Transfer of Samples in Policy Search via Multiple Importance Sampling

Andrea Tirinzoni [1] **Mattia Salvini** [1] **Marcello Restelli** [1]

Abstract

We consider the transfer of experience samples in reinforcement learning. Most of the previous works in this context focused on value-based settings, where transferring instances conveniently reduces to the transfer of (s, a, s', r) tuples. In this paper, we consider the more complex case of reusing samples in policy search methods, in which the agent is required to transfer entire trajectories between environments with different transition models. By leveraging ideas from multiple importance sampling, we propose robust gradient estimators that effectively achieve this goal, along with several techniques to reduce their variance. In the case where the transition models are known, we theoretically establish the robustness to the negative transfer for our estimators. In the case of unknown models, we propose a method to efficiently estimate them when the target task belongs to a finite set of possible tasks and when it belongs to some reproducing kernel Hilbert space. We provide empirical results to show the effectiveness of our estimators.

1. Introduction

In many real-world problems, reinforcement learning (RL) agents (Sutton & Barto, 1998) repeatedly face environments with similar, but different, dynamics. Consider, for instance, deploying a policy on a robot with different physical parameters than the ones it was trained for, or adapting from simulation to reality, or learning under non-stationarity. In all these cases, the agent could potentially exploit knowledge acquired in the past environments to speed-up the learning process of new related tasks. How this knowledge reuse can be achieved is the fundamental problem of *transfer* in RL.

Knowledge transfer has been widely studied for RL domains (Taylor & Stone, 2009; Lazaric, 2012). In this context, the agent is supposed to reuse knowledge acquired from a set of *source* tasks to accelerate the learning process of a new *target* task. Designing a transfer algorithm poses three main questions: what element should be transferred? How should the transfer be performed? Is this procedure beneficial for the target task? An answer to the last question is particularly crucial to prevent *negative transfer*, the situation in which reused knowledge harms the learning process.

Among the many kinds of knowledge that have been successfully transferred in RL, we focus on experience samples, i.e., states, actions, and rewards that the agent collects from different tasks. Most of the previous approaches in this setting focused on value-based algorithms, where the agent attempts to reuse *single* transition and reward instances from different decision processes. Taylor et al. (2008) proposed an algorithm to transfer samples so as to augment the dataset used by a model-based RL algorithm. Almost simultaneously, Lazaric et al. (2008b) designed a model-free methodology to estimate which source samples are most likely to benefit the target and used it to transfer into a batch RL algorithm. The same settings were considered by Laroche & Barlier (2017), who showed that no measure of similarity is needed to safely transfer under the restrictive assumption that tasks do not differ in their dynamics. More recently, Tirinzoni et al. (2018b) proposed a method to transfer all given samples without carrying out any explicit selection, while reweighing their contribution to the learning process proportionally to their importance for solving the target task.

One of the drawbacks of the above-mentioned approaches is that they do not easily generalize to policy search, where the agent is required to transfer long sequences of states and actions rather than only single-step information. Since policy search algorithms, despite their recent successes, typically require large batches in order to reliably estimate the expected return or its gradient, reusing samples from different tasks would be practically very useful.

In this paper, we propose a methodology to address this limitation for families of tasks differing only in their transition models. Similarly to Tirinzoni et al. (2018b), we employ importance sampling (IS) (Owen, 2013) techniques to transfer *all* source samples, while correcting the bias introduced by their different distributions. However, instead of focusing on single transitions and rewards, we show that it is pos-

[1]Politecnico di Milano, Milan, Italy. Correspondence to: Andrea Tirinzoni <andrea.tirinzoni@polimi.it>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

sible to transfer entire trajectories from different policies and environments to improve the gradient estimates of a policy search algorithm, thus reducing its sample complexity and the number of iterations required to converge. In order to accomplish this objective, we use ideas from multiple importance sampling (MIS) (Veach & Guibas, 1995) to derive gradient estimators that automatically transfer samples from several different distributions and we propose two techniques to reduce their variance. Our estimators are simple, enjoy strong theoretical guarantees, and are of general interest even outside our transfer settings. For instance, we show that they can be successfully applied to reuse samples from past policies during the learning process (a kind of intra-environment transfer). For the ideal case in which the transition models are known, we formally establish robustness to negative transfer of the proposed methods. For the more realistic case in which the transition models are unknown, we propose a methodology to estimate them which explicitly trades off between the bias and variance these models might induce on the importance weights. Finally, we empirically demonstrate the effectiveness of our estimators in three domains of increasing difficulty.

2. Preliminaries

Policy Search We define a task as a discounted Markov decision process (Puterman, 2014), $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{P}_0, \gamma \rangle$, where $\mathcal{S} \subseteq \mathbb{R}^d$ is the state space, $\mathcal{A} \subseteq \mathbb{R}^u$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{P}_0 \in \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in [0, 1]$ is a discount factor. Here $\Delta(\Omega)$ denotes the set of probability measures over a generic Ω. At each time t, the agent is in some state $\boldsymbol{s}_t$, it takes an action $\boldsymbol{a}_t$ according to some policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, it receives a reward $r_t := \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$, and it transitions to a new state $\boldsymbol{s}_{t+1}$ according to $\mathcal{P}(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$. In policy search (Peters & Schaal, 2008; Sutton et al., 2000; Deisenroth et al., 2013), a class of parameterized polices $\pi_{\boldsymbol{\theta}}$ is considered and the goal is to find the parameters maximizing the expected return,

$$\arg\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P}) := \int p(\boldsymbol{\tau}|\boldsymbol{\theta}, \mathcal{P}) \mathcal{R}(\boldsymbol{\tau}) \mathrm{d}\boldsymbol{\tau}, \qquad (1)$$

where $\boldsymbol{\tau} = (\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{s}_T)$ denotes a trajectory of T time steps and, with some abuse of notation, we set $\mathcal{R}(\boldsymbol{\tau}) := \sum_{t=0}^{T-1} \gamma^t \mathcal{R}(\boldsymbol{s}_t, \boldsymbol{a}_t)$. Common approaches for optimizing this objective function employ stochastic gradient methods. Let $g(\boldsymbol{\tau}) := \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)$, then the well-known REINFORCE algorithm (Williams, 1992) approximates the gradient of (1) using n i.i.d. trajectories from $\pi_{\boldsymbol{\theta}}$ as

$$\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P}) = \frac{1}{n} \sum_{i=1}^{n} g(\boldsymbol{\tau}_i) \mathcal{R}(\boldsymbol{\tau}_i). \qquad (2)$$

Transfer of Samples In the sample transfer problem (Taylor & Stone, 2009; Lazaric, 2012), a set of experience samples is provided from a small number of *source tasks* $\mathcal{M}_1, \ldots, \mathcal{M}_m$ and, given a new *target task*, the goal is to figure out how to reuse these instances in order to speed up the learning process. In our policy search settings, these samples are entire trajectories collected under arbitrary policies. We make the following assumption.

Assumption 1 (Task differences)**.** *Each task $\mathcal{M}_j$ is uniquely characterized by its transition kernel $\mathcal{P}_j$. State-action space, reward function, and initial state distribution are shared among tasks*[1].

Formally, our input is a dataset $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$, where each $\mathcal{D}_j = \{\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_{n_j}\}$ is a set of n_j trajectories from a fixed task-policy couple $(\boldsymbol{\theta}_j, \mathcal{P}_j)$. We assume that the policies used to collect each trajectory are known.

Multiple Importance Sampling Multiple importance sampling (MIS) (Veach & Guibas, 1995; Veach, 1997) is a very effective method for estimating the expected value of a function given samples from multiple proposal distributions. Consider a measurable space $(\mathcal{X}, \mathscr{F})$, a function $f : \mathcal{X} \to \mathbb{R}$, and $m+1$ probability measures $P, Q_1, \ldots, Q_m$ absolutely continuous w.r.t. the Lebesgue measure, with $p, q_1, \ldots, q_m$ denoting their densities. A MIS estimator for $\mu := \mathbb{E}_{x \sim P}[f(x)]$ given n_j samples from each Q_j is

$$\hat{\mu} = \sum_{j=1}^{m} \frac{1}{n_j} \sum_{i=1}^{n_j} h_j(x_{i,j}) \frac{p(x_{i,j})}{q_j(x_{i,j})} f(x_{i,j}), \qquad (3)$$

where the function h is often referred to as *heuristics* and must be a partition of unity, i.e., $\sum_j h_j(x) = 1$ for all $x \in \mathcal{X}$. It is easy to show (Veach & Guibas, 1995) that the MIS estimator is unbiased. A common and convenient choice for h is the *balance heuristics*, $h_j(x) = \frac{n_j q_j(x)}{\sum_{l=1}^{m} n_l q_l(x)}$, for which the MIS estimator reduces to an IS estimator with a mixture of proposals, $\hat{\mu} = \frac{1}{n} \sum_{j=1}^{m} \sum_{i=1}^{n_j} \frac{p(x_{i,j})}{q_\alpha(x_{i,j})} f(x_{i,j})$, where $n = \sum_{j=1}^{m} n_j$ and $q_\alpha(x) = \sum_{j=1}^{m} \alpha_j q_j(x)$, with $\alpha_j = \frac{n_j}{n}$. This estimator is also known in the literature as deterministic mixture sampling or stratification (Hesterberg, 1995; Owen & Zhou, 2000). A key property is that, when a set of n_0 samples is available from the target distribution P, the resulting weights are *bounded* by $\frac{1}{\alpha_0}$. This fact also implies that the variance of these weights is bounded, a property that rarely holds for plain IS. For these reasons, samples from P are often referred to as *defensive*. An important measure of the goodness of an IS estimator is the *effective sample size* (ESS), which is typically approximated as $\widehat{\mathrm{ESS}} := \frac{n}{1+\mathbb{V}\mathrm{ar}[p(x)/q(x)]}$ (Liu, 1996) or, equivalently, as $\widehat{\mathrm{ESS}} = \frac{n}{d_2(P||Q)}$ (Cortes et al., 2010; Ryu, 2016; Metelli et al., 2018), where $d_2(P\|Q) = \int p(x)^2 q(x)^{-1} \mathrm{d}x$ is the exponentiated second-order Renyi divergence.

[1] In practice, it is enough that the target reward function is known rather than shared. The extension to unknown rewards is simple (see, e.g., (Tirinzoni et al., 2018b)).

3. Transfer via Importance Sampling

We describe how IS techniques can be used to efficiently transfer samples in policy search in case transition models are known. We will deal with unknown models in Section 4.

3.1. Multiple Importance Sampling Estimators

Consider the transfer settings described in Section 2. Our idea is to learn the target task using standard gradient-based techniques, collecting a batch of n_0 episodes at each step, while reusing the source trajectories to augment the dataset for estimating the gradient in order to reduce its variance. Since, due to different policies and transition models, these trajectories follow different distributions than the one induced by the current policy in the target task, IS techniques can be straightforwardly employed to yield an unbiased estimator,

$$\widehat{\nabla}^{\text{IS}}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P}) = \frac{1}{n} \sum_{j=0}^{m} \sum_{i=1}^{n_j} w_j^{\text{IS}}(\boldsymbol{\tau}_{i,j}) g(\boldsymbol{\tau}_{i,j}) \mathcal{R}(\boldsymbol{\tau}_{i,j}), \quad (4)$$

where the *importance weight* $w_j^{\text{IS}}(\boldsymbol{\tau}) := \frac{p(\boldsymbol{\tau}|\boldsymbol{\theta}, \mathcal{P})}{p(\boldsymbol{\tau}|\boldsymbol{\theta}_j, \mathcal{P}_j)}$ can be computed in closed-form as

$$w_j^{\text{IS}}(\boldsymbol{\tau}) = \prod_{t=0}^{T-1} \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\boldsymbol{\theta}_j}(\boldsymbol{a}_t|\boldsymbol{s}_t)} \frac{\mathcal{P}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)}{\mathcal{P}_j(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)}. \quad (5)$$

Unfortunately, this IS scheme is likely to fail in most cases of practical interest. It is well known, especially from the literature on off-policy estimation (Precup, 2000; Hachiya et al., 2009; Thomas & Brunskill, 2016; Guo et al., 2017; Liu et al., 2018), that importance sampling on long trajectories is likely to give almost zero or huge weights, thus leading to estimators with very high (sometimes infinite) variance (Li et al., 2015; Jiang & Li, 2016). This drawback is even amplified in our transfer settings, where there is a model mismatch in addition to the one between the policies. Several variance reduction techniques, typically paying a small amount of bias, have been proposed (e.g., self-normalized estimators (Kong, 1992), truncation (Ionides, 2008), flattening (Hachiya et al., 2009)). Fortunately, MIS comes to the rescue in our settings, allowing us to get a low-variance estimator without introducing any bias. Using Equation (3), the MIS gradient estimator is

$$\widehat{\nabla}^{\text{MIS}}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P}) = \tfrac{1}{n} \sum_{j=0}^{m} \sum_{i=1}^{n_j} w_j^{\text{MIS}}(\boldsymbol{\tau}_{i,j}) g(\boldsymbol{\tau}_{i,j}) \mathcal{R}(\boldsymbol{\tau}_{i,j}), \quad (6)$$

where $w_j^{\text{MIS}}(\boldsymbol{\tau}) := \frac{n}{n_j} h_j(\boldsymbol{\tau}) w_j^{\text{IS}}(\boldsymbol{\tau})$ for a general heuristic function h. For brevity, we define $q_{\boldsymbol{\alpha}}(\boldsymbol{\tau}) := \sum_{j=0}^{m} \alpha_j p(\boldsymbol{\tau}|\boldsymbol{\theta}_j, \mathcal{P}_j)$, so that the weights for the particular choice of balance heuristics reduce to $w_j^{\text{MIS}}(\boldsymbol{\tau}) := \frac{p(\boldsymbol{\tau}|\boldsymbol{\theta}, \mathcal{P})}{q_{\boldsymbol{\alpha}}(\boldsymbol{\tau})}$.

Algorithm 1 outlines how these weighted estimators are used in our proposed transfer procedure. Line 1 initializes the policy parameters. A criterion for achieving a good jumpstart could be easily derived. Since our primary concern is the optimization rather than the initialization, we leave this step unspecified. At each iteration, we decide adaptively the batch size (i.e., the number of defensive samples) that guarantees a minimum ESS for the resulting dataset (line 4). The rationale is that, if the gradient of the current policy can be reliably estimated using the source samples, there is no need to collect new trajectories at all. In order to carry out this step, we derive a lower bound on the increase rate of the approximate ESS as a function of n_0. Due to space constraints, we defer this discussion to Appendix A. In practice, we impose a minimum batch size of n_{min} to avoid degenerate cases. The new batch (line 5) is then added to the current dataset (line 6). Note that this step has two key implications: (i) the gradient estimator always uses at least n_{min} defensive samples, and (ii) the number m of policy-model pairs in $\mathcal{D}$ grows with the number of iterations as trajectories from the target task (but policies potentially different from the current one) are stored. Finally, the parameters are updated using the chosen weighted estimator on the current dataset.

Algorithm 1 Transfer via Importance Sampling

Require: Target task $\mathcal{M}$, source dataset $\mathcal{D} = \{\langle(\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_{n_j}), \boldsymbol{\theta}_j, \mathcal{P}_j\rangle \mid j = 1, \dots, m\}$, gradient estimator $\widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P})$, effective sample size estimator $\widehat{\text{ESS}}(n_0; \mathcal{D})$, minimum effective sample size ESS_{min}, minimum batch size n_{min}, step-size sequence η_k

1: Initialize policy: $\boldsymbol{\theta}_0 \leftarrow$ INIT-POLICY$(\mathcal{M}, \mathcal{D})$
2: Initialize iteration count: $k \leftarrow 0$
3: **while** not converged **do**
4: Find the minimum $n_0 \in \{n_{\text{min}}, \dots, \text{ESS}_{\text{min}}\}$ such that $\widehat{\text{ESS}}(n_0; \mathcal{D}) \geq \text{ESS}_{\text{min}}$
5: Sample n_0 trajectories from $\mathcal{M}$ under policy $\pi_{\boldsymbol{\theta}_k}$
6: Store samples: $\mathcal{D} \leftarrow \mathcal{D} \cup \{\langle(\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_{n_0}), \boldsymbol{\theta}_k, \mathcal{P}\rangle\}$
7: Update parameters: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \eta_k \widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k, \mathcal{P})$
8: **end while**

The next two sections propose two techniques to further reduce the variance of the MIS estimator without introducing any bias.

3.2. Per-decision Estimators

Per-decision IS (Precup, 2000) is a common variance reduction technique from off-policy evaluation. It relies on the intuition that future actions cannot influence past rewards, i.e., each reward r_t should be weighted only by the probability of a trajectory up to that time. This technique can be easily combined with MIS, leading to the PD estimator,

$$\widehat{\nabla}^{\text{PD}}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P}) = \tfrac{1}{n} \sum_{j=0}^{m} \sum_{i=1}^{n_j} \sum_{t=0}^{T-1} \gamma^t w_{j,t}^{\text{PD}}(\boldsymbol{\tau}_{i,j}) g_t(\boldsymbol{\tau}_{i,j}) r_t, \quad (7)$$

where $g_t(\boldsymbol{\tau}) := g(\boldsymbol{\tau}_{0:t})$, with $\boldsymbol{\tau}_{0:t}$ being a trajectory up to time t, and $w^{\text{PD}}_{j,t}(\boldsymbol{\tau}) := \frac{n}{n_j} h_{j,t}(\boldsymbol{\tau}) w^{\text{IS}}_j(\boldsymbol{\tau}_{0:t})$. Using the balance heuristics, $w^{\text{PD}}_{j,t}(\boldsymbol{\tau}) = w^{\text{MIS}}_j(\boldsymbol{\tau}_{0:t})$. Notice that the heuristics is now a function of time. We show that, if this function is uniformly normalized over time, the resulting estimator remains unbiased.

Theorem 3.1 (Unbiasedness of PD estimator)**.** *Let $h_{j,t}(\boldsymbol{\tau})$ be a function such that, for all $t \in \{0, \dots, T-1\}$ and $\boldsymbol{\tau}$, $\sum_{j=0}^{m} h_{j,t}(\boldsymbol{\tau}) = 1$. Then, the per-decision MIS estimator in (7) is unbiased.*

3.3. Regression-based Control Variates

Control variates (CVs) are a widely applied variance reduction technique for general Monte Carlo estimators (Hammersley & Handscomb, 1964). The key idea is that a random variable with known expectation could be used to reduce the variance of a mean estimator for another random variable. Owen & Zhou (2000) have popularized the usage of CVs for IS and MIS. For the d-th dimension of the gradient, consider a vector of functions $\boldsymbol{\psi}_d(\boldsymbol{\tau}) := [\psi_{0,d}(\boldsymbol{\tau}), \dots, \psi_{m+1,d}(\boldsymbol{\tau})]$ such that $\mathbb{E}_{\boldsymbol{\tau} \sim q_{\boldsymbol{\alpha}}}[\psi_{j,d}(\boldsymbol{\tau})] = 0$ for every j. Then, our MIS estimator (6) with $\boldsymbol{\psi}_d$ as CVs becomes

$$\widehat{\nabla}^{\text{CV}}_{\boldsymbol{\theta}_d} J(\boldsymbol{\theta}, \mathcal{P}) = \widehat{\nabla}^{\text{MIS}}_{\boldsymbol{\theta}_d} J(\boldsymbol{\theta}, \mathcal{P}) - \frac{1}{n} \sum_{j=0}^{m} \sum_{i=1}^{n_j} \boldsymbol{\beta}_d^T \boldsymbol{\psi}_d(\boldsymbol{\tau}_{i,j}), \quad (8)$$

where $\boldsymbol{\beta}_d \in \mathbb{R}^{m+1}$ is the vector of CV coefficients. As shown by Owen & Zhou (2000), the proposal distributions composing the mixture of a MIS estimator, whose integral is known to be 1, can be used as very effective control variates. Thus, we consider $\psi_{j,d}(\boldsymbol{\tau}) = \frac{p(\boldsymbol{\tau}|\boldsymbol{\theta}_j, \mathcal{P}_j)}{q_{\boldsymbol{\alpha}}(\boldsymbol{\tau})} - 1$ for $j = 0, \dots, m$. Furthermore, $g(\boldsymbol{\tau})$ is a widely adopted control variate in policy search (a.k.a. baseline) since its expectation is known to be 0. Therefore, we consider $\psi_{m+1,d}(\boldsymbol{\tau}) = \frac{p(\boldsymbol{\tau}|\boldsymbol{\theta}, \mathcal{P}) g_d(\boldsymbol{\tau})}{q_{\boldsymbol{\alpha}}(\boldsymbol{\tau})}$. Finally, the vectors $\boldsymbol{\beta}^*_d$ minimizing the variance of (8) can be approximated by solving the following regression problem (Owen & Zhou, 2000):

$$\arg\min_{\boldsymbol{\beta}_d} \sum_{j=0}^{m} \sum_{i=1}^{n_j} \left(\frac{p(\boldsymbol{\tau}_{i,j})}{q_{\boldsymbol{\alpha}}(\boldsymbol{\tau}_{i,j})} g(\boldsymbol{\tau}_{i,j}) \mathcal{R}(\boldsymbol{\tau}_{i,j}) - \boldsymbol{\beta}_d^T \boldsymbol{\psi}_d(\boldsymbol{\tau}_{i,j}) \right)^2.$$

In practice, we can fit the CVs and estimate the gradient using different partitions of the current dataset to keep an unbiased estimator.

Proposition 3.1. *The estimator (8) is unbiased for any $\boldsymbol{\beta}_d$. Furthermore, under the optimal coefficients $\boldsymbol{\beta}^*_d$, $\mathbb{V}\text{ar}[\widehat{\nabla}^{\text{CV}}_{\boldsymbol{\theta}_d} J(\boldsymbol{\theta}, \mathcal{P})] \leq \mathbb{V}\text{ar}[\widehat{\nabla}^{\text{MIS}}_{\boldsymbol{\theta}_d} J(\boldsymbol{\theta}, \mathcal{P})]$.*

Note that, when the number of dimensions d of the parameter space is large, it is common to fit a unique $\boldsymbol{\beta}$ for all d. In this case, simply taking $\psi_{m+1}(\boldsymbol{\tau}) = \frac{p(\boldsymbol{\tau}|\boldsymbol{\theta}, \mathcal{P}) \sum_d g_d(\boldsymbol{\tau})}{q_{\boldsymbol{\alpha}}(\boldsymbol{\tau})}$ and solving the regression problem is therefore equivalent to (approximately) minimizing $\text{Tr}(\mathbb{C}\text{ov}[\widehat{\nabla}^{\text{CV}}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \mathcal{P})])$. Finally, we note that the CVs can be combined straightforwardly with the PD estimator of Section 3.2.

3.4. Robustness to Negative Transfer

We now show that our MIS estimator with CVs enjoys safety guarantees against negative transfer. We first propose a definition of negative transfer for policy gradient algorithms in terms of convergence to ϵ-optimal stationary points [2].

Definition 3.1 (Negative transfer)**.** *Let $\mathscr{A}$ and $\mathscr{B}$ be two policy gradient algorithms. Fix an initial parameter $\boldsymbol{\theta}_0$, a learning rate η, a batch size n, and an accuracy $\epsilon > 0$. Then, $\mathscr{A}$ negatively transfers w.r.t. $\mathscr{B}$ ($\mathscr{A} \prec \mathscr{B}$) if there exists an iteration number $k \geq 1$ such that we can guarantee that $\frac{1}{k} \sum_{l=0}^{k-1} \mathbb{E}_{\mathscr{B}}[\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_l)\|^2] \leq \epsilon$ but $\frac{1}{k} \sum_{l=0}^{k-1} \mathbb{E}_{\mathscr{A}}[\|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_l)\|^2] > \epsilon$.*

Let $\mathscr{B}_{\text{R}}$ be the REINFORCE algorithm and $\mathscr{B}_{\text{G}}$ be the GPOMDP algorithm (Baxter & Bartlett, 2001). The next result shows that Algorithm 1 using CVs and the MIS estimator ($\mathscr{A}_{\text{CV}}$) or the PD estimator ($\mathscr{A}_{\text{PDCV}}$) cannot be worse than their no-transfer counterparts.

Theorem 3.2. *Assume the return J is L-smooth (i.e., its gradient is L-Lipschitz). Let $n_{min} > 0$ be the minimum batch size for $\mathscr{A}_{CV}$ ($\mathscr{A}_{PDCV}$) and the fixed batch size for $\mathscr{B}_R$ ($\mathscr{B}_G$). Assume all algorithms start from the same parameter $\boldsymbol{\theta}_0$, use a learning rate $0 < \eta \leq \frac{2}{L}$, and that $\mathscr{A}_{CV}$ ($\mathscr{A}_{PDCV}$) uses the optimal CV coefficients $\boldsymbol{\beta}^*_d$. Then, for all $\epsilon > 0$:*

$$\mathscr{A}_{CV} \not\prec \mathscr{B}_R, \quad \mathscr{A}_{PDCV} \not\prec \mathscr{B}_G.$$

We remark that, according to our definition, the fact that $\mathscr{A}$ is robust against $\mathscr{B}$ does not necessarily imply that $\mathscr{A}$ converges faster than $\mathscr{B}$ but only that, whenever we can prove that $\mathscr{B}$ converged, we can also prove that $\mathscr{A}$ converged. Hence, we might say that $\mathscr{A}$ cannot be much worse than $\mathscr{B}$, the standard (weaker) notion of negative transfer that is often considered in the literature (Taylor & Stone, 2009).

4. The Case of Unknown Models

The transfer algorithm presented in Section 3 requires full knowledge of the transition models of each task, an assumption that rarely holds in practice. Here we consider the more realistic case in which the models (equivalently, the importance weights) are unknown and have to be estimated from data. In general, this goal can be efficiently achieved by directly estimating density ratios (Sugiyama et al., 2012). Unfortunately, in our settings, this is not a good approach for at least two reasons: (i) density ratios are not transferable, i.e., they must be recomputed for each new policy and/or task; (ii) our weights are defined over entire trajectories, high-dimensional random variables whose distributions have some structure that would not be exploited by a direct estimator. Therefore, we decide to take a more indirect ap-

[2] As a standard in non-convex optimization, convergence to stationary points could be replaced with convergence to local maxima at the cost of a more complicated analysis.

proach and estimate only the missing components, namely the transition models. However, instead of naively plugging in any density estimator or probabilistic model of the uncertain dynamics, we propose an estimator that is aware of the MIS scheme in which these models will be adopted.

From now on, we consider system dynamics of the form $\boldsymbol{s}_{t+1} = f(\boldsymbol{s}_t, \boldsymbol{a}_t) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_{\mathcal{P}}^2 \boldsymbol{I})$, and we suppose each task to be uniquely identified by its transition function f. Our main assumption is that the source tasks f_j are known, while the target task f is uncertain according to a distribution $\varphi \in \Delta(\mathcal{F})$, where $\mathcal{F}$ is the set of all possible transition functions in the family of tasks under consideration.[3] Let us fix a policy parameter $\boldsymbol{\theta}$ and a dataset $\mathcal{D}$. Let $\nabla J := \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, f)$ denote the true gradient at $\boldsymbol{\theta}$ and $\widehat{\nabla} J(\widetilde{f}) := \widehat{\nabla}_{\boldsymbol{\theta}}^{(\mathrm{MIS})} J(\boldsymbol{\theta}, \widetilde{f})$ denote its MIS estimate (6) using the balance heuristics where an arbitrary function $\widetilde{f}$ is used to compute the importance weights instead of the unknown target model f. Note that, although we assume to know the source distributions, the denominator $q_{\boldsymbol{\alpha}}(\boldsymbol{\tau}) = \sum_{j=0}^{m} \alpha_j p(\boldsymbol{\tau}|\boldsymbol{\theta}_j, f_j)$ is still unknown since at least one of its components depends on f. We write $q_{\boldsymbol{\alpha}}(\boldsymbol{\tau}; f)$ to make this dependence explicit in the remainder. Given the set $\mathcal{J} = \{0, 1, \dots, m\}$, we use $\mathcal{J}_{\mathrm{tgt}} = \{j \in \mathcal{J} | f_j = f\}$ to denote the indexes of proposals from the target task, and define $\mathcal{J}_{\mathrm{src}} = \mathcal{J} \setminus \mathcal{J}_{\mathrm{tgt}}$. Moreover, $\alpha_{\mathrm{tgt}} = \sum_{j \in \mathcal{J}_{\mathrm{tgt}}} \alpha_j$ denotes the proportion of target samples in $\mathcal{D}$, and similarly for $\alpha_{\mathrm{src}} = 1 - \alpha_{\mathrm{tgt}}$. We begin by deriving an upper bound on the mean square error (MSE) of the MIS estimator.

Theorem 4.1. *Let $\widetilde{f} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ be any function and $p_{\boldsymbol{\alpha}}(\boldsymbol{\tau}) = \sum_{j \in \mathcal{J}_{tgt}} \frac{\alpha_j}{\alpha_{tgt}} p(\boldsymbol{\tau}|\boldsymbol{\theta}_j, \widetilde{f})$. Suppose that $\|g(\boldsymbol{\tau})\mathcal{R}(\boldsymbol{\tau})\|_\infty \leq B$ almost surely. Then, for $f \sim \varphi$,*

$$\mathbb{E}\left[\|\widehat{\nabla} J(\widetilde{f}) - \nabla J\|^2\right] \leq \frac{dB^2}{n} d_2\left(p(\cdot|\boldsymbol{\theta}, \widetilde{f}) \| q_{\boldsymbol{\alpha}}(\cdot; \widetilde{f})\right) + c_1 dB^2 \sum_{t=0}^{T-1} \mathbb{E}_{\boldsymbol{\tau} \sim p_{\boldsymbol{\alpha}}}\left[\|\bar{f}(\boldsymbol{s}_t, \boldsymbol{a}_t) - \widetilde{f}(\boldsymbol{s}_t, \boldsymbol{a}_t)\|_2^2\right] + c_1 dB^2 \sum_{t=0}^{T-1} \mathbb{E}_{\boldsymbol{\tau} \sim p_{\boldsymbol{\alpha}}}\left[\mathrm{Tr}\left(\boldsymbol{\Sigma}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right)\right] + \mathcal{O}(n^{-1}), \quad (9)$$

where the expectation is w.r.t. $\boldsymbol{\tau}_{i,j} \sim p(\boldsymbol{\tau}|\boldsymbol{\theta}_j, f_j)$ and $f \sim \varphi$. Here $\bar{f}(\boldsymbol{s}, \boldsymbol{a}) := \mathbb{E}_{f \sim \varphi}[f(\boldsymbol{s}, \boldsymbol{a})]$, $\boldsymbol{\Sigma}(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{C}\mathrm{ov}_{f \sim \varphi}[f(\boldsymbol{s}, \boldsymbol{a})]$, and c_1 is a constant.

Let $\mathcal{L}(\widetilde{f})$ denote the value of this bound as a function of $\widetilde{f}$. Then, we look for the function $f^* \in \mathcal{F}$ that minimizes $\mathcal{L}$. Intuitively, we seek for a model that trades off between three different objectives: (i) when few trajectories are available and the target model is highly uncertain, it should stay close to the mixture of source distributions in order to reduce the variance of the resulting estimator (first term); (ii) as the number of samples grows, it should move towards $\bar{f}$, our best guess for the true model (second term); finally, it should give priority to the regions of the state-action space where the target model is more accurate (third term). Our MSE-aware approach to model estimation is summarized in Algorithm 2. Although appealing, optimizing the bound for the optimal transition function is non-trivial. Now we show two cases in which this can be done efficiently.

[3] A discussion on how our results can be generalized to unknown source tasks is deferred at the end of this section.

Algorithm 2 MSE-aware Model Estimation

Require: Model space $\mathcal{F}$, target trajectories $\mathcal{D}_{\mathrm{tgt}} = \{\mathcal{D}_j | \, j \in \mathcal{J}_{\mathrm{tgt}}\}$, uncertainty model φ
1: Update φ using $\mathcal{D}_{\mathrm{tgt}}$
2: Compute $\widetilde{f} \in \mathcal{F}$ minimizing the bound of Theorem 4.1
3: **return** $\widetilde{f}$

4.1. Discrete Task Family

We start by considering the simple setting in which $\mathcal{F} = \{f_1, f_2, \dots, f_H\}$ is a finite set of possible transition functions. Our uncertainty model is therefore a discrete distribution over this set. Assuming a uniform prior $\varphi_0(f) = \frac{1}{|\mathcal{F}|}$, this distribution can be updated iteratively for every $f \in \mathcal{F}$ given a batch of target trajectories $\mathcal{D}_{\mathrm{tgt}}$ under policy $\boldsymbol{\theta}_k$ as

$$\varphi_{k+1}(f) \propto \varphi_k(f) \prod_{\boldsymbol{\tau} \in \mathcal{D}_{\mathrm{tgt}}} \prod_{t=0}^{T-1} \pi_{\boldsymbol{\theta}_k}(\boldsymbol{a}_t|\boldsymbol{s}_t) \mathcal{P}_f(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t).$$

Given φ_k, the bound of Theorem 4.1 can be easily approximated for every $f \in \mathcal{F}$. Notice that all expectations in (9) are under distributions induced by the model $\widetilde{f}$ and, therefore, they can be approximated by simulating trajectories without interacting with the true environment.

4.2. Reproducing Kernel Hilbert Spaces

We now consider a more general functional space for our transition models. Let $\mathcal{X} = \mathcal{S} \times \mathcal{A}$ and consider a positive semi-definite kernel function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We suppose that $\mathcal{F}$ is the unique reproducing kernel Hilbert space (RKHS) induced by $\mathcal{K}$. In an RKHS, the reproducing property implies that every function f can be written as $f(\boldsymbol{x}) = \langle f, \mathcal{K}(\boldsymbol{x}, \cdot) \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the dot product on $\mathcal{F}$. For simplicity, we consider each dimension of our transition models separately. We refer the reader to (Micchelli & Pontil, 2005) for the extension to vector-valued RKHS.

We represent the uncertainty over the target model as a Gaussian process (GP) (Williams & Rasmussen, 2006), $f \sim \mathcal{GP}(0, \mathcal{K})$, with a zero-mean prior and $\mathcal{K}$ as covariance function. As usual in model-based RL (e.g., (Deisenroth & Rasmussen, 2011)), we train conditionally independent GPs for each dimension of the state space. Suppose that we get a set of l training inputs $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_l]^T$ and l

training targets $\boldsymbol{Y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_l]^T$ from our dataset $\mathcal{D}_{\text{tgt}}$ of trajectories from the target task, where $\boldsymbol{x}_i = (\boldsymbol{s}_i, \boldsymbol{a}_i)$ and $\boldsymbol{y}_i = f(\boldsymbol{x}_i) + \mathcal{N}(0, \sigma_{\mathcal{P}}^2 \boldsymbol{I})$. Then, the posterior mean function can be evaluated at any point $\boldsymbol{x}$ as $\bar{f}(\boldsymbol{x}) = \boldsymbol{k}(\boldsymbol{x})^T (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{Y}$ (Williams & Rasmussen, 2006), where $\boldsymbol{k}(\boldsymbol{x})$ is the vector with entries $\boldsymbol{k}_i(\boldsymbol{x}) = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x})$ and $\boldsymbol{K}$ is the Gram matrix, $K_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Now that we have an uncertainty model for our target transition function, let us move to optimize our bound $\mathcal{L}$ on the MSE. Unfortunately, minimizing $\mathcal{L}(\widetilde{f})$ w.r.t. $\widetilde{f} \in \mathcal{F}$ is not as simple as in the finite-model case since (i) $\widetilde{f}$ controls the distributions under which expectations are taken, and (ii) it appears as a product over several time steps in the Renyi divergence term. For these reasons, we now introduce some simplifications that will lead to a convenient closed-form solution. First, we approximate the two expectations by drawing a small number of trajectories from our last hypothesized model, so that their dependence on the function to be computed is removed. Secondly, we further bound our objective in a more convenient way. In order to carry out this last step, we derive an upper bound on the exponentiated Renyi divergence w.r.t. the Kullback-Leibler (KL) divergence, which could be of independent interest.

Theorem 4.2. *Let $(\mathcal{X}, \mathscr{F})$ be a measurable space, P and Q be two probability measures on $\mathcal{X}$ such that $P \ll Q$, and $Q_\alpha = \alpha P + (1-\alpha) Q$ denotes their convex combination with coefficient $\alpha \in (0,1)$. Suppose there exists a finite constant $C > 0$ such that $\operatorname{ess\,sup} \frac{dP}{dQ} \leq C$. Then,*

$$d_2(P||Q_\alpha) \leq 1 + u(\alpha) D_{KL}(P||Q), \tag{10}$$

where

$$u(\alpha) = \begin{cases} \frac{2C(1-\alpha)^2}{(\alpha C + 1 - \alpha)^3} & \text{if } C \leq \frac{1-\alpha}{2\alpha} \\ \frac{8}{27\alpha} & \text{otherwise.} \end{cases}$$

Using Theorem 4.2, our objective $\mathcal{L}(\widetilde{f})$ can be bounded in a very convenient way.

Proposition 4.1. *The objective $\mathcal{L}(\widetilde{f})$ given in (9) can be upper bounded by*

$$\mathcal{L}(\widetilde{f}) \leq k_1 \sum_{t=0}^{T-1} \mathbb{E}_{\boldsymbol{\tau} \sim p(\cdot|\boldsymbol{\theta}, \widetilde{f})} \left[\sum_{j \in \mathcal{J}_{src}} \alpha_j \|\widetilde{f}(\boldsymbol{x}_t) - f_j(\boldsymbol{x}_t)\|_2^2 \right] + k_2 \sum_{t=0}^{T-1} \mathbb{E}_{\boldsymbol{\tau} \sim p_{\boldsymbol{\alpha}}} \left[\|\widetilde{f}(\boldsymbol{x}_t) - \bar{f}(\boldsymbol{x}_t)\|_2^2 \right] + k_3,$$

where $k_1 = \frac{u(\alpha) d B^2}{2\sigma_{\mathcal{P}}^2 n (1-\alpha_0)}$, $k_2 = \frac{4\alpha_{tgt} d B^2}{\alpha_0^2 \sigma_{\mathcal{P}}^2}$, and k_3 is a constant independent of $\widetilde{f}$.

Our new bound is quite appealing. While the bias term remains unchanged, the variance is now a mixture of expected l_2 distances between $\widetilde{f}$ and the known source models f_j. In practice, we optimize a regularized version of this objective so that the representer theorem of RKHS applies. Furthermore, as mentioned above, we approximate the two expectations by drawing R trajectories from the mixture $p_{\boldsymbol{\alpha}}$ using our last hypothesized model. The resulting objective reduces to a regularized least-squares problem,

$$\arg\min_{\widetilde{f} \in \mathcal{F}} \frac{1}{R} \sum_{r=1}^{R} \sum_{t=0}^{T-1} \left(k_1 w_{r,t} \sum_{j \in \mathcal{J}_{\text{src}}} \alpha_j \|\widetilde{f}(\boldsymbol{x}_{r,t}) - f_j(\boldsymbol{x}_{r,t})\|_2^2 + k_2 \|\bar{f}(\boldsymbol{x}_{r,t}) - \widetilde{f}(\boldsymbol{x}_{r,t})\|_2^2 \right) + \lambda \|f\|_{\mathcal{K}}^2, \tag{11}$$

where $\lambda > 0$ is the regularization parameter and $w_{r,t} = \prod_{l=0}^{t} \frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_l|\boldsymbol{s}_l)}{\sum_{j \in \mathcal{J}_{\text{tgt}}} \frac{\alpha_j}{\alpha_{\text{tgt}}} \pi_{\boldsymbol{\theta}_j}(\boldsymbol{a}_l|\boldsymbol{s}_l)}$ is an importance weight to correct the distribution mismatch in the first expectation. Most importantly, its solution is available in closed form.

Proposition 4.2. *The function f^* minimizing (11) is*

$$f^*(\boldsymbol{x}) = \boldsymbol{A}^T \boldsymbol{k}(\boldsymbol{x}),$$

where $\boldsymbol{k}(\boldsymbol{x})$ is the RT-dimensional vector with entries $\mathcal{K}(\boldsymbol{x}_{r,t}, \boldsymbol{x})$ and

$$\boldsymbol{A} = (k_1 \alpha_{tgt} \boldsymbol{W} \boldsymbol{K} + k_2 \boldsymbol{K} + \lambda R \boldsymbol{I})^{-1} (k_1 \boldsymbol{W} \boldsymbol{F}_{src} + k_2 \bar{\boldsymbol{F}}),$$

with $\boldsymbol{K}$ being the Gram matrix, $\boldsymbol{W} = diag(w_{r,t})$, $\bar{\boldsymbol{F}} = [\bar{f}(\boldsymbol{x}_{1,0}), \dots, \bar{f}(\boldsymbol{x}_{R,T-1})]^T$, $\boldsymbol{F}_{src} = \sum_{j \in \mathcal{J}_{src}} \alpha_j \boldsymbol{F}_j$, and $\boldsymbol{F}_j = [f_j(\boldsymbol{x}_{1,0}), \dots, f_j(\boldsymbol{x}_{R,T-1})]^T$.

Discussion One might be wondering why the estimated transition models are not directly used for planning in a standard model-based RL algorithm instead of estimating the importance weights for a model-free approach. It is well-known that even small errors can lead to disastrous performances when the optimal policy is computed under the estimated models. Since in our case the learned models are only used to re-weight samples from the true environment, we argue that the impact of such errors is much more contained. In fact, as far as the weights keep reasonable values, the learning process could potentially be carried out effectively. Furthermore, note that our estimators are consistent as the number of target samples goes to infinity for *any* hypothesized model, not necessarily the true one.

Remark 4.1 (Unknown source models)**.** *The extension of Theorem 4.1 to unknown source models is reported in Appendix B. Unfortunately, optimizing the resulting bound over all models becomes considerably more difficult due to the combinatorial growth of the problem. However, in practice, we typically update online only the target model. In fact, we cannot require additional source samples and, thus, we can estimate the source models in batch before learning starts. Therefore, our algorithm can be straightforwardly applied, e.g.,, by fitting GPs and then plugging the MAP estimates into Theorem 4.1 instead of the true source models (hence our simplification of known sources).*

5. Related Works

The closest work to ours is the recent paper of Tirinzoni et al. (2018b). The authors propose an IS-based algorithm to transfer samples in a value-based batch RL setting, together with a methodology to estimate the weights using Gaussian processes. Besides extending these algorithms to policy search, our MIS estimators naturally combine multiple distributions and are much more robust to source tasks that are very different from the target, a drawback of plain IS.

Among the existing transfer approaches for policy search, Ammar et al. (2014) and Ammar et al. (2015) focus on a lifelong learning scenario, where the agent continually faces new tasks sampled from a common distribution and must quickly adapt to each of them. Similarly to our work, several recent papers provide theoretical guarantees on the transfer procedure (e.g., Brunskill & Li, 2013; Zhan et al., 2016; Barreto et al., 2017; Abel et al., 2018; Tirinzoni et al., 2018a). Our assumption of environments with different dynamics is also related to Hidden-parameter MDPs (Doshi velez & Konidaris, 2013; Killian et al., 2017), although we do not require transitions to be parameterized. Crucially, while we both learn transition models, we do not use these for planning but only for computing the weights, which is typically more robust to estimation errors.

Finally, our MIS estimators are related to, and might be of interest for, different RL settings such as off-policy evaluation (Precup, 2000; Hachiya et al., 2009; Thomas & Brunskill, 2016; Guo et al., 2017; Liu et al., 2018), off-policy learning (Precup et al., 2001; Mahmood et al., 2014; Geist & Scherrer, 2014; Munos et al., 2016; Metelli et al., 2018), and sample reuse (Zhao et al., 2013; Hachiya et al., 2011).

6. Experiments

We analyze the performance of our estimators with known models in Section 6.1 and with model estimation in Sections 6.2 and 6.3. Due to space constraints, we only provide the high-level configuration of each experiment, while referring the reader to Appendix D for the specific hyperparameters that were adopted (see Table 1 for a quick summary).

6.1. Linear-Quadratic Regulator

Our first test domain is the one-dimensional linear-quadratic regulator (LQR) (Dorato et al., 1995; Peters & Schaal, 2008), a well-known benchmark from the control literature. The system has linear dynamics, $s_{t+1} = As_t + Ba_t + \epsilon$, with Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_{\mathcal{P}}^2)$, and quadratic rewards.

We begin by evaluating the MIS estimators proposed in Section 3. Besides our proposed estimators, we compare to per-decision IS (PD-IS), which is widely adopted in the literature and can be straightforwardly adapted to our case, and to GPOMDP (Baxter & Bartlett, 2001) as our no-transfer baseline. The source tasks are randomly generated by uniformly sampling A in $[0.6, 1.4]$ and B in $[0.8, 1.2]$, while the target task is fixed with $A = 1$ and $B = 1$. We employ linear Gaussian policies and consider 8 source parameters, $\theta \in \{-0.1, -0.2, \dots, -0.8\}$, generating 20 episodes from each model-policy couple. To have a fair comparison, we use the same learning rate and initialization for all algorithms. Figure 1*(left)* shows the distance to the optimal target parameter as a function of the number of episodes. As expected, PD-IS shows a significant amount of negative transfer w.r.t. GPOMDP. This is due to the fact that the huge variance of the importance weights forces the algorithm to collect large batches to guarantee the required ESS. MIS and PD-MIS achieve an improvement over the no-transfer baseline, with the latter having smaller variance. When introducing CVs, the algorithm enjoys much better gradient estimates and significantly outperforms all alternatives.

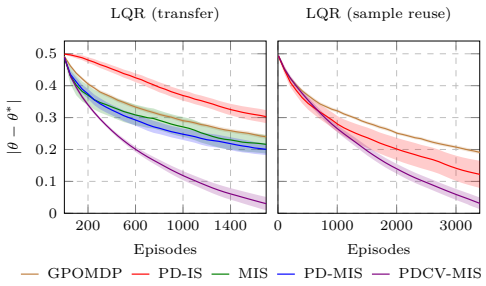

Figure 1. Comparison of the proposed gradient estimators in the LQR domain under known models. (left) transfer experiment, and (right) sample reuse experiment. Each curve is the average of 40 independent runs, each re-sampling the source tasks, with Student's t 95% confidence intervals

Sample Reuse in Policy Gradients Our estimators can be successfully adopted to reuse samples generated by previous policies in standard (no-transfer) policy gradients. Figure 1*(right)* shows the result of learning the same target task as before from scratch (i.e., without any source sample), where each algorithm uses the same (fixed) batch size, learning rate, and initialization. We can appreciate that both the per-decision and our estimator enjoy a speedup over GPOMDP, but the former suffers a much higher variance.

6.2. Cart-pole Balancing

Our second domain is the well-known Cartpole problem (Sutton & Barto, 1998), where the goal is to balance a pole on a moving cart. We generate source tasks by uniformly sampling the mass of the cart in the interval $[0.8, 1.2]$ and the length of the pole in the interval $[0.3, 0.7]$. The target task is the standard Cartpole with cart mass 1.0 and pole length 0.5. For each of 5 source tasks, we consider a sequence of 10 linear policies generated by GPOMDP during its learning

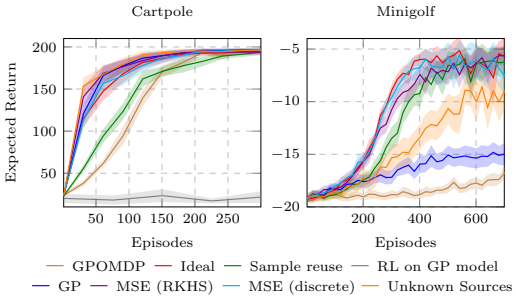

Figure 2. Comparison of our model estimation approach in the Cartpole (left) and minigolf (right) domains. Each curve is the average of 40 independent runs, each re-sampling the source tasks, with Student's t 95% confidence intervals.

process and collect 10 episodes from each. While the LQR domain was relatively short-horizon ($T = 20$), here we allow trajectories up to $T = 200$ time steps, which allows us to verify the transferability of long state-action sequences.

We now test our model estimation approaches. Besides GPOMDP, we report the performance of the sample reuse (SR) variant of our algorithm where we ignore the source tasks and transfer only past target trajectories. All transfer algorithms use the PD-MIS estimator. Figure 2*(left)* shows the results. Interestingly, all transfer approaches outperform both GPOMDP and SR, which confirms that reusing trajectories from different environments can significantly improve the quality of the learning process. Both our model estimation approaches perform comparably to the ideal estimators. While this should be expected for the discrete estimator, the continuous one works well since an accurate GP model of the Cartpole dynamics can be obtained with a relatively small amount of samples. This fact can be verified from the GP curve, where the weights have been estimated by directly plugging in the GP predictions instead of optimizing our bound. Not surprisingly, the algorithm that estimates the source models as in Remark 4.1 performs comparably to the best alternatives. However, the fact that the fitted GP is accurate for estimating the weights does not imply that it is an accurate model of the system dynamics which can be used for planning. The gray curve in Figure 2*(left)*, which shows the performance achieved by optimal policies for the estimated dynamics, confirms this statement.

6.3. Minigolf

In this last domain, we want to study how much the proposed transfer algorithms can speed up the learning process of an agent playing a minigolf game by reusing the experience made on different minigolf courses. The various tasks may differ in the length of the putter (between $70cm$ and $100cm$), in the hole size (between $10cm$ and $15cm$), and in the dynamic friction coefficient (whose range was measured empirically (Penner, 2002) between 0.065 and 0.196). The minigolf domain was originally introduced in the RL field by Lazaric et al. (2008a); here, we change the dynamics following the modeling developed by Penner (2002) in order to make the problem more realistic. A detailed description of the minigolf domain is available in Appendix D.3.

In this experiment, we adopt Gaussian policies with a fourth-order polynomial basis function. We generated 5 source tasks by randomly sampling dynamic friction coefficient, hole size, and putter length from the realistic ranges defined above. Furthermore, we considered 10 source policies of increasing quality and generated 40 episodes from each model-policy pair. The target task is fixed with a friction of 0.131, a putter of $100cm$, and a hole of diameter $10cm$. All transfer algorithms use the PDCV estimator. The results are shown in Figure 2*(right)*. Unlike the simpler Cartpole domain, GPOMDP is not able to learn the task in such a small number of episodes. Interestingly, due to the high level of noise present in this environment, direct estimation of weights using the GP predictions leads to unsatisfactory results. On the other hand, both our model estimation approaches solve the task with performance comparable to the ideal estimator. We note that the speed-up over the SR algorithm is not as remarkable as in the previous experiment due to the little amount of knowledge transfer that can be achieved in this more complicated setting. We finally point out that most of the simplifications introduced in the previous sections do not hold in this domain. In fact, the transition models are not Gaussian, while the noise is heteroscedastic and changes between tasks. Despite these relaxations, our approach can be applied without suffering any considerable performance degradation.

7. Conclusion

In this paper, we introduced a methodology which employs multiple importance sampling for transferring samples (i.e., entire trajectories) to reduce the variance of the estimated gradients in policy search. We showed that our estimators are general, in the sense that they can be of interest outside the transfer literature, and they enjoy strong theoretical properties. We proposed a methodology to estimate the unknown task models in a principled way by directly minimizing an upper bound on the MSE that these models induce on the resulting importance-weighted estimator. Finally, we empirically demonstrated the effectiveness of our algorithm in different domains, both in the ideal and estimated cases.

An interesting question is whether our method could be generalized to the policy gradient theorem (Sutton et al., 2000), where the importance weights would be on the stationary distributions of single states. This could dramatically increase the amount of transferred knowledge, as shown by Liu et al. (2018) for the case of off-policy evaluation.

References

Abel, D., Jinnai, Y., Guo, S. Y., Konidaris, G., and Littman, M. Policy and value transfer in lifelong reinforcement learning. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 20–29, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Allen-Zhu, Z. Natasha 2: Faster non-convex optimization than sgd. *arXiv preprint arXiv:1708.08694*, 2017.

Ammar, H. B., Eaton, E., Ruvolo, P., and Taylor, M. Online multi-task learning for policy gradient methods. In *International Conference on Machine Learning*, pp. 1206–1214, 2014.

Ammar, H. B., Eaton, E., Luna, J.-M., and Ruvolo, P. Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning. In *IJCAI*, 2015.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. Successor features for transfer in reinforcement learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4055–4065. Curran Associates, Inc., 2017.

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

Brunskill, E. and Li, L. Sample complexity of multi-task reinforcement learning. *arXiv preprint arXiv:1309.6821*, 2013.

Cortes, C., Mansour, Y., and Mohri, M. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pp. 442–450, 2010.

Csiszár, I. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

Deisenroth, M. and Rasmussen, C. E. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

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

Dorato, P., Abdallah, C. T., Cerone, V., and Jacobson, D. H. *Linear-quadratic control: an introduction*. Prentice Hall Englewood Cliffs, NJ, 1995.

Doshi velez, F. and Konidaris, G. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. *IJCAI : proceedings of the conference*, 2016, 08 2013.

Elvira, V., Martino, L., and Robert, C. P. Rethinking the effective sample size. *arXiv preprint arXiv:1809.04129*, 2018.

Geist, M. and Scherrer, B. Off-policy learning with eligibility traces: A survey. *The Journal of Machine Learning Research*, 15(1):289–333, 2014.

Guo, Z., Thomas, P. S., and Brunskill, E. Using options and covariance testing for long horizon off-policy policy evaluation. In *Advances in Neural Information Processing Systems*, pp. 2492–2501, 2017.

Hachiya, H., Akiyama, T., Sugiayma, M., and Peters, J. Adaptive importance sampling for value function approximation in off-policy reinforcement learning. *Neural Networks*, 22(10):1399–1410, 2009.

Hachiya, H., Peters, J., and Sugiyama, M. Reward-weighted regression with sample reuse for direct policy search in reinforcement learning. *Neural Computation*, 23(11): 2798–2832, 2011.

Hammersley, J. and Handscomb, D. *Monte Carlo Methods*. Methuen's monographs on applied probability and statistics. Methuen, 1964. ISBN 9780416523409. URL https://books.google.it/books?id=Kk4OAAAAQAAJ.

Hesterberg, T. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2): 185–194, 1995.

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

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 652–661, New York, New York, USA, 20–22 Jun 2016. PMLR.

Killian, T. W., Daulton, S., Konidaris, G., and Doshi-Velez, F. Robust and efficient transfer learning with hidden parameter markov decision processes. In *Advances in Neural Information Processing Systems*, pp. 6250–6261, 2017.

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

Laroche, R. and Barlier, M. Transfer reinforcement learning with shared dynamics. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Lazaric, A. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pp. 143–173. Springer, 2012.

Lazaric, A., Restelli, M., and Bonarini, A. Reinforcement learning in continuous action spaces through sequential monte carlo methods. In *Advances in neural information processing systems*, pp. 833–840, 2008a.

Lazaric, A., Restelli, M., and Bonarini, A. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 544–551. ACM, 2008b.

Le Cam, L. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 1986.

Li, L., Munos, R., and Szepesvari, C. Toward Minimax Off-policy Value Estimation. In Lebanon, G. and Vishwanathan, S. V. N. (eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 608–616, San Diego, California, USA, 09–12 May 2015. PMLR.

Liu, J. S. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119, 1996.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pp. 5361–5371, 2018.

Mahmood, A. R., van Hasselt, H. P., and Sutton, R. S. Weighted importance sampling for off-policy learning with linear function approximation. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 3014–3022. Curran Associates, Inc., 2014.

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

Metelli, A. M., Papini, M., Faccio, F., and Restelli, M. Policy optimization via importance sampling. In *Advances in Neural Information Processing Systems*, pp. 5447–5459, 2018.

Micchelli, C. A. and Pontil, M. On learning vector-valued functions. *Neural computation*, 17(1):177–204, 2005.

Munos, R., Stepleton, T., Harutyunyan, A., and Bellemare, M. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1054–1062, 2016.

Owen, A. and Zhou, Y. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

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

Pearson, K. X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

Penner, A. The physics of putting. *Canadian Journal of Physics*, 80(2):83–96, 2002.

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

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

Precup, D., Sutton, R., and Dasgupta, S. Off-policy temporal-difference learning with function approximation. *Proceedings of the 18th International Conference on Machine Learning*, 06 2001.

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

Ryu, E. K. *Convex optimization for Monte Carlo: Stochastic optimization for importance sampling*. PhD thesis, Stanford University, 2016.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

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

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Taneja, I. J. Generalized relative information and information inequalities. *Journal of Inequalities in Pure and Applied Mathematics*, 5(1):1–19, 2004.

Taylor, M. E. and Stone, P. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

Taylor, M. E., Jong, N. K., and Stone, P. Transferring instances for model-based reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 488–505. Springer, 2008.

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

Tirinzoni, A., Rodriguez Sanchez, R., and Restelli, M. Transfer of value functions via variational methods. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6179–6189. Curran Associates, Inc., 2018a.

Tirinzoni, A., Sessa, A., Pirotta, M., and Restelli, M. Importance weighted transfer of samples in reinforcement learning. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4936–4945, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018b. PMLR.

Veach, E. *Robust monte carlo methods for light transport simulation*. Number 1610. Stanford University PhD thesis, 1997.

Veach, E. and Guibas, L. J. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pp. 419–428. ACM, 1995.

Williams, C. K. and Rasmussen, C. E. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

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

Zhan, Y., Ammar, H. B., et al. Theoretically-grounded policy advice from multiple teachers in reinforcement learning settings with applications to negative transfer. *arXiv preprint arXiv:1604.03986*, 2016.

Zhao, T., Hachiya, H., Tangkaratt, V., Morimoto, J., and Sugiyama, M. Efficient sample reuse in policy gradients with parameter-based exploration. *Neural computation*, 25(6):1512–1547, 2013.