Car

(e) Inverted Pendulum

Figure 4: Average return as a function of the number of trajectories for P-POIS, A-POIS, D-POIS, TRPO, and PPO with *linear policy* (20 runs, 95% c.i.).

7.2. Deep Neural Policies

In this section, we adopt a deep neural network (3 layers: 100, 50, 25 neurons each) to represent the policy. The experiment setup is fully compatible with the classical benchmark (Duan et al., 2016). The value of the hyperparameters is reported in Table 5. While A-POIS and D-POIS can be directly applied to deep neural networks, P-POIS exhibits some critical issues. A high–dimensional hyperpolicy (like a Gaussian from which the weights of an MLP policy are sampled) can make $d_2(\nu_{\boldsymbol{\rho}'} \| \nu_{\boldsymbol{\rho}})$ extremely sensitive to small parameter

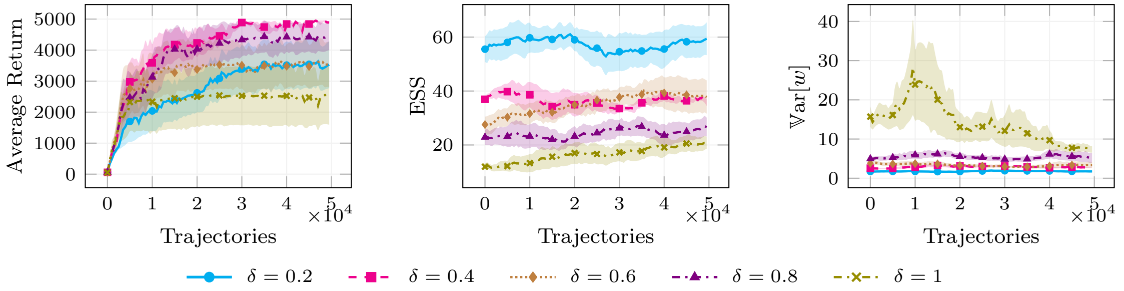


Figure 5: Average return, Effective Sample Size (ESS), and variance of the importance weights ($\mathbb{V}\mathrm{ar}[w]$) as a function of the number of trajectories for A-POIS for different values of the parameter δ in the Cartpole environment (20 runs, 95% c.i.).

Task	P-POIS (δ)	A-POIS (δ)	D-POIS (δ)
Inverted Double Pendulum	0.8	0.99	0.4
Cartpole	0.6	0.99	0.99
Mountain Car	0.3	0.99	0.99
Swimmer	0.6	0.99	0.99

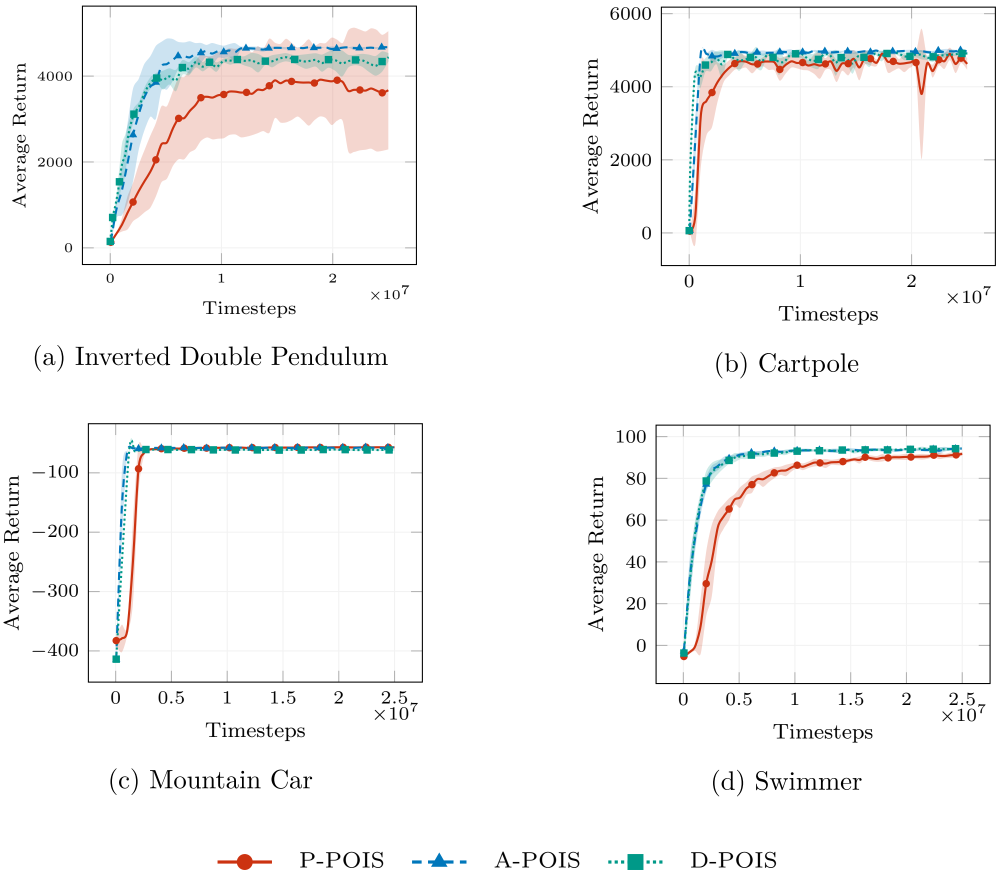
Table 5: Hyperparameter value of the individual algorithms employed in the experiments shown in Figure 6. For all versions of POIS we report the value of δ.

changes, which leads to over–conservative updates.[15] A first practical variant comes from the insight that $d_2(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}})/N$ is the inverse of the effective sample size, as reported in Equation 7. We can obtain a less conservative (although approximate) surrogate function by replacing it with $1/\widehat{\mathrm{ESS}}(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}})$. Another trick is to model the hyperpolicy as a set of independent Gaussians, each defined over a disjoint subspace of Θ (implementation details are provided in Appendix G.3). In Table 6, we augmented the results provided in (Duan et al., 2016); in Figure 6, we also provide performance plots during training, as we did in the previous section. We can see that A-POIS and D-POIS are able to achieve an overall behavior similar to the best of the action–based algorithms, approaching TRPO and beating DDPG. Similarly, P-POIS exhibits performance similar to CEM (Szita and Lörincz, 2006), the best performing among the parameter–based methods.

7.3. Multiple P-POIS

We also present some results related to the multiple importance sampling extension we introduced in Section 3.2. While this extension can be applied to every flavor of POIS we have introduced before, we only focus on the P-POIS setting with a linear policy, in order to

[15] This curse of dimensionality, related to $\dim(\boldsymbol{\theta})$, has some similarities with the dependence of the Rényi divergence on the actual horizon H in the action–based case.



(a) Inverted Double Pendulum

(b) Cartpole

(c) Mountain Car

(d) Swimmer

Figure 6: Average return as a function of the number of trajectories for A-POIS, P-POIS with deep neural policies (5 runs, 95% c.i.).

briefly present the pros and cons of this particular strategy. The hyperparameter values are reported in Table 7. To highlight the benefits of multiple importance weights on the effective sample size, we set the batch size to one. In the original P-POIS, this means that our agent collects a single trajectory per (on–line) iteration. With MIS, in principle, we could employ all the previous trajectories at each iteration, provided that we store all the past behavioral hyper–policies, together with their sampled policy parameters and the resulting returns. For computational reasons (see Table 1), we employ a finite memory, managed as a simple FIFO queue. The *capacity* of this queue is the number of most recent behavioral hyper–policies it can store (corresponding to J in Equation 10). We use weight normalization whenever possible. For MIS, we employ the self–normalized weights from Equation (12). In Figure 7 we compare, on the usual benchmark tasks, P-POIS with single importance sampling (i.e., $J = 1$) to its MIS counterpart for different values of the capacity. We also report single–IS P-POIS with a batch size of 10. The latter allows appreciating the difference between having 10 “fresh” samples from the current behavioral hyper–policy and re–using old ones with MIS instead.

Algorithm	Cart-Pole Balancing	Mountain Car	Double Inverted Pendulum	Swimmer
Random	77.1 ± 0.0	−415.4 ± 0.0	149.7 ± 0.1	−1.7 ± 0.1
REINFORCE	4693.7 ± 14.0	−67.1 ± 1.0	4116.5 ± 65.2	92.3 ± 0.1
TNPG	**3986.4 ± 748.9**	**−66.5 ± 4.5**	**4455.4 ± 37.6**	**96.0 ± 0.2**
RWR	**4861.5 ± 12.3**	−79.4 ± 1.1	3614.8 ± 368.1	60.7 ± 5.5
REPS	565.6 ± 137.6	−275.6 ± 166.3	446.7 ± 114.8	3.8 ± 3.3
TRPO	**4869.8 ± 37.6**	**−61.7 ± 0.9**	**4412.4 ± 50.4**	**96.0 ± 0.2**
DDPG	4634.4 ± 87.6	−288.4 ± 170.3	2863.4 ± 154.0	85.8 ± 1.8
A-POIS	**4842.8 ± 13.0**	−63.7 ± 0.5	**4232.1 ± 189.5**	88.7 ± 0.55
D-POIS	**4819.3 ± 59.3**	**−61.0 ± 0.5**	**4333.8 ± 115.4**	88.2 ± 1.49
CEM	4815.4 ± 4.8	−66.0 ± 2.4	2566.2 ± 178.9	68.8 ± 2.4
CMA-ES	2440.4 ± 568.3	−85.0 ± 7.7	1576.1 ± 51.3	64.9 ± 1.4
P-POIS	4428.1 ± 138.6	−78.9 ± 2.5	3161.4 ± 959.2	76.8 ± 1.6

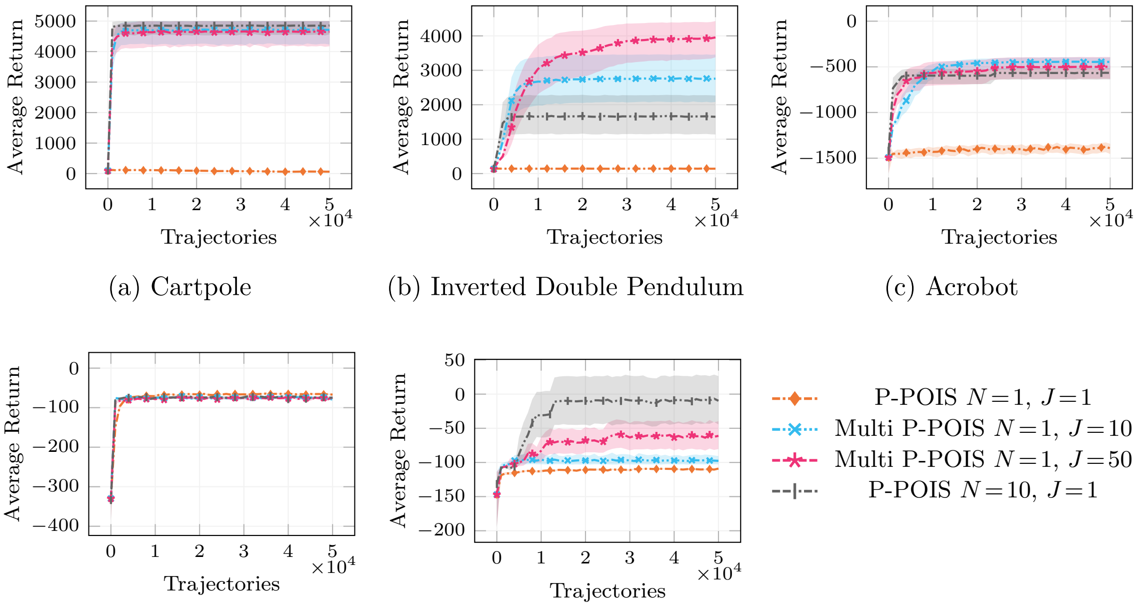
Table 6: Performance of POIS compared with Duan et al. (2016) on *deep neural policies* (5 runs, 95% c.i.). In **bold**, the performances that are not statistically significantly different from the best algorithm in each task.

Environment	$N = 1,\ J = 1$	$N = 1,\ J = 10$	$N = 1,\ J = 50$	$N = 10,\ J = 1$
Cartpole	0.0001	0.0001	0.001	0.1
Inverted Double Pendulum	0.001	0.001	0.0005	0.05
Acrobot	0.99	0.01	0.05	0.6
Mountain Car	0.6	0.0005	0.0005	0.05
Inverted Pendulum	0.99	0.2	0.1	0.1

Table 7: Hyperparameter value of the individual algorithms employed in the experiments shown in Figure 7. For all versions of POIS we report the value of δ.

In all the considered tasks, single P-POIS with unit batch size fails miserably, except in Mountain Car, in which all the tested variants show comparable behavior.[16] We can see that MIS is able to remedy this lack of samples, at least partially. In Cartpole, a capacity of 10 is enough to achieve optimal performance. The results in the Inverted Double Pendulum task are the most aligned with the intuition: a capacity of 10 yields a significant improvement compared to the single–IS case, but the latter becomes superior once equipped with a batch size of 10 fresh samples. In addition, a larger capacity ($J = 50$) is beneficial. Acrobot yields similar, though less clear, results. The outcomes in the Inverted Pendulum task are more surprising: a capacity of 10 is better than both a capacity of 50 and the large–batch variant. This could be explained by the paramount importance of exploration in this task if we consider the variance of the objective function estimate as a passive form of exploration. Another possibility is that adding more behavioral hyper–policies makes it harder to optimize the objective function. Both aspects should be further inquired by future work.

[16] Note, however, that different values of the hyper–parameter δ have been selected, via grid search, for the different variants of P-POIS. Typically, larger capacities come with larger values of δ (see Appendix H).



(a) Cartpole

(b) Inverted Double Pendulum

(c) Acrobot

(d) Mountain Car

(e) Inverted Pendulum

Figure 7: Average return as a function of the number of trajectories for P-POIS with *linear policy* for different values of the batch size N and the MIS capacity J (20 runs, 95% c.i.).

8. Conclusion

In this paper, we presented a new actor–only policy optimization algorithm, POIS, which alternates on–line and off–line optimization in order to efficiently exploit the collected trajectories, and can be used in combination with action–based and parameter–based exploration. In contrast to the state–of–the–art algorithms, POIS has a strong theoretical grounding, since its surrogate objective function derives from a statistical bound on the estimated performance, capable of capturing the uncertainty induced by importance sampling. Since POIS makes fewer compromises towards practicality compared to more popular deep RL algorithms, the latter are still expected to perform better overall. However, the experimental evaluation showed that POIS, in both its versions (action–based and parameter–based), is able to achieve a performance comparable with TRPO, PPO, and other classical algorithms on continuous control tasks of moderate size. We have proposed two extensions to the original POIS algorithm (Metelli et al., 2018), both intended to make an even more efficient use of the samples: per–decision importance weighting and multiple importance weighting. The pros and cons of these variants have been studied both theoretically and empirically. We believe that this work contributes to a deeper understanding of modern policy optimization and to the development of effective and scalable policy search methods. There is still room to reduce the gap between theory and practice, especially w.r.t. scalability to complex control tasks. Scalability issues manifest themselves differently in the action–based and in the parameter–based frameworks, as mentioned in Section 5. In the former, long–horizon tasks are the main challenge and step–based approaches should be developed to overcome this *curse of horizon* (Liu et al., 2018). In the parameter–based case, the task length is irrelevant and the main challenge is to learn policies with many parameters, which may be necessary for complex control tasks, e.g., vision–based ones. We expect compact policy representations, such as fingerprinting (Harb et al., 2020) to play an important role in making parameter based–algorithms scalable. Finally, additional future–work directions include finding a compromise between risk–aversion and exploration and a better understanding of the role of the batch–size hyper–parameter and of the optimization challenges introduced by importance–weighted objectives.

Acknowledgments

We thank Francesco Faccio for contributing to the idea of the POIS algorithm. This work has been partially supported by the Italian MIUR PRIN 2017 Project ALGADIMAR "Algorithms, Games, and Digital Market". We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40cm, Titan XP and Tesla V100 used for this research.

Index of the Appendix

In the following, we briefly recap the contents of the Appendix.

- Appendix A provides, in Table 8, a more detailed comparison of POIS with the policy–search algorithms, summarizing some features of the considered methods.
- Appendix B reports all proofs and derivations.
- Appendix C provides an analysis of the distribution of the importance weights in the case of univariate Gaussian behavioral and target distributions.
- Appendix D shows some bounds on bias and variance for the self–normalized importance sampling estimator and provides a high-confidence bound.
- Appendix E reports some details about the estimation of the Rényi divergence.
- Appendix F provides the complete asymptotic analysis of the variance of A-POIS and D-POIS.
- Appendix G illustrates some implementation details of POIS, in particular line search algorithms and practical versions of P-POIS.
- Appendix H provides the hyperparameters used in the experiments and further results.

Appendix A. Related Works Table

Algorithm	Action/Parameter based	On/Off policy	Optimization problem	Critic	Step/Episode based
REINFORCE/ G(PO)MDP (Williams, 1992; Baxter and Bartlett, 2001)	action–based	on–policy	$\max \widehat{\mathbb{E}}_{\tau\sim\boldsymbol{\theta}}\left[R(\tau)\right]$	No	episode–based
TRPO (Schulman et al., 2015a)	action–based	on–policy	$\max \widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(a_t\|s_t)\widehat{A}(s_t,a_t)\right]$ s.t. $\widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\left[D_{\mathrm{KL}}(\pi_{\boldsymbol{\theta}'}(\cdot\|s_t)\\|\pi_{\boldsymbol{\theta}}(\cdot\|s_t))\right] \leqslant \delta$	Yes	step–based
PPO (Schulman et al., 2017)	action–based	on/off–policy	$\max \widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\Big[\min\Big\{w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(a_t\|s_t)\widehat{A}(s_t,a_t),$ $\mathrm{clip}\left(w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(a_t\|s_t),1-\epsilon,1+\epsilon\right)\widehat{A}(s_t,a_t)\Big\}\Big]$	Yes	step–based
DDPG (Lillicrap et al., 2015)	action–based	off–policy	$\max \widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\left[\pi_{\boldsymbol{\theta}'}(a_t\|s_t)\widehat{Q}(s_t,a_t)\right]$	Yes	step–based
REPS (Peters et al., 2010)	action–based	on–policy	$\max \widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\left[R(s_t,a_t)\right]$ s.t. $\widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\left[D_{\mathrm{KL}}(d_D^{\boldsymbol{\theta}'}(s_t,a_t)\\|d_D^{\boldsymbol{\theta}}(s_t,a_t))\right] \leqslant \delta$	Yes	step–based
RWR (Peters and Schaal, 2007)	action–based	on–policy	$\max \widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}\left[\beta \exp\left(-\beta R(s_t,a_t)\right)\right]$	No	step–based
A-POIS	action–based	on/off–policy	$\max \widehat{\mathbb{E}}_{\tau\sim\boldsymbol{\theta}}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau)R(\tau)\right] - \lambda\sqrt{\widehat{d_2}(p(\cdot\|\boldsymbol{\theta}')\\|p(\cdot\|\boldsymbol{\theta}))/N}$	No	episode–based
D-POIS	action–based	on/off–policy	$\max \widehat{\mathbb{E}}_{\tau\sim\boldsymbol{\theta}}\left[\sum_{t=0}^{H-1}\gamma^t w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau,t)R(s_t,a_t)\right] -$ $\lambda\sqrt{\sum_{t=0}^{H-1} c_t\widehat{d_2}(p(\cdot\|\boldsymbol{\theta}',t)\\|p(\cdot\|\boldsymbol{\theta},t))/N}$	No	episode–based
PGPE (Sehnke et al., 2008)	parameter–based	on–policy	$\max \widehat{\mathbb{E}}_{\boldsymbol{\theta}\sim\boldsymbol{\rho},\tau\sim\boldsymbol{\theta}}\left[R(\tau)\right]$	No	episode–based
IW-PGPE (Zhao et al., 2013)	parameter–based	on/off–policy	$\max \widehat{\mathbb{E}}_{\boldsymbol{\theta}\sim\boldsymbol{\rho},\tau\sim\boldsymbol{\theta}}\left[w_{\boldsymbol{\rho}'/\boldsymbol{\rho}}(\boldsymbol{\theta})R(\tau)\right]$	No	episode–based
P-POIS	parameter–based	on/off–policy	$\max \widehat{\mathbb{E}}_{\boldsymbol{\theta}\sim\boldsymbol{\rho},\tau\sim\boldsymbol{\theta}}\left[w_{\boldsymbol{\rho}'/\boldsymbol{\rho}}(\boldsymbol{\theta})R(\tau)\right] - \lambda\sqrt{d_2(\nu_{\boldsymbol{\rho}'}\\|\nu_{\boldsymbol{\rho}})/N}$	No	episode–based

Table 8: Comparison of some policy optimization algorithms according to different dimensions. For brevity, we will denote $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(a_t|s_t) = \frac{\pi_{\boldsymbol{\theta}'}(a_t|s_t)}{\pi_{\boldsymbol{\theta}}(a_t|s_t)}$. For episode–based algorithms we will indicate with $\widehat{\mathbb{E}}_{\tau\sim\boldsymbol{\theta}}$ the empirical average over trajectories collected with $\pi_{\boldsymbol{\theta}}$. For step–based algorithms $\widehat{\mathbb{E}}_{t\sim\boldsymbol{\theta}}$ is the empirical average collecting samples with $\pi_{\boldsymbol{\theta}}$. For parameter–based algorithms we indicate with $\widehat{\mathbb{E}}_{\boldsymbol{\theta}\sim\boldsymbol{\rho},\tau\sim\boldsymbol{\theta}}$ the empirical expectation taken w.r.t. policy parameter $\boldsymbol{\theta}$ sampled from the hyperpolicy $\nu_{\boldsymbol{\rho}}$ and trajectory τ collected with $\pi_{\boldsymbol{\theta}}$. For the actor–critic architectures, $\widehat{Q}$ and $\widehat{A}$ are the estimated Q–function and advantage function.

Appendix B. Additional Proofs and Derivations

In this appendix, we report the proofs that are omitted in the main paper.

B.1. Proofs of Section 4

Lemma 3 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ be i.i.d. random variables sampled from Q, and $f : \mathcal{X} \rightarrow \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any clipping threshold $M < +\infty$ and $N > 0$, the bias and the variance of the estimator $\check{\mu}_{P/Q}$ can be upper bounded as:*

$$\begin{aligned}
&\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\check{\mu}_{P/Q}\right] - \mathop{\mathbb{E}}_{x\sim P}[f(x)] \right| \leqslant \|f\|_\infty \frac{d_2(P\|Q)}{M}, \\
&\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\check{\mu}_{P/Q}\right] \leqslant \|f\|_\infty^2 \frac{d_2(P\|Q)}{N}.
\end{aligned}$$

Proof Concerning the bias, we need to extend the proof of Lemma 2 of Papini et al. (2019) since we are looking for a double–sided result:

$$\begin{aligned}
\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\check{\mu}_{P/Q}\right] - \mathop{\mathbb{E}}_{x\sim P}[f(x)] \right| &= \left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\check{\mu}_{P/Q}\right] - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right] \right| \\
&= \left| \mathop{\mathbb{E}}_{x\sim Q}\left[\left(\omega_{P/Q}(x) - \min\left\{M, \omega_{P/Q}(x)\right) f(x)\right)\right] \right| \\
&= \left| \mathop{\mathbb{E}}_{x\sim Q}\left[(\omega_{P/Q}(x) - M) f(x) \mathbb{1}\left\{\omega_{P/Q}(x) > M\right\}\right] \right| \qquad \text{(P.17)} \\
&\leqslant \|f\|_\infty \mathop{\mathbb{E}}_{x\sim Q}\left[\left|\omega_{P/Q}(x) - M\right| \mathbb{1}\left\{\omega_{P/Q}(x) > M\right\}\right] \qquad \text{(P.18)} \\
&\leqslant \|f\|_\infty \mathop{\mathbb{E}}_{x\sim Q}\left[\omega_{P/Q}(x) \mathbb{1}\left\{\omega_{P/Q}(x) > M\right\}\right] \qquad \text{(P.19)} \\
&\leqslant \|f\|_\infty \mathop{\mathbb{E}}_{x\sim Q}\left[\omega_{P/Q}(x)^2 \omega_{P/Q}(x)^{-1} \mathbb{1}\left\{\omega_{P/Q}(x) > M\right\}\right] \\
&\leqslant \|f\|_\infty \mathop{\mathbb{E}}_{x\sim Q}\left[\omega_{P/Q}(x)^2\right] M^{-1} \qquad \text{(P.20)} \\
&\leqslant \|f\|_\infty d_2(P\|Q) M^{-1},
\end{aligned}$$

where line (P.17) follows from observing that the weight difference is either zero or $\omega_{P/Q}(x) - M$ based on whether $\omega_{P/Q}(x) > M$, line (P.18) is an application of Hölder's inequality, line (P.19) is obtained by observing that under the indicator function we have that $\omega_{P/Q}(x) > M$, and line (P.20) comes from observing that if $\omega_{P/Q}(x) > M$, we have $\omega_{P/Q}(x)^{-1} < M^{-1}$.

Concerning the variance, the derivation is analogous to that of Lemma 2 of Papini et al. (2019), by setting $\epsilon = 1$. ■

Theorem 3 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathscr{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ be i.i.d. random variables sampled from Q, and $f : \mathcal{X} \rightarrow \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $0 < \delta \leqslant 1$ and*

$N > 0$, using a clipping threshold $M(N,\delta) = \sqrt{\frac{3Nd_2(P\|Q)}{2\log\frac{1}{\delta}}}$, with probability at least $1-\delta$ it holds that:

$$\mathop{\mathbb{E}}_{x\sim P}[f(x)] \geqslant \underbrace{\frac{1}{N}\sum_{i=1}^{N}\breve{w}_{P/Q}(x_i)f(x_i)}_{\breve{\mu}_{P/Q}} - \|f\|_\infty(2+\sqrt{3})\sqrt{\frac{2d_2(P\|Q)\log\frac{1}{\delta}}{3N}}. \tag{22}$$

Proof We start from the version of Bernstein's inequality of Theorem 2.8 by Chung and Lu (2006) applied to the random variable $\breve{\mu}_{P/Q} = \frac{1}{N}\sum_{i=1}^{N}\breve{w}_{P/Q}(x_i)f(x_i)$ and let $\check{\lambda} = \lambda - \left|\mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right]\right|$:

$$\begin{aligned}
\Pr\left(\breve{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}[f(x)] \geqslant \lambda\right) &= \Pr\left(\breve{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right] \geqslant \mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right] + \lambda\right) \\
&\leqslant \Pr\left(\breve{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right] \geqslant \lambda - \left|\mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right]\right|\right) \\
&= \Pr\left(\breve{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right] \geqslant \check{\lambda}\right).
\end{aligned}$$

Now we apply Bernstein's inequality:

$$\begin{aligned}
\Pr\left(\breve{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}[f(x)] \geqslant \lambda\right) &\leqslant \Pr\left(\breve{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right] \geqslant \check{\lambda}\right) \\
&\leqslant \exp\left(-\frac{\check{\lambda}^2 N}{2\left(\mathbb{E}_{x\sim Q}[\breve{\omega}_{P/Q}(x)^2 f(x)^2] + \frac{\check{\lambda}}{3}\|\breve{\omega}_{P/Q}(x)f(x)\|_\infty\right)}\right) \\
&\leqslant \exp\left(-\frac{\check{\lambda}^2 N}{2\left(\|f\|_\infty^2 d_2(P\|Q) + \frac{\check{\lambda}}{3}M\|f\|_\infty\right)}\right),
\end{aligned}$$

where we applied $\|\breve{\omega}_{P/Q}(x)f(x)\|_\infty \leqslant \|f\|_\infty M$ and $\mathbb{E}_{x\sim Q}[\breve{\omega}_{P/Q}(x)^2 f(x)^2] \leqslant \|f\|_\infty^2 d_2(P\|Q)$. By calling $\delta = \exp\left(-\frac{\check{\lambda}^2 N}{2\left(\|f\|_\infty^2 d_2(P\|Q) + \frac{\check{\lambda}}{3}M\|f\|_\infty\right)}\right)$ and solving for $\check{\lambda}$, retaining the positive solution only, we obtain:

$$\begin{aligned}
\check{\lambda} &= \frac{M\|f\|_\infty \log\frac{1}{\delta}}{3N} + \frac{1}{3N}\sqrt{18\|f\|_\infty^2 d_2(P\|Q)N\log\frac{1}{\delta} + M^2\|f\|_\infty^2\left(\log\frac{1}{\delta}\right)^2} \\
&\leqslant \frac{2\|f\|_\infty M\log\frac{1}{\delta}}{3N} + \|f\|_\infty\sqrt{\frac{2d_2(P\|Q)\log\frac{1}{\delta}}{N}},
\end{aligned}$$

where we applied the subadditivity of the square root. Now, we consider the general expression of the truncation $M = \zeta\sqrt{\frac{d_2(P\|Q)N}{\log\frac{1}{\delta}}}$ where $\zeta > 0$ is a parameter whose value will be determined later. We now substitute the expression of M and the bound on the bias:

$$\lambda \leqslant \left|\mathop{\mathbb{E}}_{x\sim P}[f(x)] - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\breve{\mu}_{P/Q}\right]\right| + \frac{2\|f\|_\infty M\log\frac{1}{\delta}}{3N} + \|f\|_\infty\sqrt{\frac{2d_2(P\|Q)\log\frac{1}{\delta}}{N}}$$

$$\leqslant \|f\|_\infty \left(\frac{1}{\zeta} + \frac{2}{3}\zeta + \sqrt{2} \right) \sqrt{\frac{d_2(P\|Q) \log \frac{1}{\delta}}{N}}.$$

The result is obtained by minimizing the expression depending on ζ, which yields the value $\zeta = \sqrt{\frac{3}{2}}$. ■

Theorem 4 *Let $p_{\boldsymbol{\omega}}$ be a p.d.f. differentiable w.r.t. $\boldsymbol{\omega} \in \Omega$. Then, it holds that, for the Rényi divergence:*

$$D_\alpha(p_{\boldsymbol{\omega}'}\|p_{\boldsymbol{\omega}}) = \frac{\alpha}{2} \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right)^T \mathcal{F}(\boldsymbol{\omega}) \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right) + o(\|\boldsymbol{\omega}' - \boldsymbol{\omega}\|_2^2),$$

and for the exponentiated Rényi divergence:

$$d_\alpha(p_{\boldsymbol{\omega}'}\|p_{\boldsymbol{\omega}}) = 1 + \frac{\alpha}{2} \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right)^T \mathcal{F}(\boldsymbol{\omega}) \left(\boldsymbol{\omega}' - \boldsymbol{\omega}\right) + o(\|\boldsymbol{\omega}' - \boldsymbol{\omega}\|_2^2).$$

Proof We need to compute the second–order Taylor expansion of the α–Rényi divergence. We start considering the term:

$$I(\boldsymbol{\omega}') = \int_{\mathcal{X}} \left(\frac{p_{\boldsymbol{\omega}'}(x)}{p_{\boldsymbol{\omega}}(x)} \right)^\alpha p_{\boldsymbol{\omega}}(x) \,\mathrm{d}x = \int_{\mathcal{X}} p_{\boldsymbol{\omega}'}(x)^\alpha p_{\boldsymbol{\omega}}(x)^{1-\alpha} \,\mathrm{d}x. \tag{P.21}$$

The gradient is given by:

$$\nabla_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}') = \int_{\mathcal{X}} \nabla_{\boldsymbol{\omega}'} p_{\boldsymbol{\omega}'}(x)^\alpha p_{\boldsymbol{\omega}}(x)^{1-\alpha} \,\mathrm{d}x = \alpha \int_{\mathcal{X}} p_{\boldsymbol{\omega}'}(x)^{\alpha-1} p_{\boldsymbol{\omega}}(x)^{1-\alpha} \nabla_{\boldsymbol{\omega}'} p_{\boldsymbol{\omega}'}(x) \,\mathrm{d}x.$$

Thus, $\nabla_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}')|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} = \mathbf{0}$. We now compute the Hessian:

$$\begin{aligned}
\mathcal{H}_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}') &= \nabla_{\boldsymbol{\omega}'} \nabla_{\boldsymbol{\omega}'}^T I(\boldsymbol{\omega}') \\
&= \alpha \nabla_{\boldsymbol{\omega}'} \int_{\mathcal{X}} p_{\boldsymbol{\omega}'}(x)^{\alpha-1} p_{\boldsymbol{\omega}}(x)^{1-\alpha} \nabla_{\boldsymbol{\omega}'}^T p_{\boldsymbol{\omega}'}(x) \,\mathrm{d}x \\
&= \alpha \int_{\mathcal{X}} \Big((\alpha-1) p_{\boldsymbol{\omega}'}(x)^{\alpha-2} p_{\boldsymbol{\omega}}(x)^{1-\alpha} \nabla_{\boldsymbol{\omega}'} p_{\boldsymbol{\omega}'}(x) \nabla_{\boldsymbol{\omega}'}^T p_{\boldsymbol{\omega}'}(x) \\
&\qquad + p_{\boldsymbol{\omega}'}(x)^{\alpha-1} p_{\boldsymbol{\omega}}(x)^{1-\alpha} \mathcal{H}_{\boldsymbol{\omega}'} p_{\boldsymbol{\omega}'}(x) \Big) \,\mathrm{d}x.
\end{aligned}$$

Evaluating the Hessian in $\boldsymbol{\omega}$ we have:

$$\begin{aligned}
\mathcal{H}_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}')|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} &= \alpha(\alpha-1) \int_{\mathcal{X}} p_{\boldsymbol{\omega}}(x)^{-1} \nabla_{\boldsymbol{\omega}} p_{\boldsymbol{\omega}}(x) \nabla_{\boldsymbol{\omega}}^T p_{\boldsymbol{\omega}}(x) \,\mathrm{d}x \\
&= \alpha(\alpha-1) \int_{\mathcal{X}} p_{\boldsymbol{\omega}}(x) \nabla_{\boldsymbol{\omega}} \log p_{\boldsymbol{\omega}}(x) \nabla_{\boldsymbol{\omega}}^T \log p_{\boldsymbol{\omega}}(x) \,\mathrm{d}x = \alpha(\alpha-1)\mathcal{F}(\boldsymbol{\omega}).
\end{aligned}$$

Now, $D_\alpha(p_{\boldsymbol{\omega}'}\|p_{\boldsymbol{\omega}}) = \frac{1}{\alpha-1} \log I(\boldsymbol{\omega}')$. Thus:

$$\nabla_{\boldsymbol{\omega}'} D_\alpha(p_{\boldsymbol{\omega}'}\|p_{\boldsymbol{\omega}})|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} = \frac{1}{\alpha-1} \frac{\nabla_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}')}{I(\boldsymbol{\omega}')} \bigg|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} = \mathbf{0},$$

$$\begin{aligned}\mathcal{H}_{\boldsymbol{\omega}'} D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} &= \frac{1}{\alpha-1} \frac{I(\boldsymbol{\omega}')\mathcal{H}_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}') + \nabla_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}') \nabla_{\boldsymbol{\omega}'}^T I(\boldsymbol{\omega}')}{\left(I(\boldsymbol{\omega}')\right)^2}\Bigg|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} \\ &= \frac{1}{\alpha-1} \mathcal{H}_{\boldsymbol{\omega}'} I(\boldsymbol{\omega}')|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} = \alpha \mathcal{F}(\boldsymbol{\omega}),\end{aligned}$$

having observed that $I(\boldsymbol{\omega}) = 1$. For what concerns the $d_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})$, we have:

$$\begin{aligned}\nabla_{\boldsymbol{\omega}'} d_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} &= \nabla_{\boldsymbol{\omega}'} \exp\left(D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})\right)|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} \\ &= \exp\left(D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})\right) \nabla_{\boldsymbol{\omega}'} D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} = \mathbf{0},\end{aligned}$$

$$\begin{aligned}&\mathcal{H}_{\boldsymbol{\omega}'} d_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} = \mathcal{H}_{\boldsymbol{\omega}'} \exp\left(D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})\right)|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} \\ &\quad = \exp\left(D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})\right) \left(\mathcal{H}_{\boldsymbol{\omega}'} D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}}) + \nabla_{\boldsymbol{\omega}'} D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}}) \nabla_{\boldsymbol{\omega}'}^T D_\alpha(p_{\boldsymbol{\omega}'} \| p_{\boldsymbol{\omega}})\right)|_{\boldsymbol{\omega}'=\boldsymbol{\omega}} \\ &\quad = \alpha \mathcal{F}(\boldsymbol{\omega}).\end{aligned}$$

■

B.2. Proofs of Section 5

Proposition 1 *Let $p(\cdot|\boldsymbol{\theta})$ and $p(\cdot|\boldsymbol{\theta}')$ be the behavioral and target trajectory probability density functions. If $p(\cdot|\boldsymbol{\theta}') \ll p(\cdot|\boldsymbol{\theta})$ and $H < +\infty$, then, for any $\alpha \in [0, +\infty]$ it holds that:*

$$d_\alpha\left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta})\right) \leqslant \sup_{s \in \mathcal{S}} \left\{ d_\alpha\left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s)\right)\right\}^H.$$

Proof We prove the proposition by induction on the horizon H. We define $d_{\alpha,H}$ as the α–Rényi divergence at horizon H. For $H = 1$ we have:

$$\begin{aligned}d_{\alpha,1}\left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta})\right) &= \int_{\mathcal{S}} D(s_0) \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_0|s_0) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_0|s_0)}{\pi_{\boldsymbol{\theta}}(a_0|s_0)}\right)^\alpha \int_{\mathcal{S}} P(s_1|s_0, a_0) \,\mathrm{d}s_1 \,\mathrm{d}a_0 \,\mathrm{d}s_0 \\ &= \int_{\mathcal{S}} D(s_0) \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_0|s_0) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_0|s_0)}{\pi_{\boldsymbol{\theta}}(a_0|s_0)}\right)^\alpha \mathrm{d}a_0 \,\mathrm{d}s_0 \\ &\leqslant \int_{\mathcal{S}} D(s_0) \,\mathrm{d}s_0 \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_0|s) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_0|s)}{\pi_{\boldsymbol{\theta}}(a_0|s)}\right)^\alpha \mathrm{d}a_0 \\ &\leqslant \sup_{s \in \mathcal{S}} d_\alpha\left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s)\right),\end{aligned}$$

where the last but one passage follows from Holder's inequality. Suppose that the proposition holds for any $H' < H$, let us prove the proposition for H.

$$\begin{aligned}d_{\alpha,H}\Big(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta})\Big) &= \int_{\mathcal{S}} D(s_0) \ \ldots \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_{H-2}|s_{H-2}) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_{H-2}|s_{H-2})}{\pi_{\boldsymbol{\theta}}(a_{H-2}|s_{H-2})}\right)^\alpha \\ &\times \int_{\mathcal{S}} P(s_{H-1}|s_{H-2}, a_{H-2}) \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_{H-1}|s_{H-1}) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_{H-1}|s_{H-1})}{\pi_{\boldsymbol{\theta}}(a_{H-1}|s_{H-1})}\right)^\alpha \\ &\times \int_{\mathcal{S}} P(s_H|s_{H-1}, a_{H-1}) \,\mathrm{d}s_0 \ldots \,\mathrm{d}s_{H-1} \,\mathrm{d}a_{H-2} \,\mathrm{d}s_{H-1} \,\mathrm{d}a_{H-1} \,\mathrm{d}s_H\end{aligned}$$

$$\begin{aligned}
&= \int_{\mathcal{S}} D(s_0) \ \dots \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_{H-2}|s_{H-2}) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_{H-2}|s_{H-2})}{\pi_{\boldsymbol{\theta}}(a_{H-2}|s_{H-2})} \right)^{\alpha} \int_{\mathcal{S}} P(s_{H-1}|s_{H-2}, a_{H-2}) \\
&\quad \times \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_{H-1}|s_{H-1}) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_{H-1}|s_{H-1})}{\pi_{\boldsymbol{\theta}}(a_{H-1}|s_{H-1})} \right)^{\alpha} \mathrm{d}s_0 \dots \mathrm{d}s_{H-1} \, \mathrm{d}a_{H-2} \, \mathrm{d}s_{H-1} \, \mathrm{d}a_{H-1} \\
&\leqslant \int_{\mathcal{S}} D(s_0) \ \dots \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_{H-2}|s_{H-2}) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_{H-2}|s_{H-2})}{\pi_{\boldsymbol{\theta}}(a_{H-2}|s_{H-2})} \right)^{\alpha} \int_{\mathcal{S}} P(s_{H-1}|s_{H-2}, a_{H-2}) \\
&\quad \times \mathrm{d}s_0 \dots \mathrm{d}s_{H-1} \, \mathrm{d}a_{H-2} \, \mathrm{d}s_{H-1} \times \sup_{s \in \mathcal{S}} \int_{\mathcal{A}} \pi_{\boldsymbol{\theta}}(a_{H-1}|s) \left(\frac{\pi_{\boldsymbol{\theta}'}(a_{H-1}|s)}{\pi_{\boldsymbol{\theta}}(a_{H-1}|s)} \right)^{\alpha} \mathrm{d}a_{H-1} \\
&\leqslant d_{\alpha,H-1} \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right) \sup_{s \in \mathcal{S}} d_{\alpha} \left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s) \right) \\
&\leqslant \left(\sup_{s \in \mathcal{S}} d_{\alpha} \left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s) \right) \right)^{H},
\end{aligned}$$

where we applied Holder's inequality again and the last passage is obtained for the inductive hypothesis. ■

Proposition 2 *Let $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau)$ be the importance weight of trajectory τ and $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t)$ be the per–decision importance weight of trajectory τ up to time $t \in \{0, 1, ..., H-1\}$. Then, it holds that:*

$$\operatorname*{\mathbb{V}ar}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t) \right] \leqslant \operatorname*{\mathbb{V}ar}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) \right].$$

Proof First, recall that $w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) = w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, H-1)$. Thus, given that $t \in \{1, 2, ..., H-1\}$ by definition, we can reduce the proof to $\mathbb{V}\mathrm{ar}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t-1) \right] \leqslant \mathbb{V}\mathrm{ar}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t) \right]$. Recalling that $\mathbb{E}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t) \right] = 1$ for all t, we can just compare the second moments. Thus, we prove:

$$\operatorname*{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t-1)^2 \right] \leqslant \operatorname*{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t)^2 \right].$$

We start by unrolling the trajectory probability,

$$\begin{aligned}
\operatorname*{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau, t)^2 \right] &= \int p(\tau|\boldsymbol{\theta}) \left(\prod_{t'=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t'}|s_{\tau,t'})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t'}|s_{\tau,t'})} \right)^2 \mathrm{d}\tau \\
&= \int \mu(s_{\tau,0}) \pi_{\boldsymbol{\theta}}(a_{\tau,0}|s_{\tau,0}) P(s_{\tau,1}|s_{\tau,0}, a_{\tau,0}) \dots \pi_{\boldsymbol{\theta}}(a_{\tau,H-1}|s_{\tau,H-1}) \\
&\quad \times P(s_{\tau,H}|a_{\tau,H-1}, s_{\tau,H-1}) \left(\prod_{t'=0}^{t} \frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t'}|s_{\tau,t'})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t'}|s_{\tau,t'})} \right)^2 \\
&\quad \times \mathrm{d}s_{\tau,0} \mathrm{d}a_{\tau,0} \mathrm{d}s_{\tau,1} \dots \mathrm{d}s_{\tau,H-1} \mathrm{d}a_{\tau,H-1} \mathrm{d}s_{\tau,H}.
\end{aligned}$$

Considering that the squared importance weight is dependent only on $t' \leqslant t$, we can integrate away all the terms with $t' \geqslant t$, i.e.

$$\int P(s_{\tau,t+1}|s_{\tau,t}, a_{\tau,t}) \pi_{\boldsymbol{\theta}}(a_{\tau,t+1}|s_{\tau,t+1}) \dots$$

$$\times \pi_{\boldsymbol{\theta}}(a_{\tau,H-1}|s_{\tau,H-1})P(s_{\tau,H}|a_{\tau,H-1},s_{\tau,H-1})$$
$$\times \mathrm{d}s_{\tau,t+1}\mathrm{d}a_{\tau,t+1}\dots\mathrm{d}s_{\tau,H-1}\mathrm{d}a_{\tau,H-1}\mathrm{d}s_{\tau,H} = 1.$$

Therefore, we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau,t)^2\right] &= \int \mu(s_{\tau,0})\pi_{\boldsymbol{\theta}}(a_{\tau,0}|s_{\tau,0})P(s_{\tau,1}|s_{\tau,0},a_{\tau,0})\dots\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})\\
&\quad\times\left(\prod_{t'=0}^{t}\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t'}|s_{\tau,t'})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t'}|s_{\tau,t'})}\right)^2\mathrm{d}s_{\tau,0}\mathrm{d}a_{\tau,0}\mathrm{d}s_{\tau,1}\dots\mathrm{d}s_{\tau,t-1}\mathrm{d}a_{\tau,t-1}\\
&= \int \mu(s_{\tau,0})\pi_{\boldsymbol{\theta}}(a_{\tau,0}|s_{\tau,0})\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,0}|s_{\tau,0})}{\pi_{\boldsymbol{\theta}}(a_{\tau,0}|s_{\tau,0})}\right)^2P(s_{\tau,1}|s_{\tau,0},a_{\tau,0})\dots\\
&\quad\times\pi_{\boldsymbol{\theta}}(a_{\tau,t-1}|s_{\tau,t-1})\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t-1}|s_{\tau,t-1})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t-1}|s_{\tau,t-1})}\right)^2\\
&\quad\times P(s_{\tau,t}|s_{\tau,t-1},a_{\tau,t-1})\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t}|s_{\tau,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})}\right)^2\\
&\quad\times\mathrm{d}s_{\tau,0}\mathrm{d}a_{\tau,0}\mathrm{d}s_{\tau,1}\dots\mathrm{d}s_{\tau,t-1}\mathrm{d}a_{\tau,t-1}\mathrm{d}s_{\tau,t}\mathrm{d}a_{\tau,t}.
\end{aligned}$$

Now, using the fact that $\int\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t}|s_{\tau,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t}|s_{\tau,t})}\right)^2\mathrm{d}a_{\tau,t} = d_2(\pi_{\boldsymbol{\theta}'}(\cdot|s_{\tau,t})\|\pi_{\boldsymbol{\theta}}(\cdot|s_{\tau,t})) \geqslant 1$ and that $\int P(s_{\tau,t}|s_{\tau,t-1},a_{\tau,t-1})\mathrm{d}s_{\tau,t} = 1$, we can finally obtain:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau,t)^2\right] &\geqslant \int \mu(s_{\tau,0})\pi_{\boldsymbol{\theta}}(a_{\tau,0}|s_{\tau,0})\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,0}|s_{\tau,0})}{\pi_{\boldsymbol{\theta}}(a_{\tau,0}|s_{\tau,0})}\right)^2P(s_{\tau,1}|s_{\tau,0},a_{\tau,0})\dots\\
&\quad\times\pi_{\boldsymbol{\theta}}(a_{\tau,t-1}|s_{\tau,t-1})\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau,t-1}|s_{\tau,t-1})}{\pi_{\boldsymbol{\theta}}(a_{\tau,t-1}|s_{\tau,t-1})}\right)^2\mathrm{d}s_{\tau,0}\mathrm{d}a_{\tau,0}\mathrm{d}s_{\tau,1}\dots\mathrm{d}s_{\tau,t-1}\mathrm{d}a_{\tau,t-1}\\
&= \mathop{\mathbb{E}}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau,t-1)^2\right].
\end{aligned}$$

■

Appendix C. Analysis of the IS estimator for Gaussian distributions

In this appendix, we analyze the behavior of the importance weights when the behavioral and target distributions are Gaussians. We start by providing a closed–form expression for the Rényi divergence between multivariate Gaussian distributions (Burbea, 1984). Let $P \sim \mathcal{N}(\boldsymbol{\mu}_P,\boldsymbol{\Sigma}_P)$, $Q \sim \mathcal{N}(\boldsymbol{\mu}_Q,\boldsymbol{\Sigma}_Q)$ and $\alpha \in [0,\infty]$:

$$D_\alpha(P\|Q) = \frac{\alpha}{2}(\boldsymbol{\mu}_P-\boldsymbol{\mu}_Q)^T\boldsymbol{\Sigma}_\alpha^{-1}(\boldsymbol{\mu}_P-\boldsymbol{\mu}_Q) - \frac{1}{2(\alpha-1)}\log\frac{\det(\boldsymbol{\Sigma}_\alpha)}{\det(\boldsymbol{\Sigma}_P)^{1-\alpha}\det(\boldsymbol{\Sigma}_Q)^\alpha}, \qquad (39)$$

where $\boldsymbol{\Sigma}_\alpha = \alpha\boldsymbol{\Sigma}_Q + (1-\alpha)\boldsymbol{\Sigma}_P$ under the assumption that $\boldsymbol{\Sigma}_\alpha$ is positive–definite.

From now on, we will focus on univariate Gaussian distributions and we provide a closed–form expression for the importance weights and their probability density function

f_w. We consider $Q \sim \mathcal{N}(\mu_Q, \sigma_Q^2)$ as behavioral distribution and $P \sim \mathcal{N}(\mu_P, \sigma_P^2)$ as target distribution. We assume that $\sigma_Q^2, \sigma_P^2 > 0$ and we consider the two cases: unequal variances and equal variances. For brevity, we will indicate with $w(x)$ the weight $w_{P/Q}(x)$.

C.1. Unequal variances

When $\sigma_Q^2 \neq \sigma_P^2$, the expression of the importance weights is given by:

$$w(x) = \frac{\sigma_Q}{\sigma_P} \exp\left(\frac{1}{2} \frac{(\mu_P - \mu_Q)^2}{\sigma_Q^2 - \sigma_P^2} \right) \exp\left(-\frac{1}{2} \frac{\sigma_Q^2 - \sigma_P^2}{\sigma_Q^2 \sigma_P^2} \left(x - \frac{\sigma_Q^2 \mu_P - \sigma_P^2 \mu_Q}{\sigma_Q^2 - \sigma_P^2} \right)^2 \right), \tag{40}$$

for $x \sim Q$. Let us first notice two distinct situations: if $\sigma_Q^2 - \sigma_P^2 > 0$ the weight $w(x)$ is upper bounded by $A = \frac{\sigma_Q}{\sigma_P} \exp\left(\frac{1}{2} \frac{(\mu_P - \mu_Q)^2}{\sigma_Q^2 - \sigma_P^2} \right)$, whereas if $\sigma_Q^2 - \sigma_P^2 < 0$, $w(x)$ is unbounded but it admits a minimum of value A. Let us investigate the probability density function.

Proposition 3 *Let $Q \sim \mathcal{N}(\mu_Q, \sigma_Q^2)$ be the behavioral distribution and $P \sim \mathcal{N}(\mu_P, \sigma_P^2)$ be the target distribution, with $\sigma_Q^2 \neq \sigma_P^2$. The probability density function of $w(x) = p(x)/q(x)$ is given by:*

$$f_w(y) = \begin{cases} \frac{\overline{\sigma}}{y\sqrt{\pi \log \frac{A}{y}}} \exp\left(-\frac{1}{2}\overline{\mu}^2\right) \left(\frac{y}{A}\right)^{\overline{\sigma}^2} \cosh\left(\overline{\mu}\overline{\sigma}\sqrt{2\log\frac{A}{y}}\right), & \text{if } \sigma_Q^2 > \sigma_P^2,\ y \in [0, A], \\ \frac{\overline{\sigma}}{y\sqrt{\pi \log \frac{y}{A}}} \exp\left(-\frac{1}{2}\overline{\mu}^2\right) \left(\frac{A}{y}\right)^{\overline{\sigma}^2} \cosh\left(\overline{\mu}\overline{\sigma}\sqrt{2\log\frac{y}{A}}\right), & \text{if } \sigma_Q^2 < \sigma_P^2,\ y \in [A, \infty), \end{cases}$$

where $\overline{\mu} = \frac{\sigma_Q}{\sigma_Q^2 - \sigma_P^2}(\mu_P - \mu_Q)$ and $\overline{\sigma}^2 = \frac{\sigma_P^2}{\left|\sigma_Q^2 - \sigma_P^2\right|}$.

Proof We look at $w(x)$ as a function of random variable $x \sim Q$. We introduce the following symbols:

$$m = \frac{\sigma_Q^2 \mu_P - \sigma_P^2 \mu_Q}{\sigma_Q^2 - \sigma_P^2}, \quad \tau = \frac{\sigma_Q^2 - \sigma_P^2}{\sigma_Q^2 \sigma_P^2}.$$

Let us start computing the c.d.f.:

$$\begin{aligned} F_w(y) &= \Pr\left(w(x) \leqslant y\right) \\ &= \Pr\left(A \exp\left(-\frac{1}{2}\tau(x-m)^2 \right) \leqslant y \right) = \Pr\left(\tau(x-m)^2 \geqslant -2\log\frac{y}{A} \right). \end{aligned}$$

We distinguish the two cases according to the sign of τ and we observe that $x = \mu_Q + \sigma_Q z$ where $z \sim \mathcal{N}(0,1)$. $\boldsymbol{\tau > 0}$*:*

$$\begin{aligned} F_w(y) &= \Pr\left((x-m)^2 \geqslant \frac{2}{\tau}\log\frac{A}{y} \right) \\ &= \Pr\left(x \leqslant m - \sqrt{\frac{2}{\tau}\log\frac{A}{y}} \right) + \Pr\left(x \geqslant m + \sqrt{\frac{2}{\tau}\log\frac{A}{y}} \right) \end{aligned}$$

$$= \Pr\left(z \leqslant \frac{m-\mu_Q}{\sigma_Q} - \sqrt{\frac{2}{\tau\sigma_Q^2}\log\frac{A}{y}}\right) + \Pr\left(z \geqslant \frac{m-\mu_Q}{\sigma_Q} + \sqrt{\frac{2}{\tau\sigma_Q^2}\log\frac{A}{y}}\right).$$

We call $\overline{\mu} = \frac{m-\mu_Q}{\sigma_Q} = \frac{\sigma_Q}{\sigma_Q^2-\sigma_P^2}(\mu_P-\mu_Q)$ and $\overline{\sigma}^2 = \frac{1}{\tau\sigma_Q^2} = \frac{\sigma_P^2}{\sigma_Q^2-\sigma_P^2}$, thus we have:

$$\begin{aligned} F_w(y) &= \Pr\left(z \leqslant \overline{\mu} - \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right) + \Pr\left(z \geqslant \overline{\mu} + \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right) \\ &= \Phi\left(\overline{\mu} - \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right) + 1 - \Phi\left(\overline{\mu} + \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right), \end{aligned}$$

where Φ is the c.d.f. of a normal standard distribution. By taking the derivative w.r.t. y we get the p.d.f.:

$$\begin{aligned} f_w(y) &= \frac{\partial F_w(y)}{\partial y} \\ &= -\sqrt{2\overline{\sigma}^2}\frac{1}{2\sqrt{\log\frac{A}{y}}}\frac{y}{A}\frac{-A}{y^2}\left(\phi\left(\overline{\mu} - \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right) + \phi\left(\overline{\mu} + \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right)\right) \\ &= \frac{\sqrt{2}\overline{\sigma}}{2y\sqrt{\log\frac{A}{y}}}\left(\phi\left(\overline{\mu} - \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right) + \phi\left(\overline{\mu} + \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right)\right) \\ &= \frac{\sqrt{2}\overline{\sigma}}{2y\sqrt{\log\frac{A}{y}}}\frac{1}{\sqrt{2\pi}}\left(\exp\left(-\frac{1}{2}\left(\overline{\mu} - \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right)^2\right)\right. \\ &\quad \left. + \exp\left(-\frac{1}{2}\left(\overline{\mu} + \sqrt{2\overline{\sigma}^2\log\frac{A}{y}}\right)^2\right)\right) \\ &= \frac{\overline{\sigma}}{y\sqrt{\pi\log\frac{A}{y}}}\exp\left(-\frac{1}{2}\overline{\mu}^2\right)\exp\left(-\overline{\sigma}^2\log\frac{A}{y}\right) \\ &\quad \times \frac{\exp\left(\overline{\mu}\overline{\sigma}\sqrt{2\log\frac{A}{y}}\right) + \exp\left(-\overline{\mu}\overline{\sigma}\sqrt{2\log\frac{A}{y}}\right)}{2} \\ &= \frac{\overline{\sigma}}{y\sqrt{\pi\log\frac{A}{y}}}\exp\left(-\frac{1}{2}\overline{\mu}^2\right)\left(\frac{y}{A}\right)^{\overline{\sigma}^2}\cosh\left(\overline{\mu}\overline{\sigma}\sqrt{2\log\frac{A}{y}}\right), \end{aligned}$$

where ϕ is the p.d.f. of a normal standard distribution. $\boldsymbol{\tau < 0}$*:* The derivation takes similar steps, all it takes is to call $\overline{\sigma}^2 = -\frac{1}{\tau\sigma_Q^2} = \frac{\sigma_P^2}{\sigma_P^2-\sigma_Q^2}$, then the c.d.f. becomes:

$$F_w(y) = \Phi\left(\overline{\mu} + \sqrt{2\overline{\sigma}^2\log\frac{y}{A}}\right) - \Phi\left(\overline{\mu} - \sqrt{2\overline{\sigma}^2\log\frac{y}{A}}\right),$$

and the p.d.f. is:

$$f_w(x) = \frac{\overline{\sigma}}{y\sqrt{\pi \log \frac{y}{A}}} \exp\left(-\frac{1}{2}\overline{\mu}^2\right)\left(\frac{A}{y}\right)^{\overline{\sigma}^2} \cosh\left(\overline{\mu\sigma}\sqrt{2\log\frac{y}{A}}\right).$$

To unify the two cases we set $\overline{\sigma}^2 = \frac{\sigma_P^2}{\left|\sigma_Q^2 - \sigma_P^2\right|}$. ■

It is interesting to investigate the properties of the tail of the distribution when w is unbounded. Indeed, we discover that the distribution displays a fat–tail behavior.

Proposition 4 *If $\sigma_P^2 > \sigma_Q^2$ then there exists $c > 0$ and $y_0 > 0$ such that for any $y \geqslant y_0$, the p.d.f. f_w can be lower bounded as $f_w(y) \geqslant c y^{-1-\overline{\sigma}^2} (\log y)^{-\frac{1}{2}}$.*

Proof Let us call $z = y/A$ and let $a > 0$ be a constant, then it holds that for sufficiently large y we have:

$$f_w(y) \geqslant a z^{-1-\overline{\sigma}^2} (\log z)^{-1/2} \exp\left(\sqrt{\log z}\right)^{\sqrt{2}\overline{\mu\sigma}}. \tag{P.22}$$

To get the result, we observe that for $z > 1$ we have $\exp\left(\sqrt{\log z}\right) \geqslant 1$. Now, by replacing z with y/A we just need to change the constant a into $c > 0$. ■

As a consequence, the α–th moment of $w(x)$ does not exist for $\alpha - 1 - \overline{\sigma}^2 \geqslant -1 \implies \alpha \geqslant \overline{\sigma}^2 = \frac{\sigma_P^2}{\sigma_P^2 - \sigma_Q^2}$, this prevents from using Bernstein–like inequalities for bounding in probability the importance weights. The non–existence of finite moments is confirmed by the α–Rényi divergence. Indeed, the α–Rényi divergence is defined when $\sigma_\alpha^2 = \alpha\sigma_Q^2 + (1-\alpha)\sigma_P^2 > 0$, i.e., $\alpha < \frac{\sigma_P^2}{\sigma_P^2 - \sigma_Q^2}$.

C.2. Equal variances

If $\sigma_Q^2 = \sigma_P^2 = \sigma^2$, the importance weights have the following expression:

$$w(x) = \exp\left(\frac{\mu_P - \mu_Q}{\sigma^2}\left(x - \frac{\mu_P + \mu_Q}{2}\right)\right), \tag{41}$$

for $x \sim Q$. The weight $w(x)$ is clearly unbounded and has 0 as infimum value. Let us investigate its probability density function.

Proposition 5 *Let $Q \sim \mathcal{N}(\mu_Q, \sigma^2)$ be the behavioral distribution and $P \sim \mathcal{N}(\mu_P, \sigma^2)$ be the target distribution. The probability density function of $w(x) = q(x)/p(x)$ is given by:*

$$f_w(y) = \frac{|\widetilde{\sigma}|}{\sqrt{2\pi} y^{\frac{3}{2}}} \exp\left(-\frac{1}{2}\left(\widetilde{\mu}^2 + \widetilde{\sigma}^2 (\log y)^2\right)\right), \tag{42}$$

where $\widetilde{\mu} = \frac{\mu_P - \mu_Q}{2\sigma}$ and $\widetilde{\sigma} = \frac{\sigma}{\mu_P - \mu_Q}$.

Proof We start computing the c.d.f.:

$$\begin{aligned} F_w(y) &= \Pr\left(\exp\left\{\frac{\mu_P - \mu_Q}{\sigma^2}\left(x - \frac{\mu_P + \mu_Q}{2}\right)\right\} \leqslant y\right) \\ &= \Pr\left(\frac{\mu_P - \mu_Q}{\sigma^2}\left(x - \frac{\mu_P + \mu_Q}{2}\right) \leqslant \log y\right). \end{aligned}$$

First, we consider the case $\mu_P - \mu_Q > 0$ and observe that $x = \mu_Q + \sigma z$, where $z \sim \mathcal{N}(0,1)$:

$$F_w(y) = \Pr\left(x \leqslant \frac{\mu_P + \mu_Q}{2} + \frac{\sigma^2}{\mu_P - \mu_Q}\log y\right) = \Pr\left(z \leqslant \frac{\mu_P - \mu_Q}{2\sigma} + \frac{\sigma}{\mu_P - \mu_Q}\log y\right).$$

We call $\widetilde{\mu} = \frac{\mu_P - \mu_Q}{2\sigma}$ and $\widetilde{\sigma} = \frac{\sigma}{\mu_P - \mu_Q}$ and we have:

$$F_w(y) = \Pr\left(z \leqslant \widetilde{\mu} + \widetilde{\sigma}\log y\right) = \Phi\left(\widetilde{\mu} + \widetilde{\sigma}\log y\right).$$

We take the derivative in order to get the density function:

$$\begin{aligned} f_w(y) = \frac{\partial F_w(y)}{\partial y} &= \frac{\widetilde{\sigma}}{y}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\widetilde{\mu} + \widetilde{\sigma}\log y\right)^2\right) \\ &= \frac{\widetilde{\sigma}}{\sqrt{2\pi}y^{\widetilde{\mu}\widetilde{\sigma}+1}}\exp\left(-\frac{1}{2}\left(\widetilde{\mu}^2 + \widetilde{\sigma}^2\left(\log y\right)^2\right)\right). \end{aligned}$$

For the case $\mu_P - \mu_Q < 0$ the derivation is symmetric and the p.d.f. differs only by a minus sign. We account for this fact by considering $|\widetilde{\sigma}|$ in the final formula. ■

In the case of equal variances, the tail behavior is different.

Proposition 6 *If $\sigma_P^2 = \sigma_Q^2$ then for any $\alpha > 0$ there exist $c > 0$ and $y_0 > 0$ such that for any $y \geqslant y_0$, the p.d.f. can be upper bounded as $f_w(y) \leqslant c y^{-\alpha}$.*

Proof Condensing all the constants in c, the p.d.f. can be written as:

$$f_w(y) = c y^{-3/2}\exp\left(\left(\log y\right)^2\right)^{-\frac{\widetilde{\sigma}^2}{2}}. \tag{P.23}$$

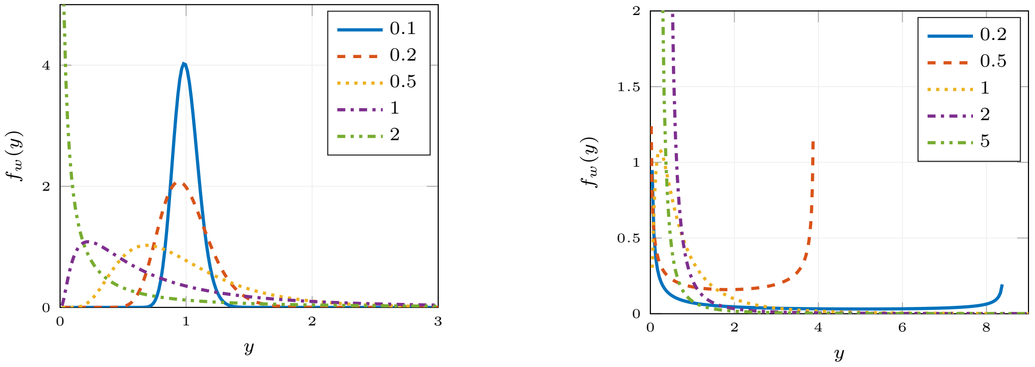
For any $\alpha > 0$, let us solve the following inequality:

$$y^{3/2}\exp\left(\left(\log y\right)^2\right)^{\frac{\widetilde{\sigma}^2}{2}} \geqslant y^{\alpha} \quad \implies \quad y \geqslant \exp\left(\frac{2}{\widetilde{\sigma}^2}\left(\alpha - \frac{3}{2}\right)\right). \tag{P.24}$$

Thus, for $y \geqslant \exp\left(\frac{2}{\widetilde{\sigma}^2}\left(\alpha - \frac{3}{2}\right)\right)$ we have that $f_w(y) \leqslant c y^{-\alpha}$. ■

This is sufficient to ensure the existence of the moments of any order, indeed the corresponding Rényi divergence is: $\frac{\alpha(\mu_P - \mu_Q)^2}{2\sigma^2}$. By the way, the distribution of $w(x)$ remains subexponential, as $\exp\left(\left(\log y\right)^2\right)^{-\frac{\widetilde{\sigma}^2}{2}} \geqslant e^{-\eta y}$ for sufficiently large y.

Figure 8 reports the p.d.f. of the importance weights for different values of mean and variance of the target distribution.



(a) equal variance

(b) equal mean

Figure 8: Probability density function of the importance weights when the behavioral distribution is $\mathcal{N}(0,1)$ and the mean is changed keeping the variance equal to 1 (a) or the variance is changed keeping the target mean equal to 1 (b).

Appendix D. Analysis of the SN Estimator

In this appendix, we provide some results regarding bias and variance of the self–normalized importance sampling estimator. Let us start with the following result, derived from (Cortes et al., 2010), that bounds the expected squared difference between non–self–normalized weight $w(x)$ and self–normalized weight $\widetilde{w}(x)$.

Lemma 4 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathcal{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ i.i.d. random variables sampled from Q. Then, for $N > 0$ and for any $i = 1, 2, \dots, N$ it holds that:*

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\widetilde{w}_{P/Q}(x_i) - \frac{w_{P/Q}(x_i)}{N}\right)^2\right] \leqslant \frac{d_2(P\|Q) - 1}{N}. \tag{43}$$

Proof The result derives from simple algebraic manipulations and from the fact that $\mathbb{V}\mathrm{ar}_{x\sim Q}\left[w_{P/Q}(x)\right] = d_2(P\|Q) - 1$.

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\widetilde{w}_{P/Q}(x_i) - \frac{w_{P/Q}(x_i)}{N}\right)^2\right] &= \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\frac{w_{P/Q}(x_i)}{\sum_{j=1}^N w_{P/Q}(x_j)}\right)^2\left(1 - \frac{\sum_{j=1}^N w_{P/Q}(x_j)}{N}\right)^2\right] \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(1 - \frac{\sum_{j=1}^N w_{P/Q}(x_j)}{N}\right)^2\right] = \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\frac{\sum_{j=1}^N w_{P/Q}(x_j)}{N}\right] \\
&= \frac{1}{N}\mathop{\mathbb{V}\mathrm{ar}}_{x_1\sim Q}\left[w_{P/Q}(x_1)\right] = \frac{d_2(P\|Q) - 1}{N}.
\end{aligned}$$

■

A similar argument can be used to derive a bound on the bias of the SN estimator.

Proposition 7 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathcal{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ i.i.d. random variables sampled from Q and $f : \mathcal{X} \to \mathbb{R}$ be a bounded function ($\|f\|_\infty < \infty$). Then, the bias of the SN estimator can be bounded as:*

$$\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \right] \right| \leqslant \|f\|_\infty \min \left\{ 2, \sqrt{\frac{d_2(P\|Q)-1}{N}} \right\}. \tag{44}$$

Proof Since it holds that $|\widetilde{\mu}_{P/Q}| \leqslant \|f\|_\infty$ the bias cannot be larger than $2\|f\|_\infty$. We now derive a bound for the bias that vanishes as $N \to \infty$. We exploit the fact that the IS estimator is unbiased, i.e., $\mathbb{E}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right] = \mathbb{E}_{x\sim P}\left[f(x)\right]$.

$$\begin{aligned}
\left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \right] \right| &= \left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right] \right] \right| = \left| \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\widetilde{\mu}_{P/Q} - \widehat{\mu}_{P/Q} \right] \right| \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\left| \widetilde{\mu}_{P/Q} - \widehat{\mu}_{P/Q} \right| \right] = \\
&= \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\left| \frac{\sum_{i=1}^N w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)} - \frac{\sum_{i=1}^N w_{P/Q}(x_i) f(x_i)}{N} \right| \right] \\
&= \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\left| \frac{\sum_{i=1}^N w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)} \right| \left| 1 - \frac{\sum_{i=1}^N w_{P/Q}(x_i)}{N} \right| \right] &\text{(P.25)} \\
&\leqslant \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\left(\frac{\sum_{i=1}^N w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)} \right)^2 \right]^{\frac{1}{2}} \mathop{\mathbb{E}}_{\mathbf{x}\sim Q} \left[\left(1 - \frac{\sum_{i=1}^N w_{P/Q}(x_i)}{N} \right)^2 \right]^{\frac{1}{2}} &\text{(P.26)} \\
&\leqslant \|f\|_\infty \sqrt{\frac{d_2(P\|Q)-1}{N}}, &\text{(P.27)}
\end{aligned}$$

where (P.26) follows from (P.25) by applying Cauchy–Schwartz inequality and (P.27) is obtained by observing that $\left(\frac{\sum_{i=1}^N w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)} \right)^2 \leqslant \|f\|_\infty^2$. ■

Bounding the variance of the SN estimator is non–trivial since the the normalization term makes all the samples interdependent. Exploiting the boundedness of $\widetilde{\mu}_{P/Q}$ we can derive trivial bounds like: $\mathbb{V}\mathrm{ar}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right] \leqslant \|f\|_\infty^2$. However, this bound does not shrink with the number of samples N. Several approximations of the variance have been proposed, like the following derived using the delta method (Ver Hoef, 2012; Owen, 2013):

$$\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right] = \frac{1}{N} \mathop{\mathbb{E}}_{x_1\sim Q} \left[w_{P/Q}^2(x_1) \left(f(x_1) - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \right)^2 \right] + o(N^{-2}). \tag{45}$$

We will not use the approximate expression for the variance, but we will directly bound the Mean Squared Error (MSE) of the SN estimator, which is the sum of the variance and the bias squared.

Proposition 8 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathcal{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ i.i.d. random variables sampled* *from Q and $f : \mathcal{X} \rightarrow \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, the* MSE *of the SN estimator can be bounded as:*

$$\mathrm{MSE}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right] \leqslant 2\|f\|_\infty^2 \min\left\{2, \frac{2d_2(P\|Q)-1}{N}\right\}. \tag{46}$$

Proof First, recall that $\widetilde{\mu}_{P/Q}$ is bounded by $\|f\|_\infty$ thus its MSE cannot be larger than $4\|f\|_\infty^2$. The idea of the proof is to sum and subtract the IS estimator $\widehat{\mu}_{P/Q}$:

$$\begin{aligned}
\mathrm{MSE}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right] &= \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right]\right)^2\right] \\
&= \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \pm \widehat{\mu}_{P/Q}\right)^2\right] && \text{(P.28)} \\
&\leqslant 2\mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\widetilde{\mu}_{P/Q} - \widehat{\mu}_{P/Q}\right)^2\right] + 2\mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\widehat{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}[f(x)]\right)^2\right] && \text{(P.29)} \\
&\leqslant 2\mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(\frac{\sum_{i=1}^N w_{P/Q}(x_i)f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)}\right)^2\left(1 - \frac{\sum_{i=1}^N w_{P/Q}(x_i)}{N}\right)^2\right] + 2\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right] && \text{(P.30)} \\
&\leqslant 2\|f\|_\infty^2 \mathop{\mathbb{E}}_{\mathbf{x}\sim Q}\left[\left(1 - \frac{\sum_{i=1}^N w_{P/Q}(x_i)}{N}\right)^2\right] + 2\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right] && \text{(P.31)} \\
&\leqslant 2\|f\|_\infty^2 \mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\frac{\sum_{i=1}^N w_{P/Q}(x_i)}{N}\right] + 2\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\widehat{\mu}_{P/Q}\right] \\
&\leqslant 2\|f\|_\infty^2 \frac{d_2(P\|Q)-1}{N} + 2\|f\|_\infty^2\frac{d_2(P\|Q)}{N} = 2\|f\|_\infty^2 \frac{2d_2(P\|Q)-1}{N},
\end{aligned}$$

where line (P.29) follows from line (P.28) by applying the inequality $(a+b)^2 \leqslant 2(a^2+b^2)$, (P.31) follows from (P.30) by observing that $\left(\frac{\sum_{i=1}^N w_{P/Q}(x_i)f(x_i)}{\sum_{i=1}^N w_{P/Q}(x_i)}\right)^2 \leqslant \|f\|_\infty^2$. ■

We can use this result to provide a high confidence bound for the SN estimator.

Proposition 9 *Let P and Q be two probability measures on the measurable space $(\mathcal{X}, \mathcal{F})$ such that $P \ll Q$ and $d_2(P\|Q) < +\infty$. Let $x_1, x_2, \dots, x_N$ i.i.d. random variables sampled from Q and $f : \mathcal{X} \rightarrow \mathbb{R}$ be a bounded function ($\|f\|_\infty < +\infty$). Then, for any $0 < \delta \leqslant 1$ and $N > 0$ with probability at least $1-\delta$:*

$$\mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \geqslant \frac{1}{N}\sum_{i=1}^N \widetilde{w}_{P/Q}(x_i)f(x_i) - 2\|f\|_\infty \min\left\{1, \sqrt{\frac{d_2(P\|Q)(4-3\delta)}{\delta N}}\right\}.$$

Proof The result is obtained by applying Cantelli's inequality and accounting for the bias. Consider the random variable $\widetilde{\mu}_{P/Q} = \frac{1}{N}\sum_{i=1}^N \widetilde{w}_{P/Q}(x_i)f(x_i)$ and let $\widetilde{\lambda} = \lambda - \left|\mathbb{E}_{x\sim P}\left[f(x)\right] - \mathbb{E}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right|$:

$$\Pr\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \geqslant \lambda\right) = \Pr\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right] \geqslant \lambda + \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right)$$

$$\begin{aligned}
&\leqslant \Pr\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right] \geqslant \lambda - \left|\mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right|\right)\\
&= \Pr\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right] \geqslant \widetilde{\lambda}\right).
\end{aligned}$$

Now we apply Cantelli's inequality:

$$\begin{aligned}
\Pr\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \geqslant \lambda\right) \leqslant \Pr\left(\widetilde{\mu}_{P/Q} - \mathop{\mathbb{E}}_{x\sim P}\left[\widetilde{\mu}_{P/Q}\right] \geqslant \widetilde{\lambda}\right) &\leqslant \frac{1}{1+\frac{\widetilde{\lambda}^2}{\mathbb{V}\mathrm{ar}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right]}}\\
&= \frac{1}{1+\frac{\left(\lambda - \left|\mathbb{E}_{x\sim P}\left[f(x)\right] - \mathbb{E}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right|\right)^2}{\mathbb{V}\mathrm{ar}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right]}}. \qquad \text{(P.32)}
\end{aligned}$$

By calling $\delta = \frac{1}{1+\frac{\left(\lambda - \left|\mathbb{E}_{x\sim P}\left[f(x)\right] - \mathbb{E}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right|\right)^2}{\mathbb{V}\mathrm{ar}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right]}}$ and considering the complementary event, we get that with probability at least $1-\delta$ we have:

$$\mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] \geqslant \widetilde{\mu}_{P/Q} - \left|\mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] - \mathop{\mathbb{E}}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right| - \sqrt{\frac{1-\delta}{\delta}\mathop{\mathbb{V}\mathrm{ar}}_{\mathbf{x}\sim Q}\left[\widetilde{\mu}_{P/Q}\right]}. \qquad \text{(P.33)}$$

Then we bound the bias term $\left|\mathbb{E}_{x\sim P}\left[f(x)\right] - \mathbb{E}_{\mathbf{x}\sim P}\left[\widetilde{\mu}_{P/Q}\right]\right|$ with Equation (44) and the variance term with the MSE in Equation (46). With some simple algebraic manipulations we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim P}\left[f(x)\right] &\geqslant \widetilde{\mu}_{P/Q} - \|f\|_\infty\sqrt{\frac{d_2(P\|Q)-1}{N}} - \|f\|_\infty\sqrt{\frac{1-\delta}{\delta}\frac{2(2d_2(P\|Q)-1)}{N}}\\
&\geqslant \widetilde{\mu}_{P/Q} - \|f\|_\infty\sqrt{\frac{d_2(P\|Q)}{N}} - \|f\|_\infty\sqrt{\frac{1-\delta}{\delta}\frac{4d_2(P\|Q)}{N}}\\
&= \widetilde{\mu}_{P/Q} - \|f\|_\infty\sqrt{\frac{d_2(P\|Q)}{N}}\left(1+2\sqrt{\frac{1-\delta}{\delta}}\right)\\
&\geqslant \widetilde{\mu}_{P/Q} - 2\|f\|_\infty\sqrt{\frac{d_2(P\|Q)}{N}}\sqrt{1+\frac{4(1-\delta)}{\delta}}\\
&\geqslant \widetilde{\mu}_{P/Q} - 2\|f\|_\infty\sqrt{\frac{d_2(P\|Q)(4-3\delta)}{\delta N}},
\end{aligned}$$

where the last line follows from the fact that $\sqrt{a}+\sqrt{b} \leqslant 2\sqrt{a+b}$ for any $a, b \geqslant 0$. Finally, recalling that the range of the SN estimator is $2\|f\|_\infty$ we get the result. ■

It is worth noting that, apart for the constants, the bound has the same dependence on d_2 as in Theorem 2. Thus, by suitably redefining the hyperparameter λ we can optimize the same surrogate objective function for both IS and SN estimators.

Appendix E. Estimation of the Rényi divergence

In this appendix, we provide the derivations related to the results presented in Remark 6. Whenever possible, we will provide derivations for a generic α–Rényi divergence.

The first estimator is obtained by simply rephrasing the definition at Equation (4) into a sample–based version:

$$\widehat{d}_{\alpha}\left(P\|Q\right) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{p(x_i)}{q(x_i)}\right)^{\alpha} = \frac{1}{N}\sum_{i=1}^{N} w_{P/Q}^{\alpha}(x_i), \tag{47}$$

where $x_i \sim Q$. This estimator is clearly unbiased and applies to any pair of probability distributions. We now upper bound its variance when $P = p(\cdot|\boldsymbol{\theta}')$ and $Q = p(\cdot|\boldsymbol{\theta})$.

Proposition 10 *The variance of the Rényi Divergence* $\widehat{d}_{\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)$ *can be upper bounded as:*

$$\underset{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\widehat{d}_{\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)\right] \leqslant \frac{1}{N} d_{2\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)^{2\alpha-1}$$

Proof

$$\begin{aligned}
\underset{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\widehat{d}_{\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)\right] &= \frac{1}{N}\underset{\tau\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\left(\frac{p(\tau|\boldsymbol{\theta}')}{p(\tau|\boldsymbol{\theta})}\right)^{\alpha}\right] \\
&\leqslant \frac{1}{N}\underset{\tau\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{E}}\left[\left(\frac{p(\tau|\boldsymbol{\theta}')}{p(\tau|\boldsymbol{\theta})}\right)^{2\alpha}\right] \\
&= \frac{1}{N} d_{2\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)^{2\alpha-1}.
\end{aligned}$$

■

It follows from Proposition 1 that $d_{2\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) \leqslant \sup_{s\in\mathcal{S}}\left\{d_{2\alpha}\left(\pi_{\boldsymbol{\theta}'}(\cdot|s)\|\pi_{\boldsymbol{\theta}}(\cdot|s)\right)\right\}^{H}$ and, consequently:

$$\underset{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\widehat{d}_{\alpha}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)\right] \leqslant \frac{1}{N}\sup_{s\in\mathcal{S}}\left\{d_{2\alpha}\left(\pi_{\boldsymbol{\theta}'}(\cdot|s)\|\pi_{\boldsymbol{\theta}}(\cdot|s)\right\}^{2\alpha-1}\right)^{H} = \mathcal{O}\left(\frac{1}{N}\mathfrak{D}_4^{H}\right).$$

Concerning the estimator at Equation (31), we can derive the following bound.

Proposition 11 *The variance of the Rényi Divergence* $\check{d}_{2}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)$ *can be upper bounded as:*

$$\begin{aligned}
\underset{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\check{d}_{2}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)\right] \leqslant \frac{1}{N}\Bigg(& d_4\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)^3 - 4d_3\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)^2 \\
& + 6d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) - 3\Bigg).
\end{aligned}$$

Proof

$$\begin{aligned}
\underset{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\check{d}_{2}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)\right] &= \underset{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[1 + \frac{1}{N}\sum_{i=1}^{N}\left(w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) - 1\right)^2\right] \\
&= \frac{1}{N}\underset{\tau\sim p(\cdot|\boldsymbol{\theta})}{\mathbb{V}\mathrm{ar}}\left[\left(w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) - 1\right)^2\right]
\end{aligned}$$

$$\leqslant \frac{1}{N} \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[\left(w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau) - 1 \right)^4 \right]$$
$$= \frac{1}{N} \Big(d_4 \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right)^3 - 4 d_3 \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right)^2 + 6 d_2 \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right) - 3 \Big).$$

■

We can further simplify the expression by applying Proposition 1, getting the order $\mathcal{O}\left(\frac{1}{N}\mathfrak{D}_4^H\right)$.

Finally, we consider the estimator at Equation (32):

$$\widetilde{d}_\alpha \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right) = \frac{1}{N} \sum_{i=1}^{N} \prod_{t=0}^{H-1} d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s_{\tau_i,t}) \| \pi_{\boldsymbol{\theta}}(\cdot|s_{\tau_i,t}) \right). \tag{48}$$

This estimator is biased, however it enjoys a better Mean Squared Error bound.

Proposition 12 *The MSE of the Rényi Divergence* $\widetilde{d}_\alpha \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right)$ *can be upper bounded as:*

$$\mathrm{MSE}_{\boldsymbol{\tau} \sim p(\cdot|\boldsymbol{\theta})} \left[\widetilde{d}_\alpha \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right) \right] \leqslant \left(1 + \frac{1}{N} \right) \sup_{s \in \mathcal{S}} \left\{ d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s) \right) \right\}^{2H}. \tag{49}$$

Proof First of all, we decompose the MSE in bias and variance. Concerning the bias, we know that it cannot be larger than the maximum difference between the true value $d_\alpha \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right)$ and the estimate $\widetilde{d}_\alpha \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right)$, i.e., $\sup_{s \in \mathcal{S}} \left\{ d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s) \right) \right\}^H$. Concerning the variance, we have:

$$\begin{aligned}
\mathop{\mathbb{V}\mathrm{ar}}_{\boldsymbol{\tau} \sim p(\cdot|\boldsymbol{\theta})} \left[\widetilde{d}_\alpha \left(p(\cdot|\boldsymbol{\theta}') \| p(\cdot|\boldsymbol{\theta}) \right) \right] &= \frac{1}{N} \mathop{\mathbb{V}\mathrm{ar}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[\prod_{t=0}^{H-1} d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s_t) \| \pi_{\boldsymbol{\theta}}(\cdot|s_t) \right) \right] \\
&\leqslant \frac{1}{N} \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[\prod_{t=0}^{H-1} d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s_t) \| \pi_{\boldsymbol{\theta}}(\cdot|s_t) \right)^2 \right] \\
&\leqslant \frac{1}{N} \sup_{s \in \mathcal{S}} \left\{ d_\alpha \left(\pi_{\boldsymbol{\theta}'}(\cdot|s) \| \pi_{\boldsymbol{\theta}}(\cdot|s) \right) \right\}^{2H}.
\end{aligned}$$

By summing the variance and the bias squared, we get the result. ■

Finally, it is worth noting that $d_\alpha(P\|Q)^2 \leqslant d_{2\alpha}(P\|Q)^{\frac{2\alpha-1}{\alpha-1}}$. Indeed, from Jensen inequality:

$$d_\alpha(P\|Q)^2 = \left(\mathop{\mathbb{E}}_{x \sim Q} \left[\left(\frac{p(x)}{q(x)} \right)^\alpha \right] \right)^{\frac{2}{\alpha-1}} \leqslant \mathop{\mathbb{E}}_{x \sim Q} \left[\left(\frac{p(x)}{q(x)} \right)^{2\alpha} \right]^{\frac{1}{\alpha-1} \cdot \frac{2\alpha-1}{2\alpha-1}} = d_{2\alpha}(P\|Q)^{\frac{2\alpha-1}{\alpha-1}}. \tag{50}$$

Appendix F. Asymptotic analysis for the Variance of A-POIS and D-POIS

In this appendix, we report the proofs of the asymptotic analysis of the variance for A-POIS and D-POIS. We will refer to Equation (14) for A-POIS and Equation (36) for P-POIS.

If $\mathfrak{D}_2 = 1$, we have that all $d_2(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})) = 1$ and thus, for A-POIS:

$$\mathop{\mathbb{V}\mathrm{ar}}_{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\text{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] \leqslant \frac{R_{\max}^2}{N}\begin{cases}\left(\frac{1-\gamma^H}{1-\gamma}\right)^2 & \text{if } \gamma < 1\\ H^2 & \text{if } \gamma = 1\end{cases} \leqslant \frac{R_{\max}^2}{N}\begin{cases}\frac{1}{(1-\gamma)^2} & \text{if } \gamma < 1\\ H^2 & \text{if } \gamma = 1\end{cases}.$$

Analogously, for D-POIS we have:

$$\begin{aligned}\mathop{\mathbb{V}\mathrm{ar}}_{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\text{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] &\leqslant \frac{R_{\max}^2}{N}\sum_{t=0}^{H-1} c_t\\ &= \frac{R_{\max}^2}{N}\begin{cases}\left(\frac{1-\gamma^H}{1-\gamma}\right) & \text{if } \gamma < 1\\ H^2 & \text{if } \gamma = 1\end{cases}\\ &\leqslant \frac{R_{\max}^2}{N}\begin{cases}\frac{1}{(1-\gamma)^2} & \text{if } \gamma < 1\\ H^2 & \text{if } \gamma = 1\end{cases}.\end{aligned}$$

We now focus on A-POIS and consider the case $\mathfrak{D}_2 > 1$. In such case, we have from Proposition 1 that $d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) \leqslant \mathfrak{D}_2^H$. Consequently:

$$\mathop{\mathbb{V}\mathrm{ar}}_{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\text{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] \leqslant \frac{R_{\max}^2}{N}\mathfrak{D}_2^H\begin{cases}\left(\frac{1-\gamma^H}{1-\gamma}\right)^2 & \text{if } \gamma < 1\\ H^2 & \text{if } \gamma = 1\end{cases} \leqslant \frac{R_{\max}^2}{N}\mathfrak{D}_2^H\begin{cases}\frac{1}{(1-\gamma)^2} & \text{if } \gamma < 1\\ H^2 & \text{if } \gamma = 1\end{cases}. \tag{51}$$

Let us now consider D-POIS. For $\gamma = 1$, we proceed as follows, recalling the inequality $d_2\left(p(\cdot|\boldsymbol{\theta}', t)\|p(\cdot|\boldsymbol{\theta}, t)\right) \leqslant \mathfrak{D}_2^t$:

$$\begin{aligned}\mathop{\mathbb{V}\mathrm{ar}}_{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\text{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] &\leqslant \frac{R_{\max}^2}{N}\sum_{t=0}^{H-1}(2H-2t-1)d_2\left(p(\cdot|\boldsymbol{\theta}', t)\|p(\cdot|\boldsymbol{\theta}, t)\right)\\ &\leqslant \frac{R_{\max}^2}{N}\sum_{t=0}^{H-1}(2H-2t-1)\mathfrak{D}_2^t\\ &= \frac{R_{\max}^2}{N}\frac{\mathfrak{D}_2^{H+1}+\mathfrak{D}_2^H - 1 - 2H(\mathfrak{D}_2-1)-\mathfrak{D}_2}{(\mathfrak{D}_2-1)^2} \leqslant \frac{R_{\max}^2}{N}\mathfrak{D}_2^H\frac{\mathfrak{D}_2+1}{(\mathfrak{D}_2-1)^2}.\end{aligned}$$

In the case $\gamma < 1$, we first derive a general expression and then we particularize it for specific ranges of γ:

$$\begin{aligned}\mathop{\mathbb{V}\mathrm{ar}}_{\boldsymbol{\tau}\sim p(\cdot|\boldsymbol{\theta})}\left[\widehat{J}_{\mathcal{M}}^{\text{D-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta})\right] &\leqslant \frac{R_{\max}^2}{N}\sum_{t=0}^{H-1}\frac{\gamma^t(\gamma^t+\gamma^{t+1}-2\gamma^H)}{1-\gamma}d_2\left(p(\cdot|\boldsymbol{\theta}', t)\|p(\cdot|\boldsymbol{\theta}, t)\right)\\ &\leqslant \frac{R_{\max}^2}{N}\sum_{t=0}^{H-1}\frac{\gamma^t(\gamma^t+\gamma^{t+1}-2\gamma^H)}{1-\gamma}\mathfrak{D}_2^t.\end{aligned}$$

By solving the summation using the properties of the geometric sum and omitting the term $\frac{R_{\max}^2}{N}$ for conciseness, we get to the result:

$$\frac{1+\gamma-\gamma^{2H+2}\mathfrak{D}_2^{H+1}+\gamma^{2H}\mathfrak{D}_2^H+(\mathfrak{D}_2-1)\gamma^{2H+1}\mathfrak{D}_2^H-\gamma^2\mathfrak{D}_2-\gamma\mathfrak{D}_2+2\gamma^{H+2}\mathfrak{D}_2-2\gamma^H}{(1-\gamma)(\gamma\mathfrak{D}_2-1)\left(\gamma^2\mathfrak{D}_2-1\right)} \tag{52}$$

For $\mathfrak{D}_2 > \frac{1}{\gamma^2}$, we have that the leading term in the bound for $H \to \infty$ is given by:

$$\frac{\gamma^{2H+2}\mathfrak{D}_2^{H+1} + \gamma^{2H}\mathfrak{D}_2^{H} + (\mathfrak{D}_2 - 1)\gamma^{2H+1}\mathfrak{D}_2^{H}}{(1-\gamma)(\gamma\mathfrak{D}_2 - 1)\left(\gamma^2\mathfrak{D}_2 - 1\right)} = \frac{(\gamma^2\mathfrak{D}_2)^H\left(\gamma\mathfrak{D}_2 + 1\right)}{(\gamma\mathfrak{D}_2 - 1)\left(\gamma^2\mathfrak{D}_2 - 1\right)}.$$

Instead, for $\mathfrak{D}_2 < \frac{1}{\gamma^2}$, all terms at the numerator go to zero as $H \to \infty$ except the following ones:

$$\frac{1 + \gamma - \gamma^2\mathfrak{D}_2 - \gamma\mathfrak{D}_2}{(1-\gamma)(\gamma\mathfrak{D}_2 - 1)\left(\gamma^2\mathfrak{D}_2 - 1\right)} = \frac{1+\gamma}{(1-\gamma\mathfrak{D}_2)\left(1-\gamma^2\mathfrak{D}_2\right)}. \tag{53}$$

The case $\mathfrak{D}_2 = \frac{1}{\gamma^2}$ needs to be treated separately, leading to the result:

$$\sum_{t=0}^{H-1} \frac{\gamma^t(\gamma^t + \gamma^{t+1} - 2\gamma^H)}{\gamma^{2t}(1-\gamma)} = \frac{\left(1-\gamma^2\right)H - 2\gamma\left(1-\gamma^H\right)}{(1-\gamma)^2} \leqslant \frac{1+\gamma}{1-\gamma}H.$$

Appendix G. Implementation details

In this appendix, we provide some aspects about our implementation of POIS.

G.1. Line Search

At each offline iteration k the parameter update is performed in the direction defined by $\mathcal{G}(\boldsymbol{\theta}_k^h)^{-1}\nabla_{\boldsymbol{\theta}_h^j}\mathcal{L}(\boldsymbol{\theta}_k^j/\boldsymbol{\theta}_{1:J}^h)$ with a step size α_k determined in order to maximize the improvement. In this update rule, $\mathcal{G}$ is a positive definite matrix that define the Riemann manifold of interest. $\mathcal{G}$ is the identity matrix in the vanilla gradient and the FIM in the case of natural gradient. For brevity, we will remove subscripts and the dependence on $\boldsymbol{\theta}_{1:J}^h$ from the involved quantities. The rationale behind our line search is the following. Suppose that our objective function $\mathcal{L}(\boldsymbol{\theta})$, restricted to the gradient direction $\mathcal{G}^{-1}(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$, represents a concave parabola in the Riemann manifold having $\mathcal{G}(\boldsymbol{\theta})$ as Riemann metric tensor. Suppose we know a point $\boldsymbol{\theta}_0$, the Riemann gradient in that point $\mathcal{G}(\boldsymbol{\theta}_0)^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)$ and another point: $\boldsymbol{\theta}_l = \boldsymbol{\theta}_0 + \alpha_l\mathcal{G}(\boldsymbol{\theta}_0)^{-1}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)$. For both points we know the value of the loss function: $\mathcal{L}_0 = \mathcal{L}(\boldsymbol{\theta}_0)$ and $\mathcal{L}_l = \mathcal{L}(\boldsymbol{\theta}_l)$, and we indicate with $\Delta\mathcal{L}_l = \mathcal{L}_l - \mathcal{L}_0$ the objective function improvement. Having this information, we can compute the vertex of that parabola, which is its global maximum. Let us call $l(\alpha) = \mathcal{L}\left(\boldsymbol{\theta}_0 + \alpha\mathcal{G}^{-1}(\boldsymbol{\theta}_0)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\right) - \mathcal{L}(\boldsymbol{\theta}_0)$. Being a parabola it can be expressed as $l(\alpha) = a\alpha^2 + b\alpha + c$. Clearly, $c = 0$ by definition of $l(\alpha)$; a and b can be determined by enforcing the conditions:

$$\begin{aligned} b = \left.\frac{\partial l}{\partial \alpha}\right|_{\alpha=0} &= \frac{\partial}{\partial \alpha}\mathcal{L}\left(\boldsymbol{\theta}_0 + \alpha\mathcal{G}^{-1}(\boldsymbol{\theta}_0)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\right) - \mathcal{L}(\boldsymbol{\theta}_0)|_{\alpha=0} = \\ &= \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)^T\mathcal{G}^{-1}(\boldsymbol{\theta}_0)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0) = \\ &= \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}, \end{aligned}$$

$$\begin{aligned} l(\alpha_l) = a\alpha_l^2 + b\alpha_l = a\alpha_l^2 + \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l = \Delta\mathcal{L}_l \quad &\Longrightarrow \\ \Longrightarrow \quad a = \frac{\Delta\mathcal{L}_l - \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l}{\alpha_l^2}. \end{aligned}$$

Therefore, the parabola has the form:

$$l(\alpha) = \frac{\Delta\mathcal{L}_l - \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l}{\alpha_l^2}\alpha^2 + \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha. \tag{54}$$

Clearly, the parabola is concave only if $\Delta\mathcal{L}_l < \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l$. The vertex is located at:

$$\alpha_{l+1} = \frac{\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l^2}{2\left(\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l - \Delta\mathcal{L}_l\right)}. \tag{55}$$

To simplify the expression, like in (Matsubara et al., 2010) we define the quantity $\alpha_l = \epsilon_l / \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}$. Thus, we get:

$$\epsilon_{l+1} = \frac{\epsilon_l^2}{2(\epsilon_l - \Delta\mathcal{L}_l)}. \tag{56}$$

Of course, we need also to manage the case in which the parabola is convex, i.e., $\Delta\mathcal{L}_l \geqslant \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l$. Since our objective function is not really a parabola we reinterpret the two cases: i) $\Delta\mathcal{L}_l > \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l$, the function is sublinear and in this case we use Equation (56) to determine the new step size $\alpha_{l+1} = \epsilon_{l+1} / \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}$; ii) $\Delta\mathcal{L}_l \geqslant \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}\alpha_l$, the function is superlinear, in this case we increase the step size multiplying it by $\eta > 1$, i.e., $\alpha_{l+1} = \eta\alpha_l$. Finally, the update rule becomes:

$$\epsilon_{l+1} = \begin{cases} \eta\epsilon_l & \text{if } \Delta\mathcal{L}_l > \frac{\epsilon_l(2\eta-1)}{2\eta} \\ \frac{\epsilon_l^2}{2(\epsilon_l - \Delta\mathcal{L}_l)} & \text{otherwise} \end{cases}. \tag{57}$$

The procedure is iterated until a maximum number of attempts is reached (say 30) or the objective function improvement is too small (say 1e-4). The pseudocode of the line search is reported in Algorithm 3.

Algorithm 3 Parabolic Line Search

```
    Input: $\mathrm{tol}_{\Delta\mathcal{L}} = 1e-4$, $M_{\mathrm{ls}} = 30$, $\mathcal{L}_0$
    Output : $\alpha^*$
  $\alpha_0 = 0$
  $\epsilon_1 = 1$
  $\Delta\mathcal{L}_{k-1} = -\infty$
  for $l = 1, 2, \dots, M_{\mathrm{ls}}$ do
      $\alpha_l = \epsilon_l / \|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)\|^2_{\mathcal{G}^{-1}(\boldsymbol{\theta}_0)}$
      $\boldsymbol{\theta}_l = \alpha_l \mathcal{G}^{-1}(\boldsymbol{\theta}_0)\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_0)$
      $\Delta\mathcal{L}_l = \mathcal{L}_l - \mathcal{L}_0$
      if $\Delta\mathcal{L}_l < \Delta\mathcal{L}_{l-1} + \mathrm{tol}_{\Delta\mathcal{L}}$ then
          return $\alpha_{l-1}$
      end if
      $\epsilon_{l+1} = \begin{cases} \eta\epsilon_l & \text{if } \Delta\mathcal{L}_l > \frac{\epsilon_l(1-2\eta)}{2\eta} \\ \frac{\epsilon_l^2}{2(\epsilon_l - \Delta\mathcal{L}_l)} & \text{otherwise} \end{cases}$
  end for
```

G.2. Practical surrogate objective functions

In practice, the Rényi divergence term d_2 in the surrogate objective functions presented so far, either exact in P-POIS or approximate in A-POIS, tends to be overly-conservative. To mitigate this problem, by observing that $d_2(P\|Q)/N = 1/\mathrm{ESS}(P\|Q)$ from Equation (7) we can replace the whole quantity with an estimator like $\widehat{\mathrm{ESS}}(P\|Q)$, as presented in Equation (7). This leads to the following approximated surrogate objective functions:

$$\widetilde{\mathcal{L}}_{\lambda}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} w_{\boldsymbol{\theta}'/\boldsymbol{\theta}}(\tau_i)R(\tau_i) - \frac{\lambda}{\sqrt{\widehat{\mathrm{ESS}}\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)}},$$

$$\widetilde{\mathcal{L}}_{\lambda}^{\mathrm{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho}) = \frac{1}{N}\sum_{i=1}^{N} w_{\boldsymbol{\rho}'/\boldsymbol{\rho}}(\boldsymbol{\theta}_i)R(\tau_i) - \frac{\lambda}{\sqrt{\widehat{\mathrm{ESS}}\left(\nu_{\boldsymbol{\rho}'}\|\nu_{\boldsymbol{\rho}}\right)}}.$$

Moreover, in all the experiments, we use the empirical maximum reward in place of the true $R_{\max}$.

G.3. Practical P-POIS for Deep Neural Policies (N-POIS)

As mentioned in Section 7.2, P-POIS applied to deep neural policies suffers from a curse of dimensionality due to the high number of (scalar) parameters (which are $\sim 10^3$ for the network used in the experiments). The corresponding hyperpolicy is a multi–variate Gaussian (diagonal covariance) with a very high dimensionality. As a result, the Rényi divergence, used as a penalty, is extremely sensitive even to small perturbations, causing an overly–conservative behavior. First, we give up the exact Rényi computation and use the practical surrogate objective function $\widetilde{\mathcal{L}}_{\lambda}^{\mathrm{P-POIS}}$ proposed in Appendix G.2. This, however, is not enough. The importance weights, being the products of thousands of probability densities, can easily become zero, preventing any learning. Hence, we decide to group the policy parameters in smaller blocks, and independently learn the corresponding hyperparameters. In general, we can define a family of M orthogonal policy–parameter subspaces $\{\Theta_m \leqslant \Theta\}_{m=1}^{M}$, where $V \leqslant W$ reads "V is a subspace of W". For each Θ_m, we consider a multi–variate diagonal–covariance Gaussian with Θ_m as support, obtaining a corresponding hyperparameter subspace $\mathcal{P}_m \leqslant \mathcal{P}$. Then, for each $\mathcal{P}_m$, we compute a separate surrogate objective (where we employ self–normalized importance weights):

$$\widetilde{\mathcal{L}}_{\lambda}^{\mathrm{N-POIS}}(\boldsymbol{\rho}'_m/\boldsymbol{\rho}_m) = \frac{1}{N}\sum_{i=1}^{N} \widetilde{w}_{\boldsymbol{\rho}'_m/\boldsymbol{\rho}_m}(\boldsymbol{\theta}_m^i) R(\tau_i) - \frac{\lambda}{\sqrt{\widehat{\mathrm{ESS}}\left(\nu_{\boldsymbol{\rho}'_m} \| \nu_{\boldsymbol{\rho}_m}\right)}},$$

where $\boldsymbol{\rho}_m, \boldsymbol{\rho}'_m \in \mathcal{P}_m, \boldsymbol{\theta}_m \in \Theta_m$. Each objective is independently optimized via natural gradient ascent, where the step size is found via a line search as usual. It remains to define a meaningful grouping for the policy parameters, i.e., for the weights of the deep neural policy. We choose to group them by network unit, or neuron (counting output units but not input units). More precisely, let denote a network unit as a function:

$$U_i(\mathbf{x}|\boldsymbol{\theta}_m) = g(\mathbf{x}^T\boldsymbol{\theta}_m),$$

where $\mathbf{x}$ is the vector of the inputs to the unit (including a 1 that multiplies the bias parameter) and $g(\cdot)$ is an activation function. To each unit U_m we associate a block Θ_m such that $\boldsymbol{\theta}_m \in \Theta_m$. In more connectivist–friendly terms, we group connections by the neuron they go into. For the network we used in the experiments, this reduces the order of the multivariate Gaussian hyperpolicies from $\sim 10^3$ to $\sim 10^2$. We call this practical variant of our algorithm Neuron–Based POIS (N-POIS). Although some design choices seem rather arbitrary, and independently optimizing hyperparameter blocks clearly neglects some potentially meaningful interactions, the practical results of N-POIS are promising, as reported in Section 7.2. Figure 9 is an ablation study showing the performance of P-POIS variants on Cartpole. Only using both the tricks discussed in this section, we are able to solve the task (this experiment is on 50 iterations only).

Appendix H. Experiments Details

In this Appendix, we report the hyperparameter values used in the experimental evaluation and some additional plots and experiments. We adopted different criteria to decide the batch size: for linear policies at each iteration 100 episodes are collected regardless of their length, whereas for deep neural policies, in order to be fully comparable with (Duan et al., 2016), 50000 timesteps are collected at each iteration regardless of the resulting number of episodes (the last episode is cut so that the number of timesteps sums up exactly to 50000). Clearly, this difference is relevant only for episodic tasks.

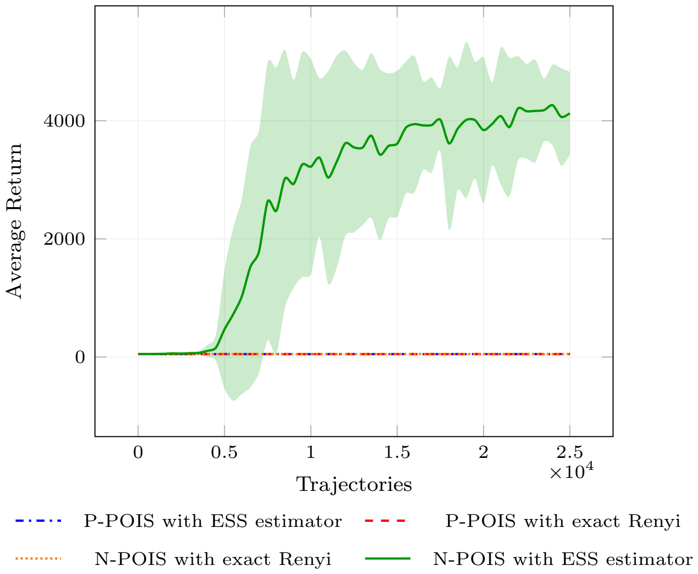


Figure 9: Ablation study for N-POIS (5 runs, 95% c.i.).

H.1. Linear policies

In the following we report the hyperparameters shared by all tasks and algorithms for the experiments with linear policies:

- Policy architecture: Normal distribution $\mathcal{N}(u_{\mathbf{M}}(\mathbf{s}), e^{2\mathbf{\Omega}})$, where the mean $u_{\mathbf{M}}(\mathbf{s}) = \mathbf{Ms}$ is a linear function in the state variables with no bias, and the variance is state–independent and parametrized as $e^{2\mathbf{\Omega}}$, with diagonal $\mathbf{\Omega}$.
- Number of runs: 20 (95% c.i.)
- seeds: 10, 109, 904, 160, 570, 662, 963, 100, 746, 236, 247, 689, 153, 947, 307, 42, 950, 315, 545, 178
- Policy initialization: mean parameters sampled from $\mathcal{N}(0, 0.01^2)$, variance initialized to 1
- Task horizon: 500
- Number of iterations: 500
- Maximum number of line search attempts (POIS only): 30
- Maximum number of offline iterations (POIS only): 10
- Episodes per iteration: 100
- Importance weight estimator (POIS only): IS for A-POIS and D-POIS, SN for P-POIS
- Natural gradient (POIS only): No for A-POIS and D-POIS, Yes for P-POIS

Table 9 reports the hyperparameters that have been tuned specifically for each task selecting the best combination based on the runs corresponding to the first 5 seeds.

Environment	A-POIS (δ)	P-POIS (δ)
Cart-Pole Balancing	0.1, 0.2, 0.3, **0.4**, 0.5	0.1, 0.2, 0.3, **0.4**, 0.5, 0.6, 0.7, 0.8, 0.9 1
Inverted Pendulum	0.8, **0.9**, 0.99, 1	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, **0.8**, 0.9 1
Mountain Car	0.8, **0.9**, 0.99, 1	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, **1**
Acrobot	0.1, 0.3, 0.5, **0.7**, 0.9	0.1, **0.2**, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 1
Double Inverted Pendulum	**0.1**, 0.2, 0.3, 0.4, 0.5	**0.1**, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 1

Environment	D-POIS (δ)
Cart–Pole Balancing	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, **0.99**
Inverted Pendulum	0.6, 0.8, 0.9, 0.99, **0.9999**, 1
Mountain Car	0.5, 0.7, **0.9**, 0.99
Acrobot	0.1, 0.3, 0.5, **0.7**, 0.9
Double Inverted Pendulum	0.1, 0.2, 0.3, **0.4**, 0.5

Environment	TRPO (step size)	PPO (step size)
Cart–Pole Balancing	0.001, 0.01, **0.1**, 1	0.001, **0.01**, 0.1 , 1
Inverted Pendulum	0.001, **0.01**, 0.1, 1	0.001, **0.01**, 0.1, 1
Mountain Car	0.001, **0.01**, 0.1, 1	0.001, 0.01, 0.1, **1**
Acrobot	0.001, 0.01, 0.1, **1**	0.001, 0.01, 0.1, **1**
Double Inverted Pendulum	0.001, 0.01, **0.1**, 1	0.001, 0.01, 0.1, **1**

Table 9: Task–specific hyperparameters for the experiments with linear policy. δ is the significance level for POIS while we report the step size for TRPO and PPO. In **bold**, the best hyperparameters found.

H.2. Deep neural policies

In the following we report the hyperparameters shared by all tasks and algorithms for the experiments with deep neural policies:

- Policy architecture: Normal distribution $\mathcal{N}(u_{\mathbf{M}}(\mathbf{s}), e^{2\boldsymbol{\Omega}})$, where the mean $u_{\mathbf{M}}(\mathbf{s})$ is a 3–layers MLP (100, 50, 25) with bias (activation functions: tanh for hidden–layers, linear for output layer), the variance is state–independent and parametrized as $e^{2\boldsymbol{\Omega}}$ with diagonal $\boldsymbol{\Omega}$.
- Number of runs: 5 (95% c.i.)
- seeds: 10, 109, 904, 160, 570
- Policy initialization: uniform Xavier initialization (Glorot and Bengio, 2010), variance initialized to 1
- Task horizon: 500
- Number of iterations: 500

- Maximum number of line search attempts (POIS only): 30
- Maximum number of offline iterations (POIS only): 20
- Timesteps per iteration: 50000
- Importance weight estimator (POIS only): IS for A-POIS and D-POIS, SN for P-POIS
- Natural gradient (POIS only): No for A-POIS and D-POIS, Yes for P-POIS

Table 10 reports the hyperparameters that have been tuned specifically for each task selecting the best combination based on the runs corresponding to the 5 seeds.

Environment	A-POIS (δ)	P-POIS (δ)
Cart–Pole Balancing	0.9, **0.99**, 0.999	0.4, 0.5, **0.6**, 0.7, 0.8
Mountain Car	0.9, **0.99**, 0.999	0.1, 0.2, **0.3**, 0.4, 0.5, 0.6, 0.7, 0.8
Double Inverted Pendulum	0.9, **0.99**, 0.999	0.4, 0.5, 0.6, 0.7, **0.8**
Swimmer	0.9, **0.99**, 0.999	0.4, 0.5, **0.6**, 0.7, 0.8

Environment	D-POIS (δ)
Cart–Pole Balancing	0.9, **0.99**, 0.999
Mountain Car	0.9, **0.99**, 0.999
Double Inverted Pendulum	0.2, **0.4**, 0.6, 0.8, 0.9
Swimmer	0.9, **0.99**, 0.999

Table 10: Task–specific hyperparameters for the experiments with deep neural policies. δ is the significance level for POIS. In **bold**, the best hyperparameters found.

H.3. MIS Experiments

The setting used for the MIS experiments presented in Section 7.3 is the same as Section H.1, but with variable batch size N and MIS capacity J. The hyper–parameter values reported in Table 7 are obtained via grid search among candidate values: $\delta = 0.99, 0.9, 0.8, 0.6, 0.4, 0.2, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001$ for all the settings. The hyper–parameter tuning was performed on five random seeds (10, 109, 904, 160, 570), and the final results reported in Figure 7 were averaged over twenty separate ones (749, 728, 524, 215, 455, 920, 635, 930, 402, 705, 938, 563, 925, 29, 173, 542, 899, 175, 152, 210).

H.4. Additional Results about Figure 10

In Figure 10 we report additional plots w.r.t. Figure 5 for A-POIS when changing the δ parameter in the Cartpole environment. It is worth noting that the value of δ has also an effect on the speed with which the variance of the policy approaches zero. Indeed, smaller policy variances induce a larger Rényi divergence and thus with a higher penalization (small δ) reducing the policy variance is discouraged. Moreover, we can see the values of the bound before and after the optimization. Clearly, the higher the value of δ, the higher the value of the bound after the optimization process, as the penalization term is weaker. It is interesting to notice that when $\delta = 1$ the bound after the optimization reaches values that are impossible to reach for any policy and this is a consequence of the high uncertainty in the importance sampling estimator.

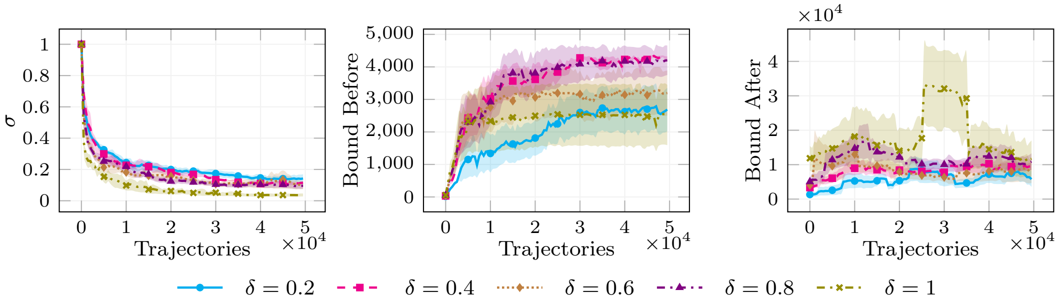


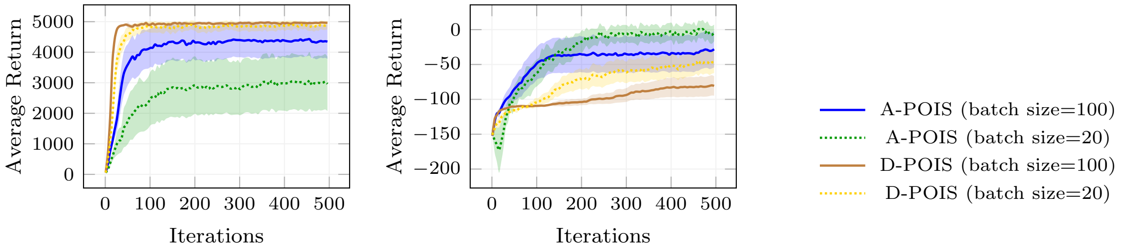
Figure 10: Standard Deviation of the policy (σ), value of the bound before and after the optimization as a function of the number of trajectories for A-POIS in the Cartpole environment for different values of δ (5 runs, 95% c.i.).

H.5. D-POIS Variance

In this section, we present empirical results to validate the properties described in Section 5.3. To underline the difference in the variance of A-POIS and D-POIS, we try to force a variance increase by lowering the batch size of episodes at each iteration, making the variance difference more noticeable. We first tested this idea on the *Cartpole* environment (Figure 11a), which has a quite trivial reward structure, assigning the same reward at each timestep. We can notice how A-POIS struggles more when the batch size is reduced, not even reaching the optimal solution, while D-POIS still reaches the top in a very short time. We also need to remind that by reducing the batch size but keeping the same number of iterations, we are effectively using less total samples (in the example, one fifth); this suggest that, in many cases, D-POIS can be more sample efficient than A-POIS. We can also observe how the variability in the performances, i.e., the confidence bound in the plot, is much larger in the A-POIS setting, which indicates that the optimization is less restricted and this results in a more explorative behavior.

This particular fact is what most probably influences the performances in the *Inverted–Pendulum* environment, shown in Figure 11b: we can see how D-POIS struggles to escape a sub–optimal solution, and how reducing the batch size (which increases the variance), actually improves the performance. The better results of A-POIS may similarly suggest that the higher variance helps in escaping the sub–optimal solution, even more with the reduced batch size.

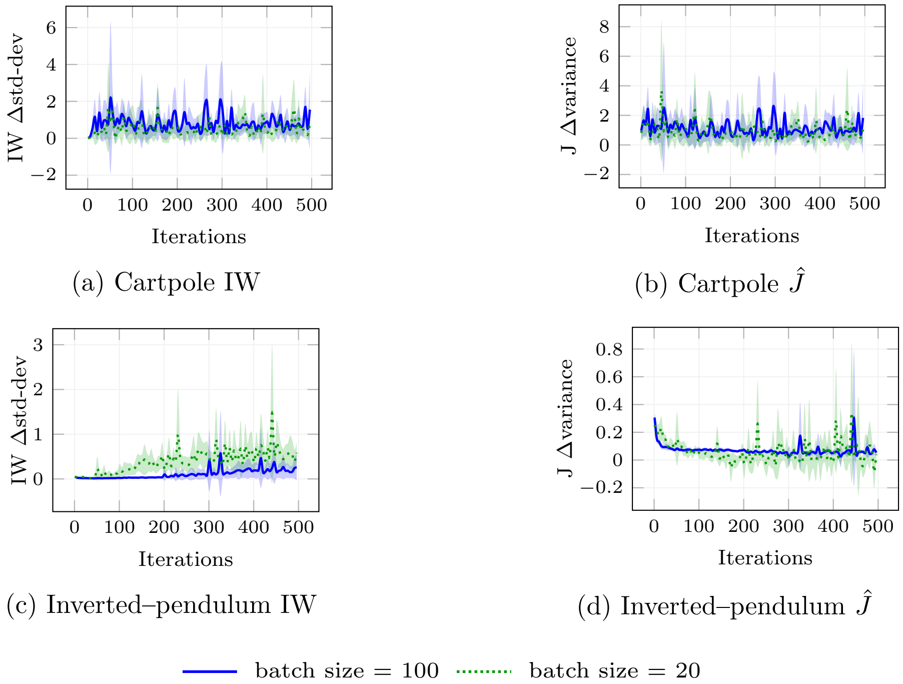
To further inspect the variance in the weights and in the estimator, we also measure directly their sample variance during the experiment. To have comparable measures in A-POIS and D-POIS, we need to align them, i.e., we perform the updates following D-POIS but also estimate the weights and the $\hat{J}$ of the A-POIS setting. In Figure 12 we show the relative increase in standard deviation of A-POIS w.r.t. D-POIS, defined as $\frac{\sigma_{\text{APOIS}} - \sigma_{\text{DPOIS}}}{\sigma_{\text{DPOIS}}}$.



(a) Cartpole

(b) Inverted Pendulum

Figure 11: Performance comparison of A-POIS and D-POIS, changing the batch size to increase the variance in the estimator.



(a) Cartpole IW

(b) Cartpole $\hat{J}$

(c) Inverted–pendulum IW

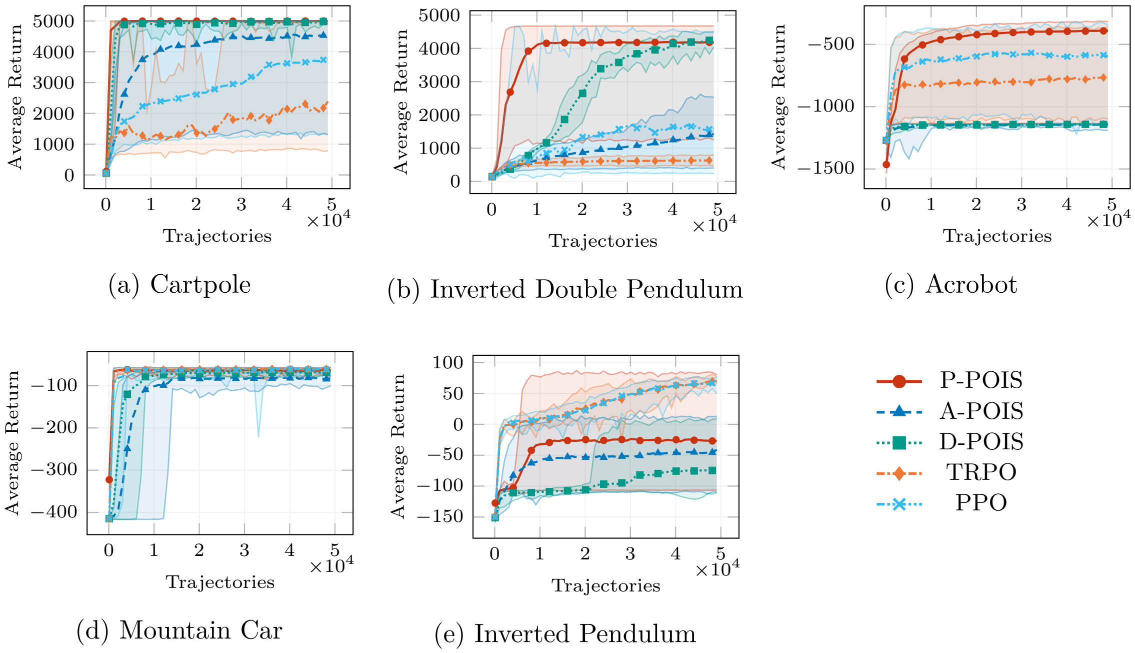
(d) Inverted–pendulum $\hat{J}$

Figure 12: Relative increase in importance–weights and estimator standard deviation under the per–decision setting.

The always positive value of this measure shows how the variance of both the weights and the estimator is lower in the per–decision setting, accordingly to the previous theoretical analysis and experimental results, in both the presented environments.

H.6. Tolerance Intervals of Linear Policies and MIS Experiments

In this appendix, we report tolerance intervals for the experiments with linear policies (Figure 13) and with MIS (Figure 14). The problem with confidence intervals is that they enclose a deterministic quantity (the mean of the population), only accounting for the



(a) Cartpole

(b) Inverted Double Pendulum

(c) Acrobot

(d) Mountain Car

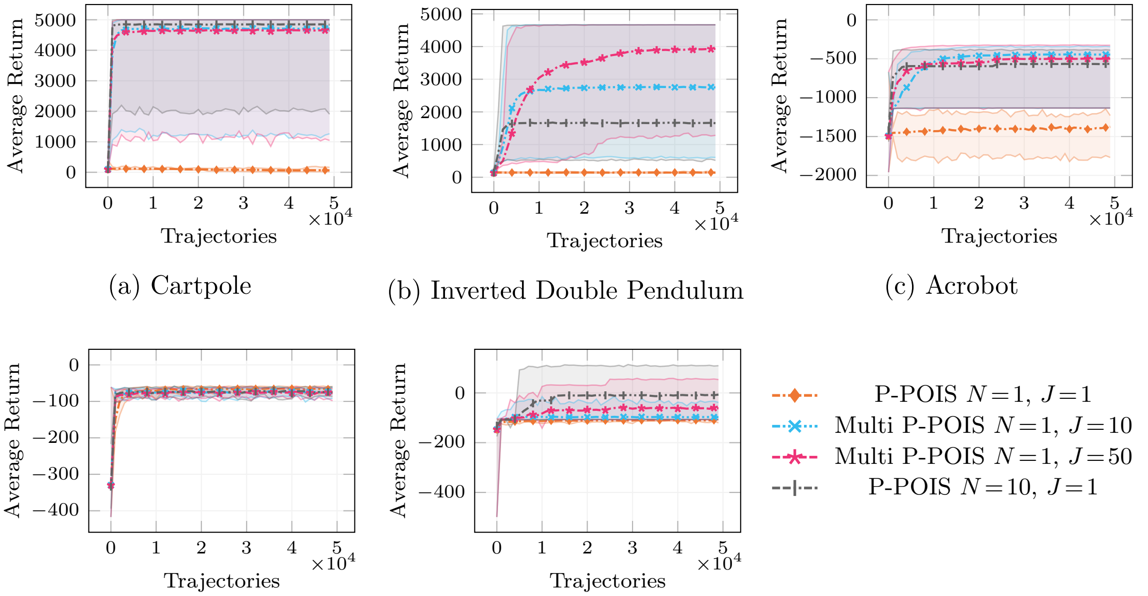
(e) Inverted Pendulum

Figure 13: Average return as a function of the number of trajectories for P-POIS, A-POIS, D-POIS, TRPO, and PPO with *linear policy* (20 runs, mean and 90%/80% non–parametric tolerance intervals).

sampling error due to the finite number of samples. If the performance of the algorithm varies drastically with the random seed (e.g., if the algorithm either performs very well or very bad), the mean–performance curve may be of little relevance, if not misleading, and so is the related confidence region. On the contrary, *tolerance intervals* (Hahn and Meeker, 2011) enclose, with the desired confidence, a specific portion of the population. We employ the non–parametric Hahn–Meeker method to compute the intervals (Hahn and Meeker, 2011). A $100(1-\alpha)\%/100p\%$ interval is one that includes a proportion p of the population with probability at least $1-\alpha$.

H.7. Individual Runs of Deep Neural Policies Experiments

In this appendix, we report for the experiments with deep neural policies the learning curves of the individual runs (Figure 15). This allows to fully appreciate the variability of the algorithms' performance w.r.t. the random seed.



(a) Cartpole

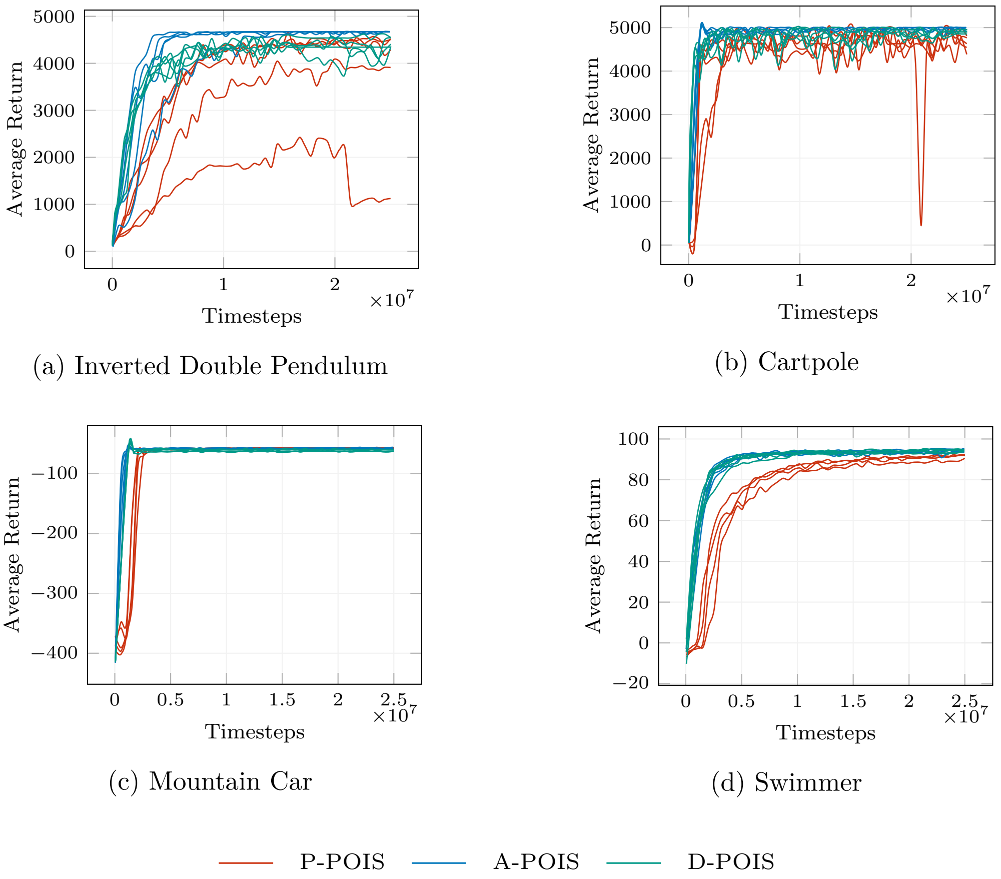
(b) Inverted Double Pendulum

(c) Acrobot

(d) Mountain Car

(e) Inverted Pendulum

Figure 14: Average return as a function of the number of trajectories for P-POIS with *linear policy* for different values of the batch size N and the MIS capacity J (20 runs, mean and 90%/80% non–parametric tolerance intervals).



(a) Inverted Double Pendulum

(b) Cartpole

(c) Mountain Car

(d) Swimmer

Figure 15: Average return of the *individual runs* as a function of the number of trajectories for A-POIS, P-POIS with deep neural policies.

References

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. *CoRR*, abs/1908.00261, 2019.

Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 151–160. PMLR, 2019.

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

Shun-ichi Amari. *Differential-geometrical methods in statistics*, volume 28. Springer Science & Business Media, 2012.

Shun-ichi Amari and Andrzej Cichocki. Information geometry of divergence functions. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 58(1):183–195, 2010.

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

Bernard Bercu, Bernard Delyon, and Emmanuel Rio. Concentration inequalities for sums. In *Concentration Inequalities for Sums and Martingales*, pages 11–60. Springer, 2015.

Jacob Burbea. The convexity with respect to gaussian distributions of divergences of order α. *Utilitas Mathematica*, 26:171–192, 1984.

FP Cantelli. Sui confini della probabilita. In *Atti del Congresso Internazionale dei Matematici: Bologna del 3 al 10 de settembre di 1928*, pages 47–60, 1929.

F Chung and L Lu. Old and new concentration inequalities. *Complex Graphs and Networks*, 107:23–56, 2006.

Kamil Ciosek and Shimon Whiteson. Expected policy gradients. In *AAAI*, pages 2868–2875. AAAI Press, 2018.

William G Cochran. *Sampling techniques*. John Wiley & Sons, 2007.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450, 2010.

Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

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

Shayan Doroudi, Philip S Thomas, and Emma Brunskill. Importance sampling for fair policy selection. Uncertainty in Artificial Intelligence, 2017.

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

Rasool Fakoor, Pratik Chaudhari, and Alexander J. Smola. P3O: policy-on policy-off policy optimization. *CoRR*, abs/1905.01756, 2019.

Carles Gelada and Marc G. Bellemare. Off-policy deep reinforcement learning by bootstrapping the covariate shift. In *AAAI*, pages 3647–3655. AAAI Press, 2019.

Manuel Gil, Fady Alajaji, and Tamas Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

Mandy Grüttner, Frank Sehnke, Tom Schaul, and Jürgen Schmidhuber. Multi-dimensional deep memory go-player for parameter exploring policy gradients. 2010.

Zhaohan Guo, Philip S Thomas, and Emma Brunskill. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pages 2489–2498, 2017.

Gerald J Hahn and William Q Meeker. *Statistical intervals: a guide for practitioners*, volume 92. John Wiley & Sons, 2011.

Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1372–1383. PMLR, 2017.

Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Jean Harb, Tom Schaul, Doina Precup, and Pierre-Luc Bacon. Policy evaluation networks. *CoRR*, abs/2002.11833, 2020.

Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, S. M. Ali Eslami, Martin A. Riedmiller, and David Silver. Emergence of locomotion behaviours in rich environments. *CoRR*, abs/1707.02286, 2017.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *AAAI*, pages 3207–3214. AAAI Press, 2018.

Timothy Classen Hesterberg. *Advances in importance sampling.* PhD thesis, Stanford University, 1988.

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

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, volume 2, pages 267–274, 2002.

Sham M Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002.

Michael J. Kearns and Satinder P. Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348, 1992.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *NeurIPS*, pages 5361–5371, 2018.

Yao Liu, Pierre-Luc Bacon, and Emma Brunskill. Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. *arXiv preprint arXiv:1910.06508*, 2019a.

Yao Liu, Adith Swaminathan, Alekh Agarwal, and Emma Brunskill. Off-policy policy gradient with state distribution correction. *CoRR*, abs/1904.08473, 2019b.

Luca Martino, Víctor Elvira, and Francisco Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401, 2017.

Takamitsu Matsubara, Tetsuro Morimura, and Jun Morimoto. Adaptive step-size policy gradients with average reward metric. In *Proceedings of 2nd Asian Conference on Machine Learning*, pages 285–298, 2010.

Alberto Maria Metelli, Matteo Papini, Francesco Faccio, and Marcello Restelli. Policy optimization via importance sampling. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 5447–5459, 2018.

Atsushi Miyamae, Yuichi Nagata, Isao Ono, and Shigenobu Kobayashi. Natural policy gradient methods with parameter-based exploration for control tasks. In *Advances in neural information processing systems*, pages 1660–1668, 2010.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Andrew Y Ng and Michael Jordan. Pegasus: A policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 406–415. Morgan Kaufmann Publishers Inc., 2000.

OpenAI. Openai five. `https://blog.openai.com/openai-five/`, 2018.

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *CoRR*, abs/1808.00177, 2018.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019.

Art B. Owen. *Monte Carlo theory, methods and examples.* 2013.

Joni Pajarinen, Hong Linh Thai, Riad Akrour, Jan Peters, and Gerhard Neumann. Compatible natural gradient policy search. *Mach. Learn.*, 108(8-9):1443–1466, 2019.

Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4026–4035, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Matteo Papini, Alberto Maria Metelli, Lorenzo Lupo, and Marcello Restelli. Optimistic policy optimization via multiple importance sampling. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 4989–4999. PMLR, 2019.

Matteo Papini, Andrea Battistello, and Marcello Restelli. Balancing learning speed and stability in policy gradient via adaptive exploration. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pages 1188–1199. PMLR, 2020.

Jing Peng and Ronald J Williams. Incremental multi-step q-learning. In *Machine Learning Proceedings 1994*, pages 226–232. Elsevier, 1994.

Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750. ACM, 2007.

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008a.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008b.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In *AAAI*, pages 1607–1612. Atlanta, 2010.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *International Conference on Machine Learning*, pages 307–315, 2013.

Doina Precup, Richard S Sutton, and Satinder P Singh. Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning*, pages 759–766. Citeseer, 2000.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Aravind Rajeswaran, Kendall Lowrey, Emanuel V Todorov, and Sham M Kakade. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems*, pages 6553–6564, 2017.

C Radhakrishna Rao. Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in statistics*, pages 235–247. Springer, 1992.

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

Mark Rowland, Anna Harutyunyan, Hado van Hasselt, Diana Borsa, Tom Schaul, Rémi Munos, and Will Dabney. Conditional importance sampling for off-policy learning. In *AISTATS*, volume 108 of *Proceedings of Machine Learning Research*, pages 45–55. PMLR, 2020.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Frank Sehnke, Christian Osendorfer, Thomas Rückstieß, Alex Graves, Jan Peters, and Jürgen Schmidhuber. Policy gradients with parameter-based exploration for control. In *International Conference on Artificial Neural Networks*, pages 387–396. Springer, 2008.

Frank Sehnke, Christian Osendorfer, Thomas Rückstieß, Alex Graves, Jan Peters, and Jürgen Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010.

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *AAAI*, pages 5668–5675. AAAI Press, 2020.

Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5729–5738, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, 2014.

Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

István Szita and András Lörincz. Learning tetris using the noisy cross-entropy method. *Neural computation*, 18(12):2936–2941, 2006.

Russ Tedrake, Teresa Weirui Zhang, and H Sebastian Seung. Stochastic policy gradient reinforcement learning on a simple 3d biped. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2849–2854. IEEE, 2004.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pages 2380–2388, 2015a.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *AAAI*, pages 3000–3006, 2015b.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard E. Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 5022–5031. PMLR, 2018.

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH*, pages 419–428. ACM, 1995.

Jay M Ver Hoef. Who invented the delta method? *The American Statistician*, 66(2):124–127, 2012.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. Natural evolution strategies. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 3381–3387. IEEE, 2008.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer, 1992.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M. Bayen, Sham M. Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *ICLR*. OpenReview.net, 2018.

Pan Xu, Felicia Gao, and Quanquan Gu. An improved convergence analysis of stochastic variance-reduced policy gradient. In *UAI*, page 191. AUAI Press, 2019a.

Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. *CoRR*, abs/1909.08610, 2019b.

Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. In *Advances in Neural Information Processing Systems*, pages 262–270, 2011.

Tingting Zhao, Hirotaka Hachiya, Voot Tangkaratt, Jun Morimoto, and Masashi Sugiyama. Efficient sample reuse in policy gradients with parameter-based exploration. *Neural computation*, 25(6):1512–1547, 2013.