Efficient Sample Reuse in Policy Gradients with Parameter-based Exploration

Tingting Zhao, Hirotaka Hachiya, Voot Tangkaratt
Tokyo Institute of Technology, Japan.
tingting@sg.cs.titech.ac.jp
hacchan@gmail.com
voot@sg.cs.titech.ac.jp

Jun Morimoto
ATR Computational Neuroscience Labs, Japan
xmorimo@atr.jp

Masashi Sugiyama
Tokyo Institute of Technology, Japan.
sugi@cs.titech.ac.jp http://sugiyama-www.cs.titech.ac.jp/~sugi

Abstract

The policy gradient approach is a flexible and powerful reinforcement learning method particularly for problems with continuous actions such as robot control. A common challenge in this scenario is how to reduce the variance of policy gradient estimates for reliable policy updates. In this paper, we combine the following three ideas and give a highly effective policy gradient method: (a) the *policy gradients with parameter based exploration*, which is a recently proposed policy search method with low variance of gradient estimates, (b) an *importance sampling technique*, which allows us to reuse previously gathered data in a consistent way, and (c) an *optimal baseline*, which minimizes the variance of gradient estimates with their unbiasedness being maintained. For the proposed method, we give theoretical analysis of the variance of gradient estimates and show its usefulness through extensive experiments.

1 Introduction

The objective of *reinforcement learning* (RL) is to let an agent optimize its decision-making policy through interaction with an unknown environment [25]. Among possible approaches, *policy search* has become a popular method because of its direct nature for policy learning [1]. Particularly, in high-dimensional problems with continuous states and actions, policy search has been shown to be highly useful in practice [14, 16].

Among policy search methods [3], gradient-based methods are popular in physical control tasks because policies are changed gradually [26, 10, 16] and thus steady performance improvement is ensured until a local optimal policy has been obtained. However, since the gradients estimated with these methods tend to have large variance and thus they may suffer from slow convergence.

Recently, a novel approach to using policy gradients called *policy gradients with parameter based exploration* (PGPE) was proposed [20]. PGPE tends to produce gradient estimates with low variance by removing unnecessary randomness from policies and introducing useful stochasticity by considering a prior distribution for policy parameters. PGPE was shown to be more promising than alternative approaches experimentally [20, 32]. However, PGPE still requires a relatively large number of samples to obtain accurate gradient estimates, which can be a critical bottleneck in real-world applications that require large costs and time in data collection.

To overcome this weakness, an *importance sampling* technique [7] is useful under the *off-policy* RL scenario, where a data-collecting policy and the current target policy are different in general [25]. An importance sampling technique allows us to reuse previously collected data, which are collected following policies different from the current one in a consistent manner [25, 22]. However, naively using an importance sampling technique significantly increases the variance of gradient estimates, which can cause sudden changes in policy updates [21, 15, 9, 28]. To mitigate this problem, variance reduction techniques such as decomposition [18], truncation [28, 27], normalization [21, 15], and flattening [9] of importance weights are often used. However, these methods commonly suffer from the bias-variance trade-off, meaning that the variance is reduced at the expense of increasing the bias.

The purpose of this paper is to propose a new approach to systematically addressing the large variance problem in policy search. Basically, this work is an extension of our previous research [32] to an *off-policy* scenario using an importance weighting technique. More specifically, we first give an off-policy implementation of PGPE called the *importance-weighted PGPE* (IW-PGPE) method for consistent sample reuse. We then derive the optimal baseline for IW-PGPE to minimize the variance of importance-weighted gradient estimates, following [8, 29]. We show that the proposed method can achieve significant performance improvement over alternative approaches in experiments with an artificial domain. We also investigate that combining the proposed method with the truncation technique can further improve the performance in high-dimensional problems.

2 Formulations of Policy Gradient

In this paper, we consider the standard framework of episodic reinforcement learning (RL) in which an agent interacts with an environment modeled as a *Markov decision process* (MDP) [25]. In this section, we first review a standard formulation of policy gradient methods [31, 10, 16]. Then we show an alternative formulation adopted in the PGPE (policy gradients with parameter based exploration) method [20].

2.1 Standard Formulation

We assume that the underlying control problem is a discrete-time MDP. At each discrete time step t, the agent observes a state $\boldsymbol{s}_t \in \mathcal{S}$, selects an action $a_t \in \mathcal{A}$, and then receives an immediate reward r_t resulting from a state transition in the environment. The state $\mathcal{S}$ and action $\mathcal{A}$ are both defined as continuous spaces in this paper[1]. The dynamics of the environment are characterized by $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t)$, which represents the transition probability density from the current state $\boldsymbol{s}_t$ to the next state $\boldsymbol{s}_{t+1}$ when action a_t is taken, and $p(\boldsymbol{s}_1)$ is the probability density of initial states. The immediate reward r_t is given according to the reward function $r(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1})$.

The agent's decision making procedure at each time step t is characterized by a parameterized policy $p(a_t|\boldsymbol{s}_t, \boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$, which represents the conditional probability density of taking action a_t in state $\boldsymbol{s}_t$. We assume that the policy is continuously differentiable with respect to its parameter $\boldsymbol{\theta}$.

A sequence of states and actions forms a *trajectory* denoted by

$$h := [\boldsymbol{s}_1, a_1, \ldots, \boldsymbol{s}_T, a_T],$$

where T denotes the number of steps called horizon length. In this paper, we assume that T is a fixed deterministic number. Note that the action a_t is chosen independently of the trajectory given $\boldsymbol{s}_t$ and $\boldsymbol{\theta}$. Then the discounted cumulative reward along h, called the *return*, is given by

$$R(h) := \sum_{t=1}^{T} \gamma^{t-1} r(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}),$$

where $\gamma \in [0, 1)$ is the discount factor for future rewards.

The goal is to optimize the policy parameter $\boldsymbol{\theta}$ so that the *expected return* is maximized. The expected return for policy parameter $\boldsymbol{\theta}$ is defined by

$$J(\boldsymbol{\theta}) := \int p(h|\boldsymbol{\theta}) R(h) \mathrm{d}h,$$

where

$$p(h|\boldsymbol{\theta}) = p(\boldsymbol{s}_1) \prod_{t=1}^{T} p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a_t) p(a_t|\boldsymbol{s}_t, \boldsymbol{\theta}).$$

The most straightforward way to update the policy parameter is to follow the gradient in policy parameter space using gradient ascent:

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} + \varepsilon \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}),$$

where ε is a small positive constant, called the learning rate.

This is a standard formulation of policy gradient methods [31, 10, 16]. The central problem is to estimate the policy gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ accurately from trajectory samples.

[1]Note that continuous formulation is not an essential restriction.

2.2 Alternative Formulation

However, standard policy gradient methods were shown to suffer from high variance in the gradient estimation due to randomness introduced by the stochastic policy model $p(a|\boldsymbol{s},\boldsymbol{\theta})$ [32]. To cope with this problem, an alternative method called *policy gradients with parameter based exploration* (PGPE) was proposed recently [20]. The basic idea of PGPE is to use a deterministic policy and introduce stochasticity by drawing parameters from a prior distribution. More specifically, parameters are sampled from the prior distribution at the start of each trajectory, and thereafter the controller is deterministic[2]. Thanks to this per-trajectory formulation, the variance of gradient estimates in PGPE does not increase with respect to trajectory length T. Below, we review PGPE.

PGPE uses a deterministic policy with typically a linear architecture:

$$p(a|\boldsymbol{s},\boldsymbol{\theta}) = \delta(a = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{s})), \tag{1}$$

where $\delta(\cdot)$ is the *Dirac delta function*, $\boldsymbol{\phi}(\boldsymbol{s})$ is an ℓ-dimensional basis function vector, and $^\top$ denotes the transpose. The policy parameter $\boldsymbol{\theta}$ is drawn from a prior distribution $p(\boldsymbol{\theta}|\boldsymbol{\rho})$ with hyper-parameter $\boldsymbol{\rho}$.

The expected return in the PGPE formulation is defined in terms of expectations over both h and $\boldsymbol{\theta}$ as a function of hyper-parameter $\boldsymbol{\rho}$:

$$\mathcal{J}(\boldsymbol{\rho}) := \iint p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho})R(h)\mathrm{d}h\mathrm{d}\boldsymbol{\theta}.$$

In PGPE, the hyper-parameter $\boldsymbol{\rho}$ is optimized so as to maximize $\mathcal{J}(\boldsymbol{\rho})$, i.e., the optimal hyper-parameter $\boldsymbol{\rho}^*$ is given by

$$\boldsymbol{\rho}^* := \arg\max_{\boldsymbol{\rho}} \mathcal{J}(\boldsymbol{\rho}).$$

In practice, a gradient method is used to find $\boldsymbol{\rho}^*$:

$$\boldsymbol{\rho} \longleftarrow \boldsymbol{\rho} + \varepsilon \nabla_{\boldsymbol{\rho}} \mathcal{J}(\boldsymbol{\rho}),$$

where $\nabla_{\boldsymbol{\rho}}\mathcal{J}(\boldsymbol{\rho})$ is the derivative of $\mathcal{J}$ with respect to $\boldsymbol{\rho}$:

$$\nabla_{\boldsymbol{\rho}}\mathcal{J}(\boldsymbol{\rho}) = \iint p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho})\nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})R(h)\mathrm{d}h\mathrm{d}\boldsymbol{\theta}. \tag{2}$$

Note that, in the derivation of the gradient, the logarithmic derivative,

$$\nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho}) = \frac{\nabla_{\boldsymbol{\rho}} p(\boldsymbol{\theta}|\boldsymbol{\rho})}{p(\boldsymbol{\theta}|\boldsymbol{\rho})},$$

[2]Note that transitions are stochastic, and thus trajectories are also stochastic even though the policy is deterministic.

was used. The expectations over h and $\boldsymbol{\theta}$ are approximated by the empirical averages:

$$\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}(\boldsymbol{\rho}) = \frac{1}{N} \sum_{n=1}^{N} \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}_n | \boldsymbol{\rho}) R(h_n), \tag{3}$$

where each trajectory sample h_n is drawn independently from $p(h|\boldsymbol{\theta}_n)$ and parameter $\boldsymbol{\theta}_n$ is drawn from $p(\boldsymbol{\theta}_n|\boldsymbol{\rho})$. We denote samples collected at the current iteration as

$$D = \{(\boldsymbol{\theta}_n, h_n)\}_{n=1}^{N}.$$

Following [20], in this paper we employ a Gaussian distribution as the distribution of the policy parameter $\boldsymbol{\theta}$ with the hyper-parameter $\boldsymbol{\rho}$. However, other distributions can also be allowed. When assuming a Gaussian distribution, the hyper-parameter $\boldsymbol{\rho}$ consists of a set of means $\{\eta_i\}$ and standard deviations $\{\tau_i\}$, which determine the prior distribution for each element θ_i in $\boldsymbol{\theta}$ of the form

$$p(\theta_i|\rho_i) = \mathcal{N}(\theta_i|\eta_i, \tau_i^2),$$

where $\mathcal{N}(\theta_i|\eta_i, \tau_i^2)$ denotes the normal distribution with mean η_i and variance τ_i^2. Then the derivative of $\log p(\boldsymbol{\theta}|\boldsymbol{\rho})$ with respect to η_i and τ_i are given as

$$\begin{aligned}
\nabla_{\eta_i} \log p(\boldsymbol{\theta}|\boldsymbol{\rho}) =& \frac{\theta_i - \eta_i}{\tau_i^2}, \\
\nabla_{\tau_i} \log p(\boldsymbol{\theta}|\boldsymbol{\rho}) =& \frac{(\theta_i - \eta_i)^2 - \tau_i^2}{\tau_i^3},
\end{aligned}$$

which can be substituted into Eq.(3) to approximate the gradients with respect to $\boldsymbol{\eta}$ and $\boldsymbol{\tau}$. These gradients give the PGPE update rules.

An advantage of PGPE is its low variance of gradient estimates: Compared with a standard policy gradient method REINFORCE [31], PGPE was empirically demonstrated to be better in some settings [20, 32]. The variance of gradient estimates in PGPE can be further reduced by subtracting an optimal baseline (Theorem 4 of [32]).

Another advantage of PGPE is its high flexibility: In standard policy gradient methods, the parameter $\boldsymbol{\theta}$ is used to determine a stochastic policy model $p(a|\boldsymbol{s}, \boldsymbol{\theta})$, and policy gradients are calculated by differentiating the policy with respect to the parameter. However, because PGPE needs not calculate the derivative of the policy, a non-differentiable controller is also allowed.

3 Off-Policy Extension of PGPE

In real-world applications such as robot control, gathering roll-out data is often costly. Thus, we want to keep the number of samples as small as possible. However, when the number of samples is small, policy gradients estimated by the original PGPE are not reliable enough.

The original PGPE is categorized as an *on-policy* algorithm [25], where data drawn from the current target policy is used to estimate policy gradients. On the other hand, *off-policy* algorithms are more flexible in the sense that a data-collecting policy and the current target policy can be different. In this section, we extend PGPE to an *off-policy* scenario using importance-weighting, which allows us to reuse previously collected data in a consistent manner. We also theoretically analyze properties of the extended method.

3.1 Importance-Weighted PGPE

Let us consider an off-policy scenario where a data-collecting policy and the current target policy are different in general. In the context of PGPE, we consider two hyper-parameters, $\boldsymbol{\rho}$ for the target policy to learn and $\boldsymbol{\rho}'$ for data collection. Let us denote data samples collected with hyper-parameter $\boldsymbol{\rho}'$ by D':

$$D' = \{(\boldsymbol{\theta}'_n, h'_n)\}_{n=1}^{N'} \overset{i.i.d}{\sim} p(h, \boldsymbol{\theta}|\boldsymbol{\rho}') = p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho}').$$

If we naively use data D' to estimate policy gradients by Eq.(3), we have an inconsistency problem:

$$\frac{1}{N'}\sum_{n=1}^{N'} \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}'_n|\boldsymbol{\rho}) R(h'_n) \overset{N'\to\infty}{\nrightarrow} \nabla_{\boldsymbol{\rho}} \mathcal{J}(\boldsymbol{\rho}),$$

which we refer to as "*non-importance-weighted PGPE*" (NIW-PGPE).

Importance sampling [7] is a technique to systematically resolve this distribution mismatch problem. The basic idea of importance sampling is to weight samples drawn from a sampling distribution to match the target distribution, which gives a consistent gradient estimator:

$$\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho}) := \frac{1}{N'}\sum_{n=1}^{N'} w(\boldsymbol{\theta}'_n) \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}'_n|\boldsymbol{\rho}) R(h'_n) \overset{N'\to\infty}{\longrightarrow} \nabla_{\boldsymbol{\rho}} \mathcal{J}(\boldsymbol{\rho}),$$

where

$$w(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}|\boldsymbol{\rho})}{p(\boldsymbol{\theta}|\boldsymbol{\rho}')}$$

is called the *importance weight*.

An intuition behind importance sampling is that if we know how "important" a sample drawn from the sampling distribution is in the target distribution, we can make adjustment by importance weighting. We call this extended method *importance-weighted PGPE* (IW-PGPE).

Now we analyze the variance of gradient estimates in IW-PGPE. For a multi-dimensional space, we consider the *trace* of the covariance matrix of gradient vectors. That is, for a random vector $\boldsymbol{A} = (A_1, \ldots, A_\ell)^\top$, we define

$$\begin{aligned}\mathbf{Var}(\boldsymbol{A}) &= \mathrm{tr}\left(\mathbb{E}\left[(\boldsymbol{A} - \mathbb{E}[\boldsymbol{A}])(\boldsymbol{A} - \mathbb{E}[\boldsymbol{A}])^\top\right]\right), \\ &= \sum_{m=1}^{\ell} \mathbb{E}\left[(A_m - \mathbb{E}[A_m])^2\right], \end{aligned} \tag{4}$$

where $\mathbb{E}$ denotes the expectation.

Let

$$B = \sum_{i=1}^{\ell} \tau_i^{-2},$$

where ℓ is the dimensionality of the basis function vector $\boldsymbol{\phi}(\boldsymbol{s})$. For a $\boldsymbol{\rho} = (\boldsymbol{\eta}, \boldsymbol{\tau})$, we have the following theorem[3]:

Theorem 1. *Assume that for all $\boldsymbol{s}$, a, and $\boldsymbol{s}'$, there exists $\beta > 0$ such that $r(\boldsymbol{s}, a, \boldsymbol{s}') \in [-\beta, \beta]$, and, for all $\boldsymbol{\theta}$, there exists $0 < w_{\max} < \infty$ such that $0 < w(\boldsymbol{\theta}) \leq w_{\max}$. Then we have the following upper bounds:*

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \leq \frac{\beta^2 (1-\gamma^T)^2 B}{N'(1-\gamma)^2} w_{\max},$$
$$\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \leq \frac{2\beta^2 (1-\gamma^T)^2 B}{N'(1-\gamma)^2} w_{\max}.$$

Theorem 1 shows that the upper bound of the variance of $\nabla_{\boldsymbol{\eta}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})$ is proportional to β^2 (the upper bound of squared rewards), $w_{\max}$ (the upper bound of the importance weight $w(\boldsymbol{\theta})$), B (the trace of the inverse Gaussian covariance), and $(1-\gamma^T)^2/(1-\gamma)^2$, and is inverse-proportional to sample size N'. It is interesting to see that the upper bound of the variance of $\nabla_{\boldsymbol{\tau}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})$ is twice larger than that of $\nabla_{\boldsymbol{\eta}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})$.

It is also interesting to see that the upper bounds are the same as the upper bounds for the plain PGPE (Theorem 1 of [32]) except for the factor w_{max}; when $w_{\max} = 1$, the bounds are reduced to those of the plain PGPE method. However, if the sampling distribution is significantly different from the target distribution, $w_{\max}$ can take a large value and thus IW-PGPE tends to produce a gradient estimator with large variance (at least in terms of its upper bound). Therefore, IW-PGPE may not be a reliable approach as it is.

Below, we give a variance reduction technique for IW-PGPE, which leads to a highly effective policy gradient algorithm.

3.2 Variance Reduction by Baseline Subtraction for IW-PGPE

To cope with the large variance of gradient estimates in IW-PGPE, several techniques have been developed in the context of sample reuse, for example, by flattening [9], truncating [28], and normalizing [21] the importance weight. Indeed, from Theorem 1, we can see that decreasing $w_{\max}$ by flattening or truncating the importance weight reduces the upper bounds of the variance of gradient estimates. However, all of those techniques are based on the bias-variance trade-off, and thus they lead to biased estimators.

Another, and possibly more promising variance reduction technique is subtraction of a constant *baseline* [24, 30, 8, 29], which reduces the variance *without* increasing the bias.

[3] Proofs of all theorems are provided in Appendix, which are basically extensions of the proofs for the plain PGPE given in [32] to importance-weighting scenarios.

Here, we derive an optimal baseline for IW-PGPE to minimize the variance, and analyze its theoretical properties.

A policy gradient estimator with a baseline $b \in \mathbb{R}$ is defined as

$$\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^{b}_{\mathrm{IW}}(\boldsymbol{\rho}) := \frac{1}{N'} \sum_{n=1}^{N'} (R(h'_n) - b) w(\boldsymbol{\theta}'_n) \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}'_n | \boldsymbol{\rho}).$$

It is well known that $\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^{b}_{\mathrm{IW}}(\boldsymbol{\rho})$ is still a consistent estimator of the true gradient for any constant b [8]. Here, we determine the constant baseline b so that the variance is minimized, following the line of [32]. Let b^* be the optimal constant baseline for IW-PGPE that minimizes the variance:

$$b^* := \arg\min_{b} \mathbf{Var}[\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^{b}_{\mathrm{IW}}(\boldsymbol{\rho})].$$

Then the following theorem gives the optimal constant baseline for IW-PGPE:

Theorem 2. *The optimal constant baseline for IW-PGPE is given by*

$$b^* = \frac{\mathbb{E}_{p(h,\boldsymbol{\theta}|\boldsymbol{\rho}')}[R(h) w^2(\boldsymbol{\theta}) \|\nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})\|^2]}{\mathbb{E}_{p(h,\boldsymbol{\theta}|\boldsymbol{\rho}')}[w^2(\boldsymbol{\theta}) \|\nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})\|^2]},$$

and the excess variance for a constant baseline b is given by

$$\mathbf{Var}[\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^{b}_{\mathrm{IW}}(\boldsymbol{\rho})] - \mathbf{Var}[\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})] = \frac{(b - b^*)^2}{N'} \mathbb{E}_{p(h,\boldsymbol{\theta}|\boldsymbol{\rho}')}[w^2(\boldsymbol{\theta}) \|\nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})\|^2],$$

where $\mathbb{E}_{p(h,\boldsymbol{\theta}|\boldsymbol{\rho}')}[\cdot]$ denotes the expectation of the function of random variables h and $\boldsymbol{\theta}$ with respect to $(h, \boldsymbol{\theta}) \sim p(h, \boldsymbol{\theta}|\boldsymbol{\rho}')$.

The above theorem gives an analytic expression of the optimal constant baseline for IW-PGPE. It also shows that the excess variance is proportional to the squared difference of baselines $(b - b^*)^2$ and the expectation of the product of squared importance weight $w(\boldsymbol{\theta})$ and the squared norm of characteristic eligibility $\|\nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})\|^2$, and is inverse-proportional to sample size N'.

Next, we analyze contributions of the optimal baseline to variance reduction in IW-PGPE:

Theorem 3. *Assume that for all $\boldsymbol{s}$, a, and $\boldsymbol{s}'$, there exists $\alpha > 0$ such that $r(\boldsymbol{s}, a, \boldsymbol{s}') \geq \alpha$, and, for all $\boldsymbol{\theta}$, there exists $w_{\min} > 0$ such that $w(\boldsymbol{\theta}) \geq w_{\min}$. Then we have the following lower bounds:*

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\eta}} \widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \geq \frac{\alpha^2 (1 - \gamma^T)^2 B}{N'(1 - \gamma)^2} w_{\min},$$

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}} \widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\tau}} \widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \geq \frac{2\alpha^2 (1 - \gamma^T)^2 B}{N'(1 - \gamma)^2} w_{\min}.$$

Assume that for all $\boldsymbol{s}$, a, *and* $\boldsymbol{s}'$, *there exists* $\beta > 0$ *such that* $r(\boldsymbol{s}, a, \boldsymbol{s}') \in [-\beta, \beta]$, *and, for all* $\boldsymbol{\theta}$, *there exists* $0 < w_{\max} < \infty$ *such that* $0 < w(\boldsymbol{\theta}) \leq w_{\max}$. *Then we have the following upper bounds:*

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \leq \frac{\beta^2(1-\gamma^T)^2B}{N'(1-\gamma)^2}w_{\max},$$
$$\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \leq \frac{2\beta^2(1-\gamma^T)^2B}{N'(1-\gamma)^2}w_{\max}.$$

This theorem shows that the bounds of the variance reduction in IW-PGPE brought by the optimal constant baseline depend on the bounds of the importance weight. If importance weights are larger, using the optimal baseline can reduce the variance more.

Based on Theorems 1 and 3, we get the following corollary:

Corollary 4. *Assume that for all* $\boldsymbol{s}$, a, *and* $\boldsymbol{s}'$, *there exists* $0 < \alpha < \beta$ *such that* $r(\boldsymbol{s}, a, \boldsymbol{s}') \in [\alpha, \beta]$, *and, for all* $\boldsymbol{\theta}$, *there exists* $0 < w_{\min} < w_{\max} < \infty$ *such that* $w_{\min} \leq w(\boldsymbol{\theta}) \leq w_{\max}$. *Then we have the following upper bounds:*

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \leq \frac{(1-\gamma^T)^2B}{N'(1-\gamma)^2}(\beta^2 w_{\max} - \alpha^2 w_{\min}),$$
$$\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \leq \frac{2(1-\gamma^T)^2B}{N'(1-\gamma)^2}(\beta^2 w_{\max} - \alpha^2 w_{\min}).$$

Comparing Theorem 1 and this corollary, we can see that the upper bounds for IW-PGPE with the optimal constant baseline are smaller than those for IW-PGPE with no baseline because $\alpha^2 w_{\min} > 0$. Although they are just upper bounds, they can still intuitively show that subtraction of the optimal constant baseline contributes to mitigating the large variance caused by importance weighting. If $w_{\min}$ is larger, then the upper bounds for IW-PGPE with the optimal constant baseline can be much smaller than those for IW-PGPE with no baseline.

4 Experimental Results

In this section, we experimentally investigate the usefulness of the proposed method, importance-weighted PGPE with the optimal constant baseline (which we denote by IW-PGPE$_{\mathrm{OB}}$ hereafter). In the experiments, we estimate the optimal constant baseline using all collected data, as suggested in [8, 16, 29]. This approach introduces bias into the method because the same sample-set is used both for estimating the gradient and the baseline. Another possibility is to split the data into two parts: One is used for estimating the optimal constant baseline and the other is used for estimating the gradient. However, we found that this splitting approach does not work well in our preliminary experiments. The MATLAB implementation of IW-PGPE$_{\mathrm{OB}}$ is available from: `http://sugiyama-www.cs.titech.ac.jp/~tingting/software.html`.

4.1 Illustrative Example

First, we illustrate the behavior of PGPE methods using a toy dataset.

4.1.1 Setup

The dynamics of the environment is defined as

$$s_{t+1} = s_t + a_t + \varepsilon,$$

where $s_t \in \mathbb{R}$, $a_t \in \mathbb{R}$, and $\varepsilon \sim \mathcal{N}(0, 0.5^2)$ is stochastic noise. The initial state s_1 is randomly chosen from the standard normal distribution. The linear deterministic controller is represented by $a_t = \theta s_t$ for $\theta \in \mathbb{R}$. The immediate reward function is given by

$$r(s_t, a_t) = \exp\left(-s_t^2/2 - a_t^2/2\right) + 1,$$

which is bounded in $(1, 2]$. In the toy dataset experiments, we always set the discount factor at $\gamma = 0.9$, and we always use the adaptive learning rate $\varepsilon = 0.1/\|\nabla_\rho \hat{\mathcal{J}}(\boldsymbol{\rho})\|$ [11].

Here, we compare the following PGPE methods:

- **PGPE:** Plain PGPE without data reuse [20].
- **PGPE$_{\mathbf{OB}}$:** Plain PGPE with the optimal constant baseline without data reuse [32].
- **NIW-PGPE:** Data-reuse PGPE without importance weights.
- **NIW-PGPE$_{\mathbf{OB}}$:** Data-reuse PGPE$_{\mathrm{OB}}$ without importance weights.
- **IW-PGPE:** Importance-weighted PGPE.
- **IW-PGPE$_{\mathbf{OB}}$:** Importance-weighted PGPE with the optimal baseline.

Suppose that a small amount of samples consisting of N trajectories with length T is available at each iteration. More specifically, given the hyper-parameter $\boldsymbol{\rho}_L = (\eta_L, \tau_L)$ at the Lth iteration, we first choose the policy parameter θ_n^L from $p(\theta|\boldsymbol{\rho}_L)$, and then run the agent to generate trajectory h_n^L according to $p(h|\theta_n^L)$. Initially, the agent starts from a randomly selected state s_1 following the initial state probability density $p(s_1)$ and chooses an action based on the policy $p(a_t|s_t, \theta_n^L)$. Then the agent makes a transition following the dynamics of the environment $p(s_{t+1}|s_t, a_t)$ and receives a reward $r_t = r(s_t, a_t, s_{t+1})$. The transition is repeated T times to get a trajectory, which is denoted as $h_n^L = \{s_t, a_t, r_t, s_{t+1}\}_{t=1}^T$. We repeat the procedure N times, and, the samples gathered at the Lth iteration is obtained, which is expressed as $D^L = \{(\theta_n^L, h_n^L)\}_{n=1}^N$.

In the data-reuse methods, we estimate gradients at each iteration based on the current data and all previously collected data $D^{1:L} = \{D^l\}_{l=1}^L$, by the estimated gradients to update the policy hyper-parameters (i.e., mean η and standard deviation τ). In the plain PGPE method and the plain PGPE$_{\mathrm{OB}}$ method, we only use the on-policy data D^L to estimate the gradients at each iteration, by the estimated gradients to update the policy

hyper-parameters. If the deviation parameter τ takes a value smaller than 0.05 during the parameter-update process, we set it at 0.05.

Below, we experimentally evaluate the variance, bias, and mean squared error of the estimated gradients, trajectories of learned hyper-parameters, and obtained returns.

4.1.2 Estimated Gradients

We investigate how data reuse influences estimated gradients over iterations. Below, we focus on gradients with respect to the mean parameter η.

We randomly choose initial mean parameter η from the standard normal distribution, and fix the initial deviation parameter at $\tau = 1$. We collect $N = 10$ trajectories with the trajectory length $T = 10$ at each iteration, and update hyper-parameters over 20 iterations. Here, the variance and squared bias of estimated gradients at each iteration (e.g., at the Lth iteration, $L = 1, \ldots, 20$) are investigated for $M = 10000$ trials:

$$\mathrm{Var} := \frac{1}{M}\sum_{m=1}^{M}\left\|\nabla_{\eta_L}\widehat{\mathcal{J}}^m(\boldsymbol{\rho}_L) - \frac{1}{M}\sum_{m'=1}^{M}\nabla_{\eta_L}\widehat{\mathcal{J}}^{m'}(\boldsymbol{\rho}_L)\right\|^2,$$

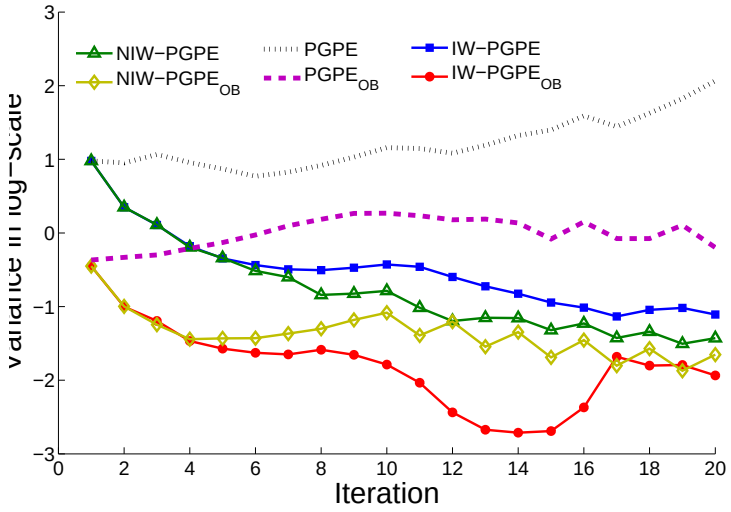
$$\mathrm{Bias}^2 := \left\|\frac{1}{M}\sum_{m=1}^{M}\nabla_{\eta_L}\widehat{\mathcal{J}}^m(\boldsymbol{\rho}_L) - \nabla_{\eta_L}\mathcal{J}(\boldsymbol{\rho}_L)\right\|^2,$$

where $\nabla_{\eta_L}\widehat{\mathcal{J}}^m(\boldsymbol{\rho}_L)$ is an estimated gradient in the m-th trial. More specifically, we estimate the gradients M times with different random seeds at the Lth iteration as follows: We generate samples $D_m^{1:L} = \{D_m^l\}_{l=1}^L$ following the corresponding distributions $\{D_m^l \overset{i.i.d}{\sim} p(h,\theta|\boldsymbol{\rho}_l)\}_{l=1}^L$ in each trial ($m = 1, \ldots, M$), and we estimate the gradient $\nabla_{\eta_L}\widehat{\mathcal{J}}^m(\boldsymbol{\rho}_L)$ with the generated samples $D_m^{1:L}$. The variance and squared bias at the Lth iteration are calculated based on the estimated gradients from M trials. In this experiment, the true gradient $\nabla_{\eta_L}\mathcal{J}(\boldsymbol{\rho}_L)$ at the Lth iteration is approximated by the plain PGPE method using Eq.(3) with $N = 10000$ on-policy samples. Note that the sum of the variance and squared bias agrees with the mean squared error:

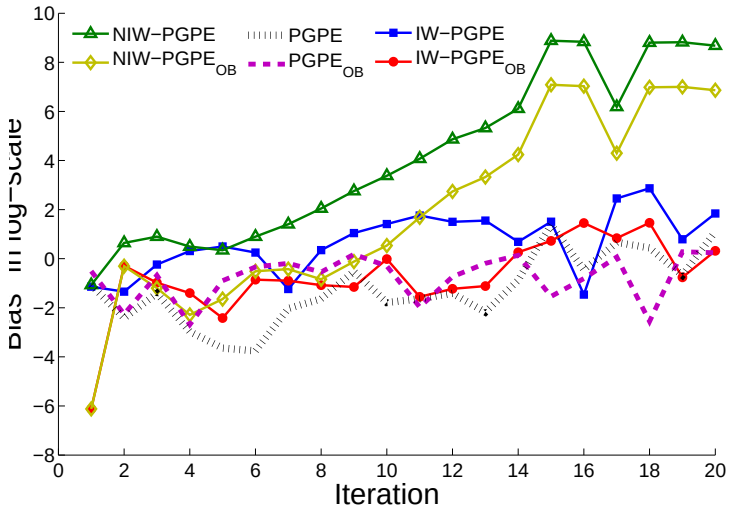
$$\mathrm{Var} + \mathrm{Bias}^2 = \frac{1}{M}\sum_{m=1}^{M}\|\nabla_{\eta_L}\widehat{\mathcal{J}}^m(\boldsymbol{\rho}_L) - \nabla_{\eta_L}\mathcal{J}(\boldsymbol{\rho}_L)\|^2. \tag{5}$$

We update the hyper-parameters $\boldsymbol{\rho}_L$ based on the estimated true gradient $\nabla_{\eta_L}\mathcal{J}(\boldsymbol{\rho}_L)$, and obtain $\boldsymbol{\rho}_{L+1}$. Then, we investigate the variance and bias at the next iteration, i.e., the $(L+1)$th iteration, following the above procedures. Figure 1 shows the variance and squared bias over 20 iterations.

From Figure 1(a), we can see that IW-PGPE$_{\mathrm{OB}}$ provides gradient estimates with the lowest variance among the compared methods. IW-PGPE has a larger variance than NIW-PGPE, which well agrees with our theoretical analysis: According to Theorem 1, upper bounds of the variance are proportional to the importance weight, which is always 1 in NIW-PGPE, but is very large in IW-PGPE if the target distribution is significantly



(a) Variance

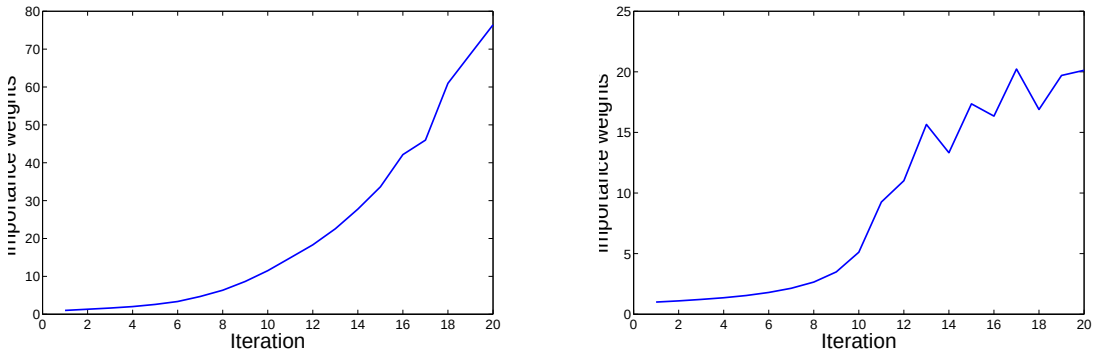


(b) Bias^2

Figure 1: Variance and Bias^2 of gradient estimates with respect to the mean parameter η through parameters update iterations.

different from the sampling distribution. In order to see whether the upper bound of importance weights is really large, we measure the maximum value of importance weights over iterations, which is shown in Figure 2. Figure 2(a) shows that the maximum value of importance weights tends to be larger over iterations, which further illustrates how importance weights influence the variance of gradient estimates in IW-PGPE.

We can also see that the gap in the variance between IW-PGPE and IW-PGPE$_{\mathrm{OB}}$ tends to be larger over iterations, which is also consistent with our theoretical analysis:



(a) IW-PGPE (b) IW-PGPE$_{\mathrm{OB}}$

Figure 2: Average maximum values of importance weights over 20 runs through parameter update iterations.

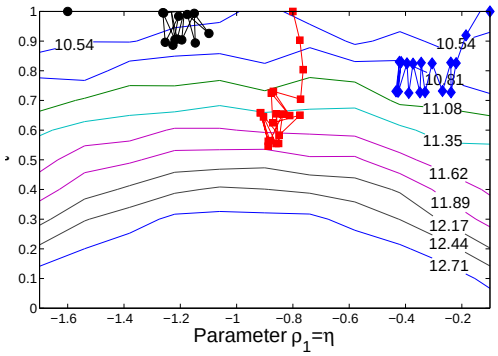
According to Theorem 3, the larger the importance weight is, the more the optimal constant baseline contributes to reducing the variance. The importance weight may get larger at later iterations, because distributions in the first and the last iterations may be significantly different (Figure 2 exactly illustrates this phenomenon). Thus, variance reduction from IW-PGPE to IW-PGPE$_{\mathrm{OB}}$ by the optimal constant baseline tends to be more significant in later iterations. Gradient estimates in both NIW-PGPE$_{\mathrm{OB}}$ and IW-PGPE$_{\mathrm{OB}}$ are with smaller variance than the plain PGPE$_{\mathrm{OB}}$ method, because the more data we use, the smaller variance of gradient estimates we can obtain as expected from the theory. IW-PGPE$_{\mathrm{OB}}$ provides smaller variance than NIW-PGPE$_{\mathrm{OB}}$, which is our expected result: According to Theorem 3, if the importance weights are larger, using the optimal constant baseline can reduce variance more, while the importance weights are always 1 in NIW-PGPE$_{\mathrm{OB}}$ (see Figure 2(b)). The plain PGPE$_{\mathrm{OB}}$ has smaller variance than the plain PGPE, which well agrees with the results reported in [32].

Figure 1(b) shows that introduction of the optimal baseline does not increase the bias. NIW-PGPE and NIW-PGPE$_{\mathrm{OB}}$ have very large bias, because naively reusing previous data leads to an inconsistent and biased gradient estimator. The bias of gradient estimates in IW-PGPE is fairly small, because IW-PGPE is not only consistent, but also unbiased. The plain PGPE and plain PGPE$_{\mathrm{OB}}$ are also with small bias, as expected.

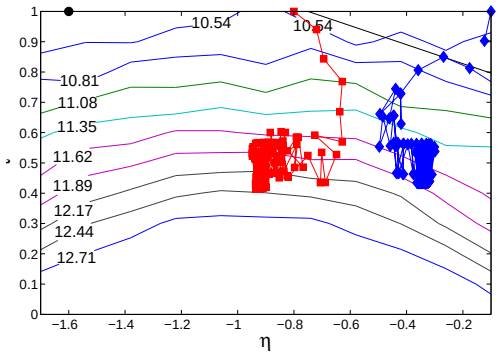
Because our proposed IW-PGPE$_{\mathrm{OB}}$ has small bias and the smallest variance among the compared methods, it also gives the smallest mean squared error (see Eq.(5)).

4.1.3 Hyper-Parameter Trajectories

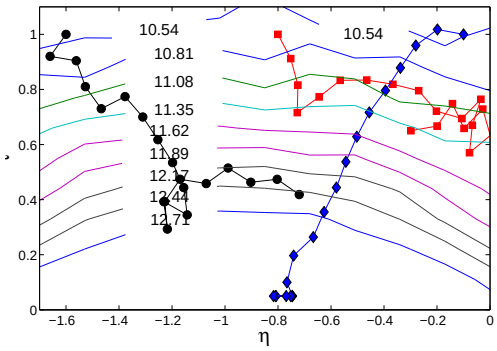
Next, we illustrate how learned hyper-parameters change over iterations. Here we compare the behavior of the following three methods: NIW-PGPE, IW-PGPE and our proposed method IW-PGPE$_{\mathrm{OB}}$. We fix the initial deviation parameter at $\tau = 1$, and test the three different initial mean parameters: $\eta = -1.6$, -0.8, and -0.1. Figure 3 depicts the contour of the expected return, where the maximum of the return surface is located at the middle



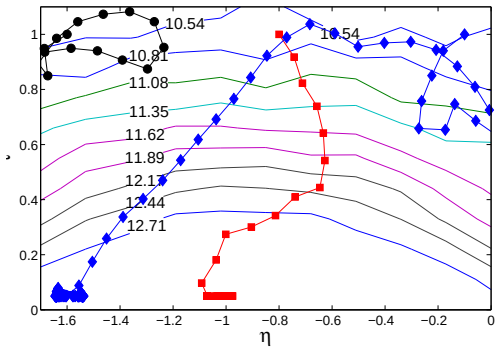
(a) NIW-PGPE ($N = 10$)



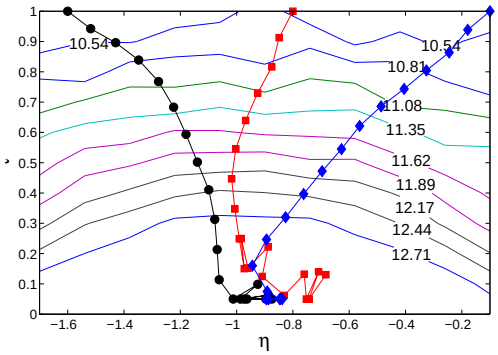
(b) NIW-PGPE ($N = 1$)



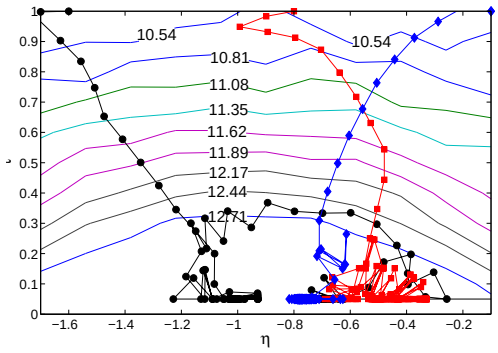
(c) IW-PGPE ($N = 10$)



(d) IW-PGPE ($N = 1$)



(e) IW-PGPE$_{\mathrm{OB}}$ ($N = 10$)



(f) IW-PGPE$_{\mathrm{OB}}$ ($N = 1$)

Figure 3: Trajectories of policy hyper-parameters over iterations.

bottom.

First, let us investigate how the hyper-parameters change over 20 iterations in a large-sample case with $N = 10$. From Figure 3(a), we can see that NIW-PGPE can not properly update the solutions, which means that the inconsistency can not be overcome by increasing the number of samples. On the other hand, Figure 3(c) shows that IW-PGPE can lead the solutions to an area with large returns sometimes, but can not always reach an area with large returns after 20 iterations. This indicates that the consistency of importance weighting tends to be helpful when the number of samples is large, but it can not converge rapidly because of the large variance. Figure 3(e) shows that IW-PGPE$_{\mathrm{OB}}$ gives the reliable update directions and the three paths converge rapidly to the vicinity of the maximum point without detours. This shows that the optimal constant baseline highly contributes to improving the convergence property of IW-PGPE.

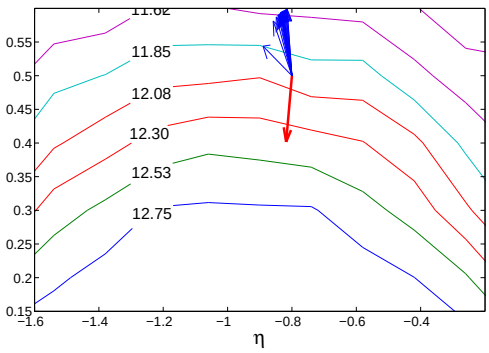
Next, we investigate the performance over 200 iterations with only $N = 1$. Figure 3(b) shows that NIW-PGPE can not properly update the solutions to the maximum point because of the inconsistency, and Figure 3(d) shows that the IW-PGPE solutions can not always reach an area with large returns (middle bottom) after 200 iterations, which is because the variance in IW-PGPE is crucial in this extreme scenario. However, Figure 3(e) shows that the proposed IW-PGPE$_{\mathrm{OB}}$ can still find fairly reliable update directions with only $N = 1$.

Next, we investigate the directions of estimated gradients more systematically. We fix the starting point at $\eta = -0.8$ and $\tau = 0.5$. The true gradient direction is calculated by the plain PGPE method with 10000 on-policy samples. In this experiment, we first collect $N' = 10$ off-policy samples, which are drawn from $\mathcal{N}(-1.6, 1)$. We then reuse these off-policy samples to estimate the gradients in the data-reuse methods. We calculate the gradients 20 times with different random seeds, and investigate the angle between the true gradient and the estimated gradients. The results are summarized in Figure 4. In Figure 4(a), the red line denotes the true gradient and blue lines are the estimated gradients by the NIW-PGPE method. The histograms of angles between the true gradient and the estimated gradients are plotted in Figure 4(b). The graph shows that the angles are concentrated in $[-150, -90]$, which further explains the inconsistent property of the NIW-PGPE method. Observing the angle distribution for IW-PGPE in Figure 4(d), we can see that the angles are widely distributed in $[-180, 180]$, which clearly illustrates the large variance problem of IW-PGPE. On the other hand, the angles for the IW-PGPE$_{\mathrm{OB}}$ method are concentrated in $[-60, 60]$, which highlights the small variance and consistent properties of IW-PGPE$_{\mathrm{OB}}$.

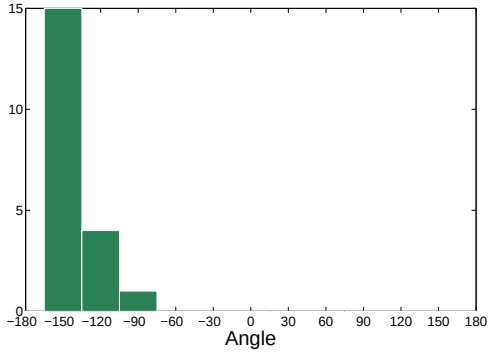
4.1.4 Performance of Learned Policies

Finally, we evaluate average expected returns obtained by each method over 20 runs. The expected return at each trial is approximated using 100 newly-drawn test episodic data (which are not used for policy learning). The initial mean parameter η is chosen randomly from the standard normal distribution, and the deviation parameter is fixed at $\tau = 1$.

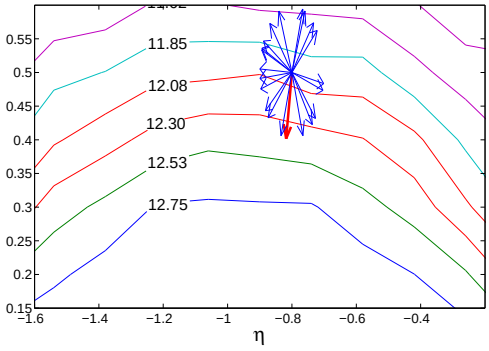
Figure 5 shows that IW-PGPE$_{\mathrm{OB}}$ improves the performance over iterations and con-



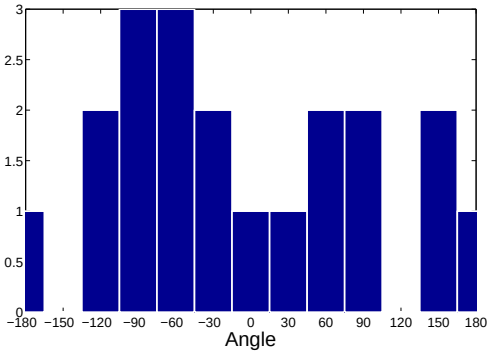
(a) NIW-PGPE



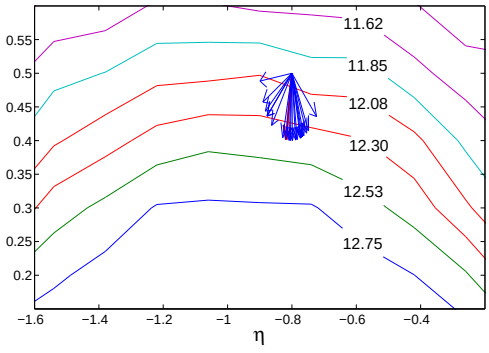
(b) NIW-PGPE



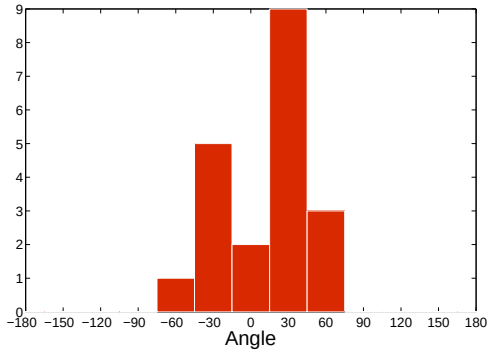
(c) IW-PGPE



(d) IW-PGPE



(e) IW-PGPE$_{\mathrm{OB}}$



(f) IW-PGPE$_{\mathrm{OB}}$

Figure 4: Directions of estimated gradients.

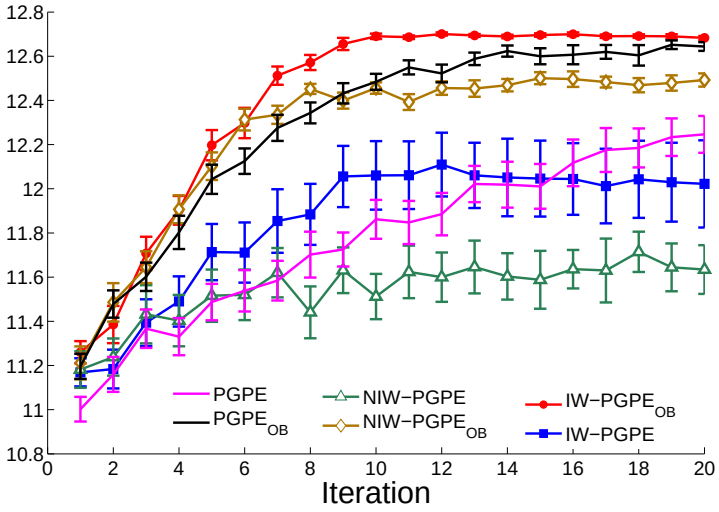


Figure 5: Average expected returns through policy update iterations over 20 runs for toy data. Error bars denote standard errors.

verges very fast. The performance of NIW-PGPE is not largely improved over iterations, which is caused by biased gradient estimates (see Figure 3(a) again). IW-PGPE works better than NIW-PGPE, but the performance is saturated after 9 iterations. IW-PGPE$_{\mathrm{OB}}$ does not outperform NIW-PGPE$_{\mathrm{OB}}$ that much at the first several iterations, because the difference between the target distribution and a sampling distribution is not that large at the beginning. However, the upper bound of importance weights tends to become larger over iterations (see Figure 2(b) again), which makes IW-PGPE$_{\mathrm{OB}}$ more reliable than NIW-PGPE$_{\mathrm{OB}}$ in the latter iterations. The plain PGPE$_{\mathrm{OB}}$ method works fairly well with $N = 10$ on-policy samples, but it is still not as good as IW-PGPE$_{\mathrm{OB}}$.

4.2 Mountain Car

Next, we evaluate our proposed method in the *mountain car* task, which is illustrated in Figure 6. The task consists of a car and two hills whose landscape is described as $\sin(3x)$. The top of the right hill is the goal to which we want to guide the car.

We compare the following 7 methods:

- **TIW-eNAC:** Truncated importance-weight episodic natural actor-critic, which is an episodic version of the sample-reuse NAC method [28, 17]. Following the same line as [28], we truncate the importance weight as $w = \min\{w, 2\}$.
- **IW-REINFORCE$_{\mathrm{OB}}$:** Importance-weighted REINFORCE with the optimal baseline, which is basically a combination of the off-policy implementation of the episodic REINFORCE method [12] and the optimal baseline [16], although we could not exactly find this method in literature.

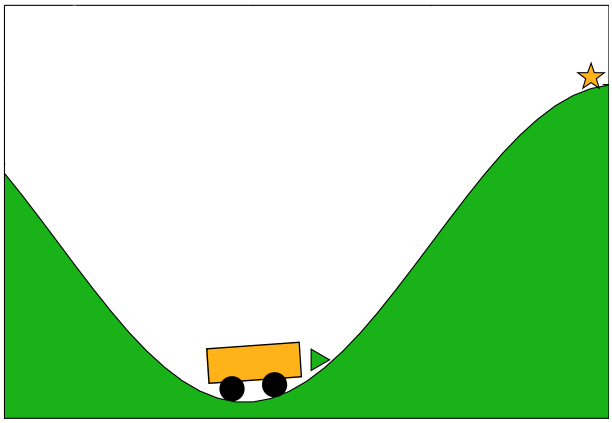


Figure 6: Mountain car.

- **R^3:** Reward-weighted regression with sample reuse [9].
- **PGPE$_{\mathbf{OB}}$:** Plain PGPE$_{\mathrm{OB}}$ without data reuse.
- **NIW-PGPE$_{\mathbf{OB}}$:** Data-reuse PGPE$_{\mathrm{OB}}$ without importance weighting.
- **IW-PGPE:** Importance-weighted PGPE.
- **IW-PGPE$_{\mathbf{OB}}$:** Importance-weighted PGPE with the optimal baseline.

The state space $\mathcal{S}$ is two-dimensional and continuous, which consists of the horizontal position $x[m] \in [-1.2, 0.5]$ and the velocity $\dot{x}[m/s] \in [-1.5, 1.5]$, i.e., $\boldsymbol{s} = (x, \dot{x})^\top$. This is non-linearly transformed to a feature space via a basis function vector $\boldsymbol{\phi}(\boldsymbol{s})$. We use 12 Gaussian kernels with mean $\boldsymbol{c}$ and standard deviation $\kappa = 1$ as the basis functions,

$$\boldsymbol{\phi}(\boldsymbol{s}) = \exp\left(-\frac{\|\boldsymbol{s} - \boldsymbol{c}\|^2}{2\kappa^2}\right),$$

where the kernel centers $\boldsymbol{c}$ are distributed over the following grid points:

$$\{-1.2, -0.35, 0.5\} \times \{-1.5, -0.5, 0.5, 1.5\}.$$

The action space $\mathcal{A}$ is one-dimensional and continuous, which corresponds to the force applied to the car (note that the force of the car is not strong enough to climb up the slope to directly reach the goal). We use the Gaussian policy model for IW-REINFORCE$_{\mathrm{OB}}$, TIW-eNAC, and R^3:

$$p(a|\boldsymbol{s}, \boldsymbol{\theta}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(a - \boldsymbol{\mu}^\top \boldsymbol{\phi}(\boldsymbol{s}))^2}{2\sigma^2}\right), \tag{6}$$

where $\boldsymbol{\mu}$ is the mean policy parameter and σ is the deviation policy parameter. We employ a linear deterministic policy model (1) for the PGPE methods, which corresponds to Eq.(6) with $\sigma \to 0$.

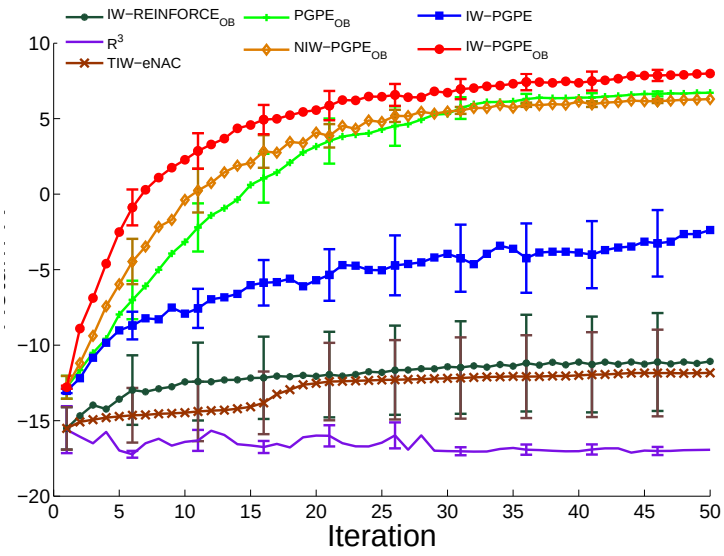


Figure 7: Average expected returns over 10 runs as functions of the number of iterations for the mountain-car task. Error bars are standard errors.

The dynamics of the car (i.e., the update rules of the position and the velocity) are given by

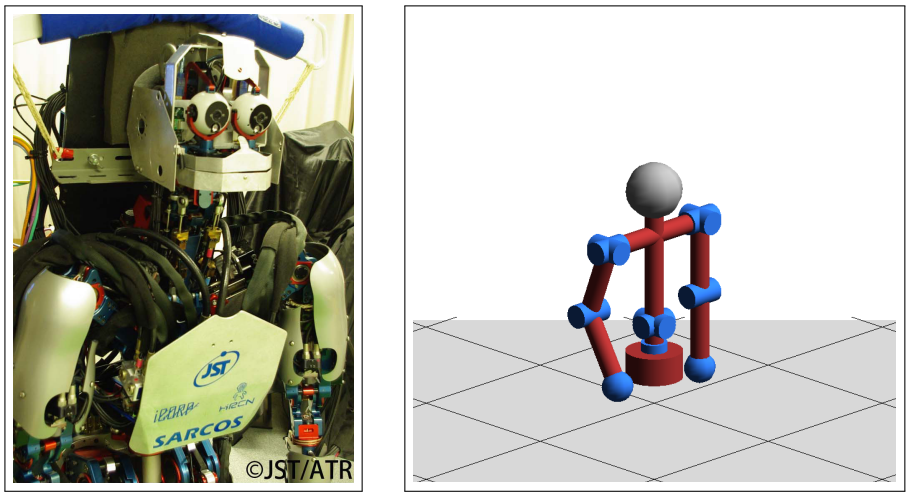
$$x_{t+1} = x_t + \dot{x}_{t+1}\Delta t,$$
$$\dot{x}_{t+1} = \dot{x}_t + (-9.8w\cos(3x_t) + \frac{a_t}{w} - k\dot{x}_t)\Delta t,$$

where a_t is the action taken at time t. We set the problem parameters as follows: The mass of the car $w = 0.2$[kg], the friction coefficient $k = 0.3$, and the simulation time step $\Delta t = 0.1$[s]. The reward function is defined as

$$r(\boldsymbol{s}_t, a_t, \boldsymbol{s}_{t+1}) = \begin{cases} 1 & \text{if } x_{t+1} \geq 0.45, \\ -1 & \text{otherwise.} \end{cases}$$

The initial mean parameter $\boldsymbol{\eta}$ is chosen randomly from the standard normal distribution, and the initial deviation parameter is set at $\tau = 1$. The initial state of the car is set at the bottom of the mountain with the velocity $\dot{x} = 0$. The agent collects $N = 10$ episodic samples with trajectory length $T = 40$ at each iteration. In the data reuse methods, we reuses all previous data at later iterations. In the plain PGPE$_{\mathrm{OB}}$ method, we just use $N = 10$ on-policy samples at each iteration to estimate policy gradients. The discount factor is set at $\gamma = 0.95$. The learning rate is $\varepsilon = 1/\|\nabla_\rho \hat{\mathcal{J}}(\boldsymbol{\rho})\|$.

We investigate average expected returns over 10 trials as functions of policy-update iterations. The expected return at each trial is computed over 100 newly-drawn test episodic samples (which are not used for policy learning). The experimental results are plotted in Figure 7. This shows that IW-PGPE$_{\mathrm{OB}}$ improves the performance very fast over policy-update iterations, and it achieves superior performance improvement than



(a) CB-i (b) Simulated upper-body model

Figure 8: Humanoid robot CB-i and its upper-body model.

all other methods. IW-PGPE can also improve the performance over iterations well, implying that the consistency of the IW estimator is useful in this task. However, it is outperformed by the proposed IW-PGPE$_{\mathrm{OB}}$, perhaps because the estimation variance in IW-PGPE is large. NIW-PGPE$_{\mathrm{OB}}$ performs fairly well, which maybe because the bias of policy gradient estimators is not that crucial in this experiment. The plain PGPE$_{\mathrm{OB}}$ can improve the performance throughout the iterations, which indicates that $N = 10$ on-policy samples is enough for this mountain-car task. Other data-reuse methods can improve the performance over iterations, but slowly, and they are outperformed by the compared PGPE methods. IW-REINFORCE$_{\mathrm{OB}}$ outperforms TIW-eNAC, which maybe because the optimal constant baseline contributes significantly in IW-REINFORCE$_{\mathrm{OB}}$ and truncating the importance weights can lead to a larger bias over iterations in TIW-eNAC. R^3 can not improve the performance over iterations. Overall, thanks to the low variance, IW-PGPE$_{\mathrm{OB}}$ achieves smooth and fast policy improvement throughout iterations, and its performance is the best among the compared methods.

4.3 Upper-body Humanoid Control

Finally, we evaluate the performance of our proposed method on a highly nonlinear dynamic control problem of the simulated upper-body model of the humanoid robot *CB-i* [4] (see Figure 8(a)). We use its simulator in our experiments (see Figure 8(b)). The goal is to lead the end-effector of the right arm (right hand) to a target object.

4.3.1 Setup

We compare the performance of the following 4 methods:

- **IW-REINFORCE$_{\mathrm{OB}}$:** Importance-weighted REINFORCE with the optimal base-

line.

- **NIW-PGPE$_{\mathbf{OB}}$:** Data-reuse PGPE$_{\mathrm{OB}}$ without importance weighting.
- **PGPE$_{\mathbf{OB}}$:** Plain PGPE$_{\mathrm{OB}}$ without data reuse.
- **IW-PGPE$_{\mathbf{OB}}$:** Importance-weighted PGPE with the optimal baseline.

The simulation is based on the upper body of the CB-i humanoid robot illustrated in Figure 8(b), which has 9 degrees of freedom corresponding to main joints of the upper body: The shoulder pitch, shoulder roll, elbow pitch of the right arm, shoulder pitch, shoulder roll, elbow pitch of the left arm, waist yaw, torso roll, and torso pitch.

At each time step, the controller receives states from the system and sends out actions. The state space is 18-dimensional, which corresponds to the current angle and the current angular velocity of each joint. The action space is 9-dimensional, which corresponds to the target angle of each joint. Both states and actions are continuous.

The initial positions of the robot and an object are fixed, where the initial position of the robot is set at the state of standing up straight with the arms down, and the position of the target object depends on the task. Note that the position of the target object is only used in the designing of the reward function. The reward function is given by

$$r_t = k_1 \exp(-10 d_t) - k_2 \min\{c_t, 10000\},$$

where $k_1 = 1$, $k_2 = 0.0005$, d_t is the distance between the robot's right hand and the target object at the time step t, and c_t is the sum of control costs for each joint. Note that the results may change with different k_1 and k_2 for the reward function. In order to keep the value of $\exp(-10 d_t)$ and c_t in the reward function to the same order of magnitude, we need to choose k_1 and k_2 reasonably. We use the same policy model as the mountain car experiment, i.e., the linear deterministic policy for PGPE and the Gaussian policy for IW-REINFORCE$_{\mathrm{OB}}$ with the basis function $\phi(\boldsymbol{s}) = \boldsymbol{s}$.

The initial mean parameter η is randomly chosen from the standard normal distribution, and the initial standard deviation parameter τ is set to 1. To evaluate the usefulness of the data reuse methods with a small number of samples, the agent collects only $N = 3$ on-policy samples with trajectory length $T = 100$ at each iteration. In the data reuse methods, we reuse all previous data at later iterations. In the plain PGPE$_{\mathrm{OB}}$, we just use the on-policy samples to estimate the gradients. The discount factor is set at $\gamma = 0.9$, and the learning rate is set at $\varepsilon = 0.1/\|\nabla_\rho \hat{\mathcal{J}}(\boldsymbol{\rho})\|$.

4.3.2 Reaching Task with 2 Degrees of Freedom

First, we investigate the performance on the reaching task with only 2 degrees of freedom. We fix the body of the robot and use only the right shoulder pitch and right elbow pitch. Figure 9 depicts the averaged expected return over 10 trials as a function of the number of iterations. The expected return at each trial is computed from 50 newly-drawn test episodic data (which are not used for policy learning). The graph shows that IW-PGPE$_{\mathrm{OB}}$

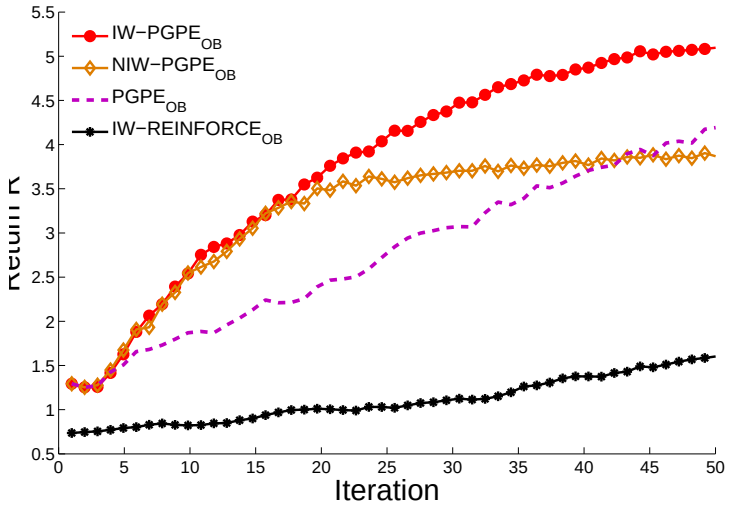
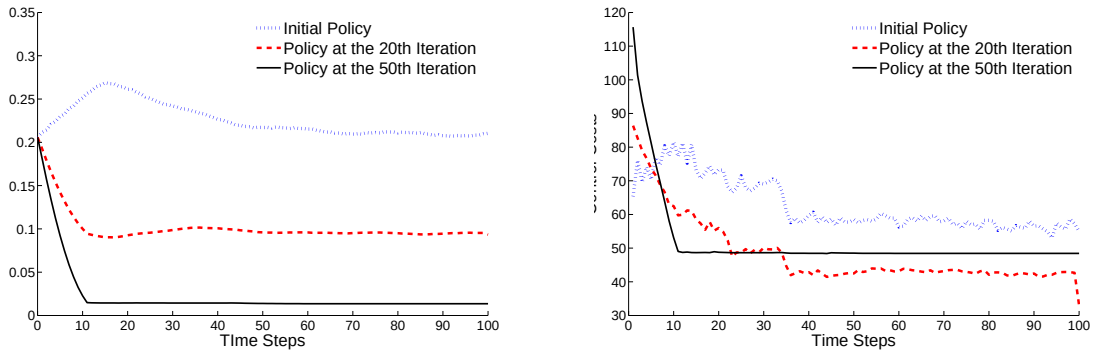


Figure 9: Average expected returns over 10 runs as functions of the number of iterations for the reaching task with 2 degrees of freedom (right shoulder pitch and right elbow pitch).

nicely improves the performance over iterations only with a small number of on-policy samples. The plain PGPE$_{\mathrm{OB}}$ can also improve the performance over iterations, but slowly. NIW-PGPE$_{\mathrm{OB}}$ is not as good as IW-PGPE$_{\mathrm{OB}}$ especially at the later iterations, which is because of the inconsistent property of the NIW estimator. The initial mean parameter is randomly chosen in this experiment, which makes IW-REINFORCE$_{\mathrm{OB}}$ not able to improve the performance significantly over iterations. This result is consistent with the observation that the REINFORCE method is sensitive to the initial parameter values [32].

The distance from the right hand to the object and the control costs along the trajectory are also investigated. We test the initial policy, the policy obtained at the 20th iteration by IW-PGPE$_{\mathrm{OB}}$, and the policy obtained at the 50th iteration by IW-PGPE$_{\mathrm{OB}}$. The results are shown in Figure 10. From Figure 10(a), it is clear to see that the policy obtained at the 50th iteration decreases the distance fastest compared with the initial policy and the policy obtained at the 20th iteration. This means the robot can reach the object fast by using the learned policy. On the other hand, Figure 10(b) shows that the control cost required for executing the policy obtained at the 50th iteration decreases steadily until the reaching task is completed. This is because the robot mainly adjusts the shoulder pitch in the beginning, which consumes a larger amount of energy than the energy required for controlling the elbow pitch. Then, once the right hand gets closer to the target object, the robot starts to adjust the elbow pitch reach the target object. The policy obtained at the 20th iteration actually consumes less control costs, but it cannot move the arm to the target object.

Figure 11 shows a typical solution of the reaching task with 2 degrees of freedom by IW-PGPE$_{\mathrm{OB}}$ (with the policy obtained at the 50th iteration). The images show that the



(a) Distance (b) Control costs

Figure 10: Distance and control costs of arm reaching with 2 degrees of freedom using the policy learned by IW-PGPE$_{\mathrm{OB}}$.

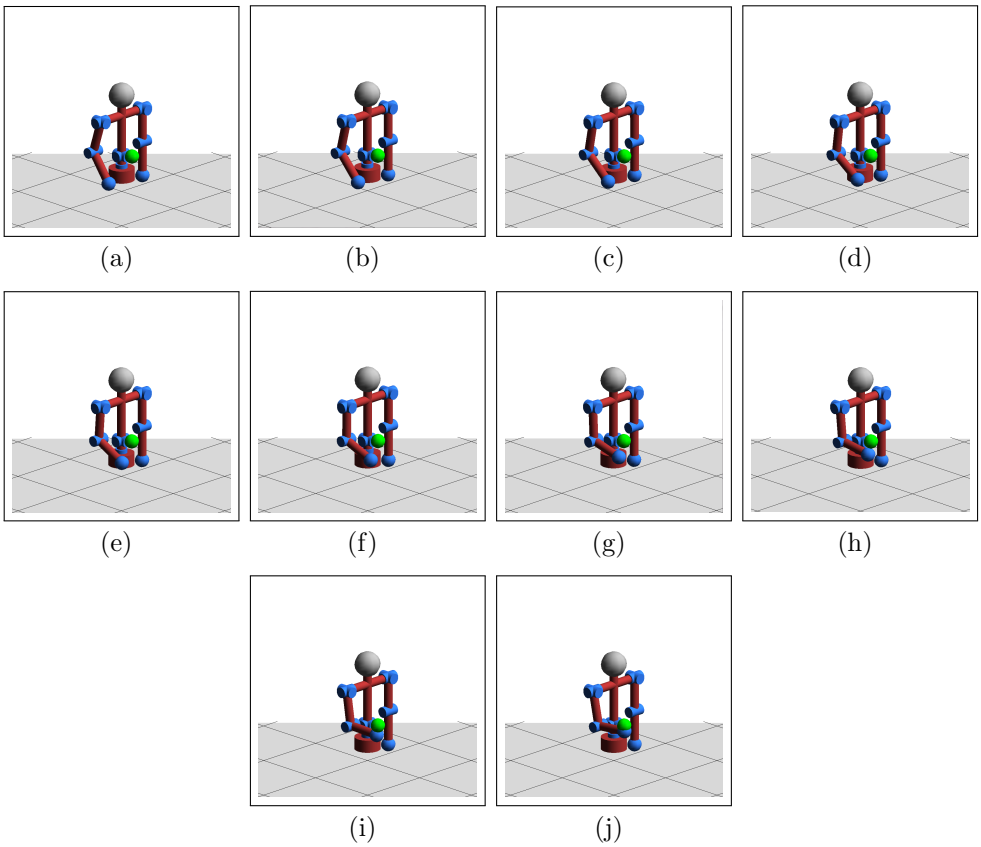


Figure 11: Typical example of arm reaching with 2 degrees of freedom using the policy obtained by IW-PGPE$_{\mathrm{OB}}$ at the 50th iteration.

policy learned by our proposed method successfully leads the right hand to the target object within only 10 time steps.

4.3.3 Reaching Task with 4 Degrees of Freedom

Next, we evaluate the performance on the reaching task with 4 degrees of freedom. We use the right shoulder pitch, right elbow pitch, right shoulder roll, and torso yaw joint. By using the torso yaw joint, the robot can reach a distant object which can not be achieved by only using the right arm. The results are shown in Figure 12. The graph shows that IW-PGPE$_{\mathrm{OB}}$ achieves fast policy improvement throughout iterations, and the performance is the best among the compared methods.

Figure 13 depicts a representative example of object reaching with 4 degrees of freedom by IW-PGPE$_{\mathrm{OB}}$. Note that the object is distant from the robot and it can not be reached by only using the right arm. The robot first adjusts the torso yaw joint, and then uses the right arm to reach the object. The images show that the policy learned by our proposed method successfully leads the right hand to the distant object.

4.3.4 Reaching Task with All Degrees of Freedom

At last, we evaluate the performance on the reaching task with all degrees of freedom. The position of the target object is the same as the task in the 4-degrees-of-freedom setting.

In this experiment, we use all degrees of freedom to reach the object. This increases the dimensionality of the state space, which actually may grow the values of importance weights exponentially [22, 5]. In order to mitigate the large values of importance weights, we decided not to reuse all previously collected samples, but only samples collected in the last 5 iterations. This allows us to keep the difference between the sampling distribution and the target distribution reasonably small, and thus the values of importance weights can be suppressed to some extent. Furthermore, following [28], we truncate the importance weights as $w = \min\{w, 2\}$. This version of IW-PGPE$_{\mathrm{OB}}$ is denoted as Truncated IW-PGPE$_{\mathrm{OB}}$ below.

The results are shown in Figure 14. The graph shows that the performance of Truncated IW-PGPE$_{\mathrm{OB}}$ is the best, which implies that the truncation of importance weights is helpful when applying our proposed method to high-dimensional problems.

Through all the arm-reaching experiments, we can see that the returns tend to be lower as the dimension is increased, even though we run the higher-dimensional experiment for a larger number of iterations. In the task with all degrees of freedom (Figure 14), the largest number of iteration is 400. If we continue the experiment for more iterations, the returns may sligtly increase, but are still less than the returns in the low-dimensional experiments. This is because the more joints the robot uses, the larger energy will be consumed, and thus the returns tend to be lower in high-dimensional cases.

Overall, the proposed IW-PGPE$_{\mathrm{OB}}$ is shown to be a promising method, although in the last experiment it is obvious that just like other importance weight-based methods, the performance degrades in high-dimensional problems without the use of additional correction techniques such as weight truncation.

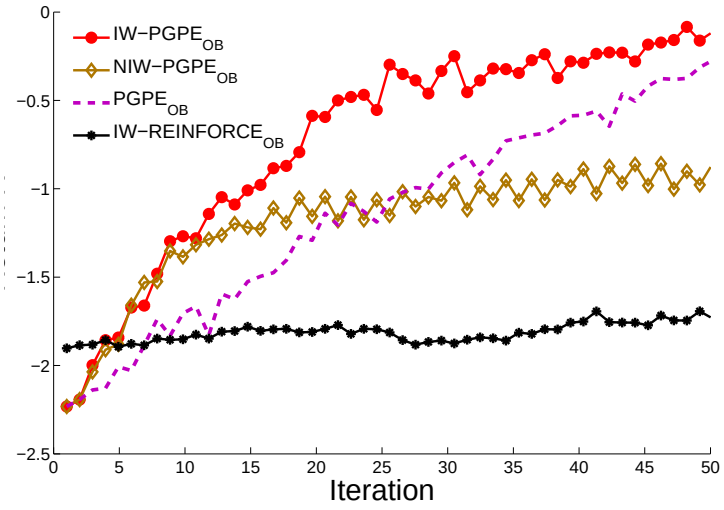


Figure 12: Average expected returns over 10 runs as functions of the number of iterations for the reaching task with 4 degrees of freedom (right shoulder pitch, right elbow pitch, right shoulder roll, and torso yaw joint).

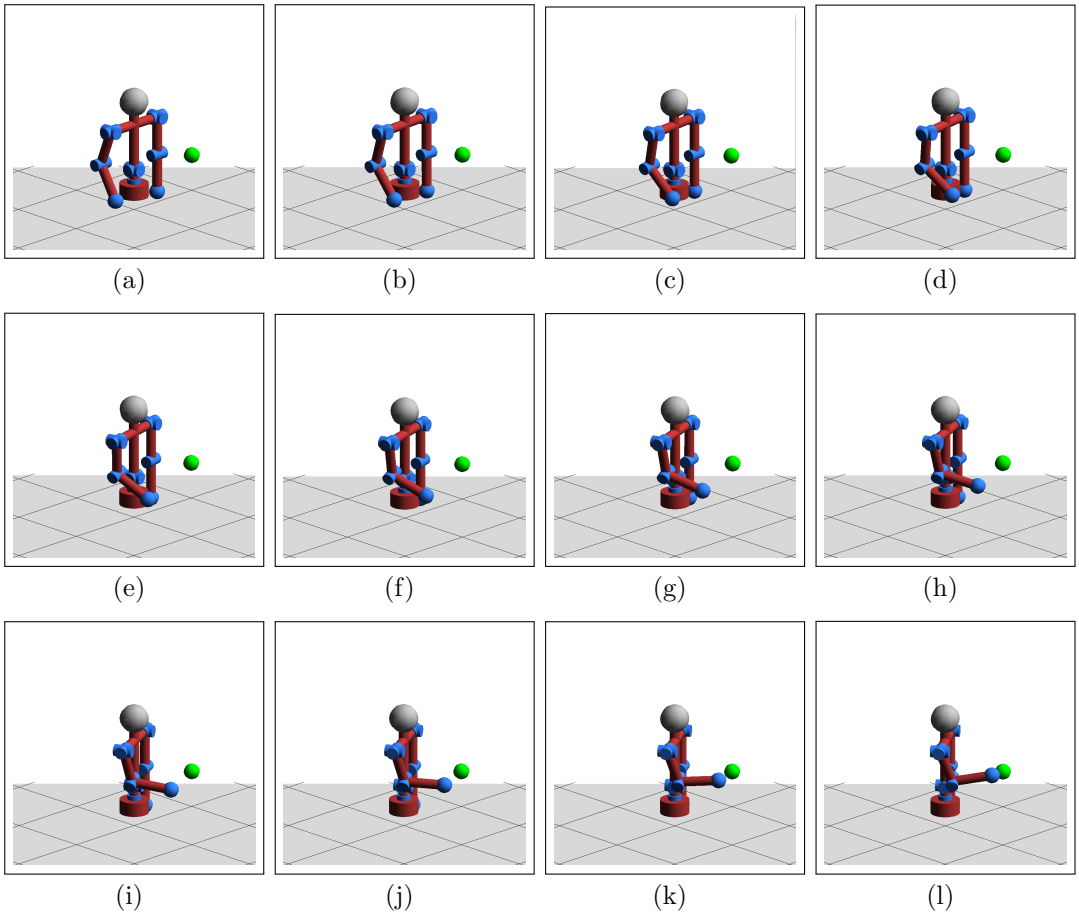


Figure 13: Typical example of arm reaching with 4 degrees of freedom using the policy obtained by IW-PGPE$_{\mathrm{OB}}$ at the 50th iteration.

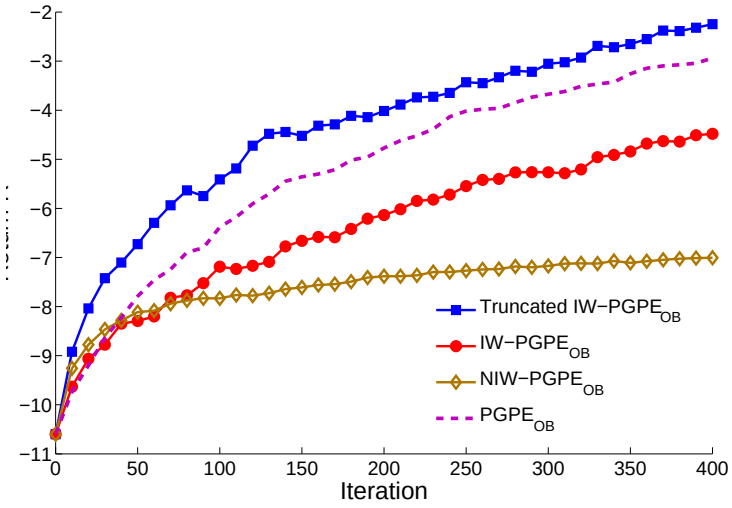


Figure 14: Average expected returns over 10 runs as functions of the number of iterations for the reaching task with all degrees of freedom.

5 Discussions and Conclusions

In many real-world reinforcement learning problems, reducing the number of training samples is desirable because the sampling cost is often much higher than the computational cost. In this paper, we proposed a new policy gradient method equipped with efficient sample reuse, which systematically combines a reliable policy gradient method, PGPE, with importance sampling and the optimal constant baseline. We showed that the introduction of the optimal constant baseline can mitigate the large-variance problem of importance weighting under some conditions. Through experiments with an artificial domain, the usefulness of the proposed method was demonstrated. More over, through robotic experiments, we found that the truncation technique was helpful when applying the proposed method to high-dimensional problems.

The low variance of PGPE was brought by considering a deterministic policy and introducing the stochasticity by drawing a policy parameter from a prior distribution. This per-trajectory formulation was indeed shown to be useful in reducing the variance of policy gradient estimates. However, PGPE has limitations, too. For example, the use of a finite horizon is essential in PGPE, because the gradient estimates need full trajectories. In particular, it is not straightforward to handle the infinite-horizon case. Another issue is an extension to a partially-observable case. It is known that for every finite Markov decision problem (MDP) there exists a deterministic policy that is optimal [19]. However, in a partially-observable MDP (POMDP), the best stationary stochastic policy can be arbitrarily better than the best stationary deterministic policy [23]. Thus, the deterministic policy in PGPE can be a limitation when extending it to the POMDP framework. It is trivial to extend the current formulation to consider stochastic policies. However, this may lead to an increase of variance and thus slow down convergence. These

issues need to be further investigated in the future work.

The baseline and importance weighting techniques are two independent techniques. More specifically, importance weighting is used in the off-policy scenario to efficiently reuse previously collected samples, by using importance weighting the consistency between the data sampling distribution and the target distribution is kept. On the other hand, the optimal constant baseline is used to reduce the variance of gradient estimates.

The use of a baseline technique has been first proposed in terms of reinforcement comparison in [24], which intuitively means the comparison between the expected return R and the baseline b: If $R > b$ we adjust learned parameters ρ so as to increase the probability of θ, and, if $R < b$, we do the opposite. Based on this idea, Williams [30] demonstrated that a baseline technique did not introduce bias, which is because the expectation of the coefficient of b is zero, i.e., $\mathbb{E}\left[\frac{\nabla_\rho p(\theta|\rho)}{p(\theta|\rho)}\right] = 0$. The effect of the baseline on variance is considered in [6]. The intuition behind the baseline is that subtracting a baseline from the return reduces the magnitude, and thus reduces the variance. Technically, subtracting a baseline can be viewed as a *control variate technique* [7], which is an effective approach to reducing variance of Monte Carlo estimates of integrals. The experimental results in the paper suggest that the removal of the baseline is possibly the primary factor in improving performance compared with the importance weighting techniques.

In episodic policy gradient methods, the optimal baseline which does not bias policy gradient estimates is given by a single scalar for all trajectories [16]. However, in the non-episodic policy gradient methods, the optimal baseline can depend on the current state [8, 13, 17]. Thus, if a good parameterization for the baseline is known, e.g., in a generalized linear form $b(s_t) = \boldsymbol{w}^T \boldsymbol{\phi}(s_t)$, this can significantly improve the gradient estimation process. However, the selection of the basis function can be difficult and often impractical in robotics [16]. On the other hand, it is interesting to see that if the value function is used as the baseline function in non-episodic policy gradient methods, such as in [17, 26], the term $Q(s, a) - V(s)$ will lead to the *advantage function* [2], where $Q(s, a)$ is action value function and $V(s)$ is the value function.

Acknowledgements

The authors would like to thank anonymous reviewers for their feedback on our earlier manuscript, which highly contributed to improving the readability of this paper. TZ, VT, JM, and MS were supported by MEXT KAKENHI 23120004. HH was supported by the FIRST program. TZ was also supported by the MEXT scholarship, VT was also supported by the JASSO scholarship, and JM was also supported by the SRBPS and MEXT.

Appendix

In the appendix, we give proofs of the theorems.

A Proof of Theorem 1

Proof. Due to the fact that the sampled data $\{(\boldsymbol{\theta}'_n, h'_n)\}_{n=1}^{N'}$ are independent and identically distributed, we have

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] = \frac{1}{N'}\mathbf{Var}\left[w(\boldsymbol{\theta})\nabla_{\boldsymbol{\eta}}\log p(\boldsymbol{\theta}|\boldsymbol{\rho})R(h)\right], \tag{7}$$

where h and $\boldsymbol{\theta}$ are random variables and follow the distributions $p(h, \boldsymbol{\theta}|\boldsymbol{\rho}')$.

Note that we consider the trace of the covariance matrix of gradient vectors, that is, the sum of the variance of the components of the vector. Then by upper-bounding the variance with the second moment, we have the following upper bound:

$$\begin{aligned}
&\mathbf{Var}\left[w(\boldsymbol{\theta})R(h)\nabla_{\boldsymbol{\eta}}\log p(\boldsymbol{\theta}|\boldsymbol{\rho})\right] \\
&\leq \sum_{i=1}^{\ell}\mathbb{E}_{p(h,\boldsymbol{\theta}|\boldsymbol{\rho}')}\left[(w(\boldsymbol{\theta})R(h)\nabla_{\eta_i}\log p(\boldsymbol{\theta}|\boldsymbol{\rho}))^2\right] \\
&= \sum_{i=1}^{\ell}\iint p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho}')\left(\frac{p(\boldsymbol{\theta}|\boldsymbol{\rho})}{p(\boldsymbol{\theta}|\boldsymbol{\rho}')}\right)^2(R(h))^2(\nabla_{\eta_i}\log p(\boldsymbol{\theta}|\boldsymbol{\rho}))^2\mathrm{d}h\mathrm{d}\boldsymbol{\theta} \\
&= \sum_{i=1}^{\ell}\iint p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho})w(\boldsymbol{\theta})(R(h))^2(\nabla_{\eta_i}\log p(\boldsymbol{\theta}|\boldsymbol{\rho}))^2\mathrm{d}h\mathrm{d}\boldsymbol{\theta} \\
&\leq \sum_{i=1}^{\ell}\left(\frac{\beta(1-\gamma^T)}{1-\gamma}\right)^2 w_{\max}\iint p(h|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\rho})(\nabla_{\eta_i}\log p(\boldsymbol{\theta}|\boldsymbol{\rho}))^2\mathrm{d}h\mathrm{d}\boldsymbol{\theta} \\
&= \sum_{i=1}^{\ell}\left(\frac{\beta(1-\gamma^T)}{1-\gamma}\right)^2 w_{\max}\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{\rho})}\left[(\nabla_{\eta_i}\log p(\boldsymbol{\theta}|\boldsymbol{\rho}))^2\right],
\end{aligned}$$

where $\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{\rho})}[\cdot]$ denotes the expectation of the function of random variable $\boldsymbol{\theta}$ with respect to $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{\rho})$. Subsequently, given the proof of the first part of Theorem 1 in [32], we get the upper bound of $\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right]$.

Similarly, given the same technique and the proof of the later part of Theorem 1 in [32], we could get the conclusion of the upper bound of $\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right]$.

□

B Proof of Theorem 2

Proof. First, let us derive some elementary expressions. Let $\boldsymbol{A}$, $\boldsymbol{C}$ be random variables taking values in the ℓ-dimensional space and let b be a scalar. Then,

$$\mathbf{Var}[\boldsymbol{A} - b\boldsymbol{C}] = \mathbf{Var}[\boldsymbol{A}] + b^2\,\mathbf{Var}[\boldsymbol{C}] - b\,\mathbf{Cov}[\boldsymbol{A}, \boldsymbol{C}] - b\,\mathbf{Cov}[\boldsymbol{C}, \boldsymbol{A}].$$

We still consider the trace of the covariance matrix of gradient vectors for multi-dimensional space. Assume that $\mathbb{E}[\boldsymbol{C}] = \mathbf{0}$. Then, we could have

$$\begin{aligned}\mathbf{Var}[\boldsymbol{A} - b\boldsymbol{C}] &= \mathbf{Var}[\boldsymbol{A}] + b^2\,\mathbf{Var}[\boldsymbol{C}] - 2b\,\mathbf{Cov}[\boldsymbol{A}, \boldsymbol{C}] \\ &= \mathbf{Var}[\boldsymbol{A}] + \mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}] \left\{ b^2 - 2b \frac{\mathbb{E}[\boldsymbol{A}^\top \boldsymbol{C}]}{\mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}]} \right\} \\ &= \mathbf{Var}[\boldsymbol{A}] + \mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}] \left\{ \left(b - \frac{\mathbb{E}[\boldsymbol{A}^\top \boldsymbol{C}]}{\mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}]} \right)^2 - \left(\frac{\mathbb{E}[\boldsymbol{A}^\top \boldsymbol{C}]}{\mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}]} \right)^2 \right\}.\end{aligned} \tag{8}$$

Simple calculus shows that the foregoing is minimized when

$$b = \frac{\mathbb{E}[\boldsymbol{A}^\top \boldsymbol{C}]}{\mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}]}.$$

The optimal baseline for IW-PGPE follows immediately by plugging in

$$\boldsymbol{A} = Rw \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})$$

and

$$\boldsymbol{C} = w \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})$$

for $\boldsymbol{A}$ and $\boldsymbol{C}$. Note that Eq.(8) uses the conclusion of $\mathbb{E}[w \nabla_{\boldsymbol{\rho}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho})] = \mathbf{0}$, which can be found in the proof of Theorem 4 in [32].

As the sampled data are independent and identically distributed, we have

$$\mathbf{Var}[\nabla_{\boldsymbol{\rho}} \hat{\mathcal{J}}^b_{\mathrm{IW}}(\boldsymbol{\rho})] = \frac{1}{N'} \mathbf{Var}[\boldsymbol{A} - b\boldsymbol{C}].$$

Then, according to Eq.(8) and the definition of b^*, we could have

$$\begin{aligned}&\mathbf{Var}[\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^b_{\mathrm{IW}}(\boldsymbol{\rho})] - \mathbf{Var}[\nabla_{\boldsymbol{\rho}} \widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})] \\ &= \frac{1}{N'} \left(b^2 \mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}] - 2b \mathbb{E}[\boldsymbol{A}^\top \boldsymbol{C}] + \frac{(\mathbb{E}[\boldsymbol{A}^\top \boldsymbol{C}])^2}{\mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}]} \right) \\ &= \frac{1}{N'} (b - b^*)^2 \, \mathbb{E}[\boldsymbol{C}^\top \boldsymbol{C}],\end{aligned}$$

where the expectation is over random variables h and $\boldsymbol{\theta}$ such that $(h, \boldsymbol{\theta}) \sim p(h, \boldsymbol{\theta}|\boldsymbol{\rho}')$. This completes the proof of Theorem 2. □

C Proof of Theorem 3

Proof. We define $\nabla_{\boldsymbol{\eta}}$ and $\nabla_{\boldsymbol{\eta}_i}$ as

$$\begin{aligned}\nabla_{\boldsymbol{\eta}} &= \nabla_{\boldsymbol{\eta}} \log p(\boldsymbol{\theta}|\boldsymbol{\rho}), \\ \nabla_{\boldsymbol{\eta}_i} &= \nabla_{\boldsymbol{\eta}_i} \log p(\boldsymbol{\theta}|\boldsymbol{\rho}).\end{aligned}$$

We still denote the subscripts $\boldsymbol{\rho}'$ as $p(h, \boldsymbol{\theta}|\boldsymbol{\rho}')$. According to Theorem 2, by setting $b = 0$, it is easy to know that

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] = \frac{\left(\mathbb{E}_{\boldsymbol{\rho}'}[R(h)w^2(\boldsymbol{\theta})\nabla^\top_{\boldsymbol{\eta}}\nabla_{\boldsymbol{\eta}}]\right)^2}{N'\mathbb{E}_{\boldsymbol{\rho}'}[w^2(\boldsymbol{\theta})\nabla^\top_{\boldsymbol{\eta}}\nabla_{\boldsymbol{\eta}}]}.$$

We already know that

$$\mathbb{E}_{\boldsymbol{\rho}'}[R(h)w^2(\boldsymbol{\theta})\nabla^\top_{\boldsymbol{\eta}}\nabla_{\boldsymbol{\eta}}] \leq \frac{\beta(1-\gamma^T)}{(1-\gamma)}\mathbb{E}_{\boldsymbol{\rho}'}\left[w^2(\boldsymbol{\theta})\nabla^\top_{\boldsymbol{\eta}}\nabla_{\boldsymbol{\eta}}\right].$$

Hence,

$$\begin{aligned}
&\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \\
&\leq \frac{\beta^2(1-\gamma^T)^2}{N'(1-\gamma)^2}\mathbb{E}_{\boldsymbol{\rho}'}\left[w^2(\boldsymbol{\theta})\nabla^\top_{\boldsymbol{\eta}}\nabla_{\boldsymbol{\eta}}\right] \\
&\leq \frac{\beta^2(1-\gamma^T)^2}{N'(1-\gamma)^2}w_{\max}\sum_{i=1}^{\ell}\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{\rho})}\left[(\nabla_{\eta_i})^2\right] \qquad (9) \\
&= \frac{\beta^2(1-\gamma^T)^2 B}{N'(1-\gamma)^2}w_{\max}, \qquad (10)
\end{aligned}$$

where Eq.(9) is based on the same technique used in Section A, and Eq.(10) is given by results of the proof of Theorem 1 in [32].

Similarly, we can have the lower bound as

$$\mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\eta}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] \geq \frac{\alpha^2(1-\gamma^T)^2 B}{N'(1-\gamma)^2}w_{\min}.$$

By using the same techniques, we get the bounds of the variance reduction of gradient estimation with respect to the deviation parameter $\boldsymbol{\tau}$,

$$\begin{aligned}
\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] &\leq \frac{2\beta^2(1-\gamma^T)^2 B}{N'(1-\gamma)^2}w_{\max}, \\
\mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}_{\mathrm{IW}}(\boldsymbol{\rho})\right] - \mathbf{Var}\left[\nabla_{\boldsymbol{\tau}}\widehat{\mathcal{J}}^{b^*}_{\mathrm{IW}}(\boldsymbol{\rho})\right] &\geq \frac{2\alpha^2(1-\gamma^T)^2 B}{N'(1-\gamma)^2}w_{\min},
\end{aligned}$$

which completes the proof. □

References

[1] J. Bagnell, S. Kakade, A. Ng, and J. Schneider. Policy search by dynamic programming. In *Advances in Neural Information Processing Systems*, volume 16, pages 831–388. MIT Press, 2004.

[2] L. C. Baird. Advantage updating. Technical Report WL-TR-93-1146, Wright Lab., 1993.

[3] L. Buşoniu, R. Babuška, B. De Schutter, and D. Ernst. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, Boca Raton, Florida, 2010.

[4] G. Cheng, S. Hyon, J. Morimoto, A. Ude, G.H. Joshua, Glenn Colvin, Wayco Scroggin, and C. J. Stephen. Cb: A humanoid research platform for exploring neuroscience. *Advanced Robotics*, 21(10):1097–1114, 2007.

[5] C. Cortes, Y. Mansour, and M. Mohri. Learning bounds for importance weighting. In *Advances in Neural Information Processing Systems 23*, pages 442–450, 2010.

[6] P. Dayan. Reinforcement Comparison. In *Proceedings of the 1990 Connectionist Models Summer School*, pages 45–51, San Mateo, CA, 1990. Morgan Kaufmann.

[7] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, Berlin, Germany, 1996.

[8] E. Greensmith, P. L. Bartlett, and J. Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

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

[10] S. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*, pages 1531–1538, 2002.

[11] T. Matsubara, T. Morimura, and J. Morimoto. Adaptive step-size policy gradients with average reward metric. *Journal of Machine Learning Research - Proceedings Track*, 13:285–298, 2010.

[12] N. Meuleau, L. Peshkin, and K. E. Kim. Exploration in gradient-based reinforcement learning. Technical Report 2001-003, MIT, 2001.

[13] T. Morimura, E. Uchibe, and K. Doya. Natural actor-critic with baseline adjustment for variance reduction. *Artificial Life and Robotics*, 13:275–279, 2008.

[14] A. Ng and M. 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, 2000.

[15] L. Peshkin and C. R. Shelton. Learning from scarce experience. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 498–505, 2002.

[16] J. Peters and S. Schaal. Policy gradient methods for robotics. In *Proceedings of the IEEE/RSJ International Conferece on Inatelligent Robots and Systems*, pages 2219–2225, 2006.

[17] J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[18] D. Precup, R. S. Sutton, and S. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pages 759–766, 2000.

[19] S. M. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, 1983.

[20] F. Sehnke, C. Osendorfer, T. Rückstiess, A. Graves, J. Peters, and J. Schmidhuber. Parameter-exploring policy gradients. *Neural Networks*, 23(4):551–559, 2010.

[21] C. R. Shelton. Policy improvement for POMDPs using normalized importance sampling. In *Proceedings of the Seventeenth International Conference on Uncertainty in Artificial Intelligence*, pages 496–503, 2001.

[22] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[23] S. P. Singh, T. Jaakkola, and M. I. Jordan. Learning without state-estimation in partially observable markovian decision processes. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 284–292. Morgan Kaufmann, 1994.

[24] R. Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts, 1984.

[25] R. S. Sutton and G. A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 1998.

[26] R. S. Sutton, D. Mcallester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063. MIT Press, 1999.

[27] E. Uchibe and K. Doya. Competitive-cooperative-concurrent reinforcement learning with importance sampling. In *Proceedings of International Conference on Simulation of Adaptive Behavior: From Animals and Animats*, pages 287–296. MIT Press, 2004.

[28] P. Wawrzynski. Real-time reinforcement learning by sequential actor-critics and experience replay. *Neural Networks*, 22:1484–1497, 2009.

[29] L. Weaver and N. Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Processings of the Seventeeth Conference on Uncertainty in Artificial Intelligence*, pages 538–545, 2001.

[30] R. J. Williams. Toward a theory of reinforcement-learning connectionist systems. Technical Report NU-CCS-88-3, College of Computer Science, Northeastern University, Boston, MA, 1988.

[31] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[32] T. Zhao, H. Hachiya, G. Niu, and M. Sugiyama. Analysis and improvement of policy gradient estimation. *Neural Networks*, 26:118–129, 2012.