

---

# Zeroth-order Deterministic Policy Gradient

---

**Harshat Kumar**

Dept. of Electrical and Systems Eng.  
University of Pennsylvania  
Philadelphia, PA 19104  
harshat@seas.upenn.edu

**Dionysios S. Kalogierias**

Dept. of Electrical and Systems Eng.  
University of Pennsylvania  
Philadelphia, PA 19104  
dionysis@seas.upenn.edu

**George J. Pappas**

Dept. of Electrical and Systems Eng.  
University of Pennsylvania  
Philadelphia, PA 19104  
pappasg@seas.upenn.edu

**Alejandro Ribeiro**

Dept. of Electrical and Systems Eng.  
University of Pennsylvania  
Philadelphia, PA 19104  
aribeiro@seas.upenn.edu

## Abstract

Deterministic Policy Gradient (DPG) removes a level of randomness from standard randomized-action Policy Gradient (PG), and demonstrates substantial empirical success for tackling complex dynamic problems involving Markov decision processes. At the same time, though, DPG loses its ability to learn in a model-free (i.e., actor-only) fashion, frequently necessitating the use of critics in order to obtain consistent estimates of the associated policy-reward gradient. In this work, we introduce Zeroth-order Deterministic Policy Gradient (ZDPG), which approximates policy-reward gradients via two-point stochastic evaluations of the  $Q$ -function, constructed by properly designed low-dimensional action-space perturbations. Exploiting the idea of random horizon rollouts for obtaining unbiased estimates of the  $Q$ -function, ZDPG lifts the dependence on critics and restores true model-free policy learning, while enjoying built-in and provable algorithmic stability. Additionally, we present new finite sample complexity bounds for ZDPG, which improve upon existing results by up to two orders of magnitude. Our findings are supported by several numerical experiments, which showcase the effectiveness of ZDPG in a practical setting, and its advantages over both PG and Baseline PG.

## 1 Introduction

Reinforcement Learning (RL) has proven itself a well-known and effective approach for tackling dynamic optimization problems, and is ubiquitous in many application areas [1], such as robotic manipulation [16], supply chain management [10], games [28, 31], and wireless communications [34, 5], to name a few. A general approach for solving RL problems is *policy search*, which relies on directly learning a policy to maximize rewards through interaction with an unknown environment. Hypothesizing that actions are chosen according to a parameterized distribution (a.k.a. a randomized policy), *Policy Gradient (PG)* is a type of policy search algorithm that repeatedly updates its policy through a specially-derived stochastic gradient [30], enabling solutions in complex problems with continuous state-action spaces. Backing its empirical success are a range of theoretical guarantees including guaranteed policy improvement with smoothness assumptions [26], sample complexity bounds [35], and variance reduction [24].

*Deterministic Policy Gradient (DPG)* considers RL problems by naturally assuming that the policy is a deterministic mapping from states to actions [29]. This formulation removes policy randomization, otherwise integral to classical PG, and permits the complete characterization of the gradient of the

Table 1: Summary of Rate Results (for number of iterations  $T$  and accuracy  $\epsilon$ )

	Grad-Lipschitz $Q$ -function		Hessian-Lipschitz $Q$ -function	
Number of Rollouts ( $N$ )	1	$T$	1	$\sqrt{T}$
Smoothness Parameter ( $\mu$ )	$T^{-1/4}$	$T^{-1/2}$	$T^{-1/6}$	$T^{-1/4}$
Stepsize ( $\alpha$ )	$T^{-3/4}$	$T^{-1/2}$	$T^{-2/3}$	$T^{-1/2}$
Convergence Rate	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-2})$

composed policy-reward function with respect to the parameters of the policy. DPG has been shown to outperform its stochastic counterparts empirically on a wide variety of RL problems [19, 12]. To evaluate the gradient, however, conventional DPG algorithms rely on critics [17], which use dynamic programming and function approximation to estimate the state-action value, or  $Q$ -function. Also, guarantees for convergence of  $Q$ -learning methods exist only for linear functions [2, 3], which work under the assumption that a rich feature representation of the state is available. Finding such a feature representation is known to itself be a nontrivial problem [33]. Furthermore, the bias on the gradient is difficult to characterize [18], resulting in suboptimal convergence rates [32].

In this work, we introduce *Zeroth-order Deterministic Policy Gradient (ZDPG)*, which advocates the use of (zeroth-order) noisy estimates of the  $Q$ -function for approximating the corresponding (deterministic) policy-reward gradient, in the standard infinite horizon MDP formulation. In particular, by exploiting the recently introduced idea of *random horizon rollouts* for obtaining unbiased stochastic estimates of the  $Q$ -function [25, 35], we show that it is possible to efficiently construct consistent two-point approximations (single or in batches) of the associated policy-reward gradient [4, 22]; this is achieved by evaluating pairs of the  $Q$ -function at low-dimensional random action-space perturbations, involving *initial actions only*. Such a construction is rather important from a computational perspective, for two main reasons. First, for most popular choices of policy parameterizations (e.g., Deep Neural Networks (DNNs)), the dimension of the corresponding parameter space is typically much larger than the dimension of the action space, where our random perturbations are introduced [32]. This is a key feature, because it is well-known that the dimension of the perturbation space has drastic negative effects on the performance of zeroth-order methods [4, 22]. Second, the fact that we (randomly) perturb only the initial action of the underlying dynamic program makes our ZDPG approach readily implementable, and easily applicable to the infinite horizon setting. As compared with standard PG and DPG, ZDPG enjoys the following additional operational advantages:

- As with DPG, ZDPG eliminates the need for randomized policies. Although we are not the first to achieve this goal [29], we would still like to perpetuate the narrative that randomized policies, although most common, are *not* really necessary in PG-based RL.
- *At the same time*, while ZDPG exploits gradient information of the adopted (and known) policy parameterization, it *restores* true *model-free* policy learning (as in standard PG); this is naturally lost in DPG, where the use of an auxiliary critic is typically invoked [29].
- ZDPG enjoys built-in and provable quasi-gradient variance stability (albeit at the expense of an additional system rollout), which is very similar in principle to (heuristic) baseline PG techniques [11].

We also present detailed analysis characterizing the convergence rate of our proposed ZDPG algorithm under common problem regularity conditions. From a technical standpoint, our analysis is different from that of conventional zeroth-order methods for nonconvex optimization – which follows the general structure of defining a smoothed surrogate objective (reward), analyzing the convergence of the resulting smoothed problem, and then relating those results back to the original problem (see, e.g., [15, 13]). More specifically, our convergence analysis is based on the explicit construction of a smoothed policy-reward gradient surrogate (not necessarily corresponding to the gradient of some smoothed objective surrogate), which admits a zeroth-order representation matching exactly our rollout-pair-based two-point policy-reward gradient approximations. A key feature of the proposed gradient surrogate is that it constitutes a uniform approximation to the true (deterministic) policy-reward gradient of Silver *et al.* [29]. This fact is then exploited to analyze the resulting stochastic quasi-gradient algorithm (ZDPG) as a method for solving the original DPG problem, directly.

Our rate results are summarized in Table 1. Specifically, we show that under the assumption of a Lipschitz and smooth  $Q$ -function, ZDPG with a single rollout pair achieves a rate of order  $\mathcal{O}(\epsilon^{-4})$

(given accuracy  $\epsilon > 0$ ), whereas if the  $Q$ -function has also Lipschitz Hessian, ZDPG achieves a rate of order  $\mathcal{O}(\epsilon^{-3})$ . If, further, multiple rollout pairs are available (mini-batch equivalent), then the aforementioned rates both improve to an order of  $\mathcal{O}(\epsilon^{-2})$ , in which case the number of rollout pairs are precisely  $T$  and  $\sqrt{T}$ , respectively, where  $T$  is the total number of ZDPG iterations. To the best of our knowledge, our results are the first of their kind pertinent to the standard infinite horizon setting, also outperforming previous results reported for the finite horizon setting [32] by up to two orders of magnitude (depending, of course, on problem conditioning).

The performance of ZDPG is empirically evaluated on a point agent navigation problem at the presence of environment obstacles, where there is no knowledge of the system dynamics. Our numerical results demonstrate that ZDPG substantially outperforms both standard PG and PG with a baseline, under both noiseless and noisy system dynamics, and in terms of both mean performance and its variance. In particular, our simulations elucidate the fact that ZDPG enjoys clear operational advantages over PG with a baseline, which may be considered as the analogous two-point update for the stochastic policy setting.

## 2 Deterministic Policy Gradient

We consider a Reinforcement Learning (RL) problem where an agent moves through a continuous and compact state space  $\mathcal{S} \subset \mathbb{R}^q$ , and takes actions in a finite dimensional action space  $\mathcal{A} = \mathbb{R}^p$ . The agent's task is to accumulate as much reward as possible in the long term, where the reward is revealed to the agent by the environment at each step. This generic problem may be abstracted by a Markov decision process (MDP) as a tuple  $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ , where  $R : \mathcal{S} \times \mathbb{R}^p \rightarrow \mathbb{R}$  is the reward function, and  $P_{s \rightarrow s'}^a := p(s' | (s, a) \in \mathcal{S} \times \mathcal{A})$  determines the probability of moving to state  $s'$  starting from state  $s$  and action  $a$ , satisfying the Markov property  $p(s_{t+1} | (s_u, a_u) \in \mathcal{S} \times \mathcal{A}, \forall u \leq t) = p(s_{t+1} | (s_t, a_t) \in \mathcal{S} \times \mathcal{A})$ . The value  $\gamma \in (0, 1)$  is the discount factor which determines how much future rewards matter to the behavior of the agent. Lastly, the *deterministic* policy  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  defines a mapping from a state to an action. Then, the problem of interest is to choose a policy that maximizes the expected discounted sum of rewards over all starting states; this means finding a policy that maximizes the expected value function  $V : \mathcal{S} \rightarrow \mathbb{R}$  starting from state  $s$  or, formally,

$$\max_{\pi} \mathbb{E}_{s \sim \rho^0} \left[ V(s) := \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t = \pi(s_t)) \middle| s \right] \right], \quad (1)$$

where  $\rho^0$  is the initial state distribution. By conditioning the value function on an action, we can define the state-action function  $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$  by

$$Q^{\pi}(s, a) := R(s, a) + \gamma \mathbb{E}_{s' \sim p(\cdot | s, a)} [V(s')] = \mathbb{E} \left[ R(s_0, a_0) + \sum_{t=1}^{\infty} \gamma^t R(s_t, \pi(s_t)) \middle| s_0 = s, a_0 = a \right]. \quad (2)$$

The  $Q$ -function determines the quality of taking an action  $a$  at state  $s$  and then following policy  $\pi$  for the remainder of time. By selecting the first action according to the policy, the  $Q$ -function is equivalent to the value function, and hence the problem (1) is equivalently written as

$$\max_{\pi} \mathbb{E}_{s \sim \rho^0} [Q(s, \pi(s))]. \quad (3)$$

Solving (3) requires searching over a function space, which is generally an intractable problem. To circumvent this complexity, the policy is parameterized by some  $\theta \in \Theta \subset \mathbb{R}^d$  so that the search is over a Euclidean space of finite dimension. The problem of interest consequently becomes

$$\max_{\theta} \mathbb{E}_{s \sim \rho^0} [Q^{\pi_{\theta}}(s, \pi_{\theta}(s))] =: J(\theta). \quad (4)$$

In their seminal work [29], Silver *et al.* proved that, under mild problem regularity conditions, the gradient of (4) can be expressed by

$$\nabla J(\theta) = \mathbb{E}_{s \sim \rho^{\pi}} [\nabla_{\theta} \pi_{\theta}(s) \nabla_a Q^{\pi_{\theta}}(s, a)|_{a=\pi_{\theta}(s)}], \quad (5)$$

where  $\rho^{\pi}$  is the so-called (improper) *discounted state distribution* under policy  $\pi$ , defined as  $\rho^{\pi}(s') = \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p_0(s) p(s \rightarrow s', t, \theta) ds$ , and  $p(s \rightarrow s', t, \theta)$  denotes the density at state  $s'$  after starting from state  $s$  and transitioning with  $t$  time steps given policy parameterized by  $\theta$ . Next, we formally define some standard assumptions on the policy gradient, reward function, and  $Q$ -function, as follows.

**Assumption 1 (Bounded Policy Gradient)** The policy gradient  $\nabla_{\theta}\pi_{\theta}(s)$  is uniformly bounded on  $\Theta \times \mathcal{S}$ , i.e., there is  $B_{\Theta} < \infty$ , such that  $\sup_{(\theta,s) \in \Theta \times \mathcal{S}} \|\nabla_{\theta}\pi_{\theta}(s)\| \leq B_{\Theta}$ .

**Assumption 2 (Bounded Rewards)** The reward  $R$  is uniformly bounded on  $\mathcal{S} \times \mathcal{A}$ , i.e., there is  $U_R < \infty$ , such that  $\sup_{(s,a) \in \mathcal{S} \times \mathcal{A}} |R(s,a)| \leq U_R$ .

By Assumption 2, it also follows that the  $Q$ -function is also uniformly bounded as

$$|Q^{\pi_{\theta}}(s,a)| \leq \sum_{t=0}^{\infty} \gamma^t \cdot U_R = \frac{1}{1-\gamma} U_R =: \mathcal{Q}.$$

Both Assumptions 1 and 2 are standard for deriving performance guarantees in the policy search literature [35, 18, 2], and are easily satisfied by common choices of the deterministic policy and reward functions. Additionally, we require sufficient smoothness of the  $Q$ -function, as follows.

**Assumption 3 (Lipschitzness of the  $Q$ -function)** The  $Q$ -function is action-Lipschitz uniformly on  $\Theta \times \mathcal{S}$ . That is, there is a number  $L < \infty$ , such that, for every  $a_1, a_2 \in \mathcal{A}$ ,

$$\sup_{(\theta,s) \in \Theta \times \mathcal{S}} |Q^{\pi_{\theta}}(s,a_1) - Q^{\pi_{\theta}}(s,a_2)| \leq L \|a_1 - a_2\|.$$

**Assumption 4 (Smoothness of the  $Q$ -function)** The  $Q$ -function is action-smooth uniformly on  $\Theta \times \mathcal{S}$ . That is, there is a number  $G < \infty$ , such that, for every  $a_1, a_2 \in \mathcal{A}$ ,

$$\sup_{(\theta,s) \in \Theta \times \mathcal{S}} \|\nabla_a Q^{\pi_{\theta}}(s,a_1) - \nabla_a Q^{\pi_{\theta}}(s,a_2)\| \leq G \|a_1 - a_2\|.$$

Intuitively, Assumptions 3 and 4 means that similar actions from the same state should have similar values. This principle has informed Actor-Critic algorithms in practice as they are known to avoid overfitting to narrow peaks in the value estimate [7]. Furthermore, existing convergence analysis for zeroth-order action space exploration with finite horizon use these assumptions as well [32]. Lastly, in some of our results we assume that the problem is well-conditioned with Lipschitz Hessians.

**Assumption 5 (Smoothness of  $\nabla Q$ )** The gradient field  $\nabla Q$  is action-smooth uniformly on  $\Theta \times \mathcal{S}$ . That is, there is a number  $H < \infty$ , such that, for every  $a_1, a_2 \in \mathcal{A}$ ,

$$\sup_{(\theta,s) \in \Theta \times \mathcal{S}} \|\nabla_a^2 Q^{\pi_{\theta}}(s,a_1) - \nabla_a^2 Q^{\pi_{\theta}}(s,a_2)\| \leq H \|a_1 - a_2\|.$$

Assumption 5, though not entirely standard, holds for a variety of problems with well behaved reward functions and policy representations, such as the Linear Quadratic Regulator (LQR) [6, 20, 21] and its many (and potentially nonlinear) variations. A reward function may be designed to achieve such an assumption, as well [23]. The trade-off between our various smoothness assumptions and the convergence rates achieved in this work is demonstrated in Table 1.

In order to compute a stochastic approximation of (5), we need to both sample from the discounted state distribution  $\rho^{\pi}(s)$  and evaluate the gradient expression. Because the policy is deterministic,  $\nabla_{\theta}\pi_{\theta}(s)$  is the Jacobian of the policy  $\pi_{\theta}$  evaluated at  $s$  which can be explicitly calculated. Instead of using a critic network to estimate the  $\nabla_a Q^{\pi_{\theta}}(s,a)$  [29], which requires many state-action-reward samples before converging to something meaningful, we propose the use of two point zeroth-order action space perturbed  $Q$ -function estimates. In the following section, we review how to use zeroth-order information to evaluate stochastic gradient estimates.

### 3 Zeroth-order Gradient Estimates and ZDPG

First, we introduce a smoothed approximation to the  $Q$ -function by exploiting low-dimensional perturbations with respect to the action  $a$ . Let  $\mu > 0$  be the *smoothing parameter*, and let  $\mathbf{u} \sim \mathcal{N}(0, I_p)$ . Then, the  $\mu$ -smoothed  $Q$ -function is defined as

$$Q_{\mu}^{\pi_{\theta}}(s,a) := \mathbb{E}_{\mathbf{u}} [Q^{\pi_{\theta}}(s, a + \mu \mathbf{u})].$$

By introducing smoothing, we are able to sample an unbiased stochastic estimate of  $\nabla_a Q_{\mu}^{\pi_{\theta}}(s,a)$  via a two-point evaluation of the original  $Q$ -function. Namely, we recall the following key property (in terms of the  $Q$ -function), true in general for every globally Lipschitz function as originally presented in [22], then extended to a wider class of functions in [15].

---

**Algorithm 1** Zeroth-order Deterministic Policy Gradient

---

**Require:**  $\theta_0, \gamma, \mu, \alpha, N$

```
for  $t = 0, 1, \dots$  do
  Sample  $T_Q \sim \text{Geom}(1 - \gamma)$ 
  Sample  $s_t \sim (1 - \gamma)\rho^{\pi_{\theta_t}}$  [Algorithm 3 with  $(T, \theta) = (T_Q, \theta_t)$ ]
  Sample  $\mathbf{u} \sim \mathcal{N}(0, I_p)$ 
  Initialize Estimates  $\hat{Q}_t^+ \leftarrow 0, \hat{Q}_t^- \leftarrow 0$ 
  for  $n = 0, 1, \dots, N - 1$  do
     $\hat{Q}_{t,n}^+ \leftarrow$  Algorithm 2 with  $(T, s_0, a_0, \theta) = (T_Q, s_t, \pi_{\theta_t}(s_t) + \mu\mathbf{u}, \theta_t)$ 
     $\hat{Q}_{t,n}^- \leftarrow$  Algorithm 2 with  $(T, s_0, a_0, \theta) = (T_Q, s_t, \pi_{\theta_t}(s_t), \theta_t)$ 
     $\hat{Q}_t^+ \leftarrow \hat{Q}_t^+ + \frac{1}{N}\hat{Q}_{t,n}^+$ 
     $\hat{Q}_t^- \leftarrow \hat{Q}_t^- + \frac{1}{N}\hat{Q}_{t,n}^-$ 
  end for
  Evaluate  $\Psi_t = \nabla_{\theta}\pi_{\theta_t}(s_t)$ 
   $g_t \leftarrow \frac{1}{1-\gamma}\Psi_t \left( \frac{\hat{Q}_t^+ - \hat{Q}_t^-}{\mu} \right) \mathbf{u}$ 
   $\theta_{t+1} \leftarrow \theta_t + \alpha g_t$ 
end for
```

---

**Lemma 1** [15, Lemma 2] *For every  $\mu > 0$ , the  $\mu$ -smoothed  $Q$ -function surrogate  $Q_{\mu}^{\pi_{\theta}}$  is differentiable, and its gradient admits the representations*

$$\begin{aligned} \nabla_a Q_{\mu}^{\pi_{\theta}}(s, a) &\equiv \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0, I_p)} \left[ \frac{Q^{\pi_{\theta}}(s, a + \mu\mathbf{u}) - Q^{\pi_{\theta}}(s, a)}{\mu} \mathbf{u} \right] \\ &\equiv \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(0, I_p)} \left[ \frac{Q^{\pi_{\theta}}(s, a + \mu\mathbf{u}) - Q^{\pi_{\theta}}(s, a - \mu\mathbf{u})}{2\mu} \mathbf{u} \right]. \end{aligned} \quad (6)$$

Driven by Lemma 1, we propose the following estimate of the objective gradient defined in (5) where, instead of the true gradient  $\nabla_a Q^{\pi_{\theta}}(s, a)$  we use the smoothed gradient  $\nabla_a Q_{\mu}^{\pi_{\theta}}(s, a)$  to obtain the quasi-gradient

$$\hat{\nabla} J(\theta) = \mathbb{E}_{s \sim \rho^{\pi}} [\nabla_{\theta}\pi_{\theta}(s) \nabla_a Q_{\mu}^{\pi_{\theta}}(s, a)|_{a=\pi_{\theta}(s)}]. \quad (7)$$

Lemma 1 provides a procedure to obtain an unbiased estimate of  $\nabla_a Q_{\mu}^{\pi_{\theta}}(s, a)$ , which is to sample  $\mathbf{u} \sim \mathcal{N}(0, I_p)$  and then evaluate the  $Q$ -function with *initial* actions  $a$  and  $a + \mu\mathbf{u}$ . By selecting the smoothing parameter  $\mu$  sufficiently small, the gradient estimate becomes close to the deterministic policy gradient (see Section 4, Theorem 1). Obtaining two samples of the  $Q$ -function at a specific state may not be practical on a real system, but can be made feasible through the use of simulators which allow resetting and reproduction of the same stochastic environment [27]. The resulting stochastic search algorithm using the two point zeroth-order gradient estimate is presented in pseudocode in Algorithm 1, where  $\alpha$  is the policy parameter stepsize. We employ Monte-Carlo variance reduction by running the system  $N \geq 1$  times ( $N = 1$  implies no reduction).

Algorithm 1 functions with the assumption that both an unbiased estimate of the  $Q$ -function evaluated at any state-action pair as well as sampling from the discounted state distribution  $\rho^{\pi_{\theta}}$  are feasible. Here, we describe both procedures which involve sampling a horizon length from a geometric distribution. In [25, Proposition 2], it was shown that the procedure described by Algorithm 2 results in unbiased estimates of the  $Q$ -function defined by (2). As the next result asserts, the aforementioned estimates are of finite variance, as well. Due to lack of space, we defer all subsequent proofs of our theoretical results to the supplementary material.

**Lemma 2** *Let Assumption 2 be in effect. Then, the estimate  $\hat{Q}$  produced by Algorithm 2 is of bounded variance, independent of the inputs of Algorithm 2.*

For later reference, let  $w_{t,n}^{\pm} := \hat{Q}_{t,n}^{\pm} - Q^{\pi_{\theta_t}}(\pm)$ , where " $\pm$ " refers to the respective initial state-action pair. Also introduced in [25], Algorithm 3 describes a similar procedure to obtain samples  $s \sim (1 - \gamma)\rho^{\pi_{\theta}}$ , where  $(1 - \gamma)\rho^{\pi_{\theta}}$  is the *proper* (i.e., scaled to sum to unity) version of the discounted state distribution  $\rho^{\pi_{\theta}}$ .

---

**Algorithm 2**  $Q$ -function Sampler [25]

---

**Require:**  $\hat{Q} \leftarrow 0, T, s_0, a_0, \theta$   
**for**  $t = 0, 1, \dots, T - 1$  **do**  
    Collect Reward  $\hat{Q} \leftarrow \hat{Q} + R(s_t, a_t)$   
    Advance System  $s_{t+1} \sim \mathbb{P}(s' | s_t, a_t)$   
    Select Action  $a_{t+1} \leftarrow \pi_\theta(s_{t+1})$   
**end for**  
Collect Reward  $\hat{Q} \leftarrow \hat{Q} + R(s_T, a_T)$

---



---

**Algorithm 3** Discounted State Sampler [25]

---

**Require:**  $T, \theta$   
Sample  $s_0 \sim \rho^0$   
**for**  $t = 0, 1, \dots, T - 1$  **do**  
    Select Action  $a_t \leftarrow \pi_\theta(s_t)$   
    Advance System  $s_{t+1} \sim \mathbb{P}(s' | s_t, a_t)$   
**end for**  
Return  $s_T$

---

## 4 Convergence Analysis

Somewhat departing from conventional analysis of zeroth-order methods, we instead bound the bias between our proposed gradient (7) with the true gradient (5) and characterize the rate using a biased variant of stochastic gradient descent. First, we present the theorem which relates the smoothed objective function gradient  $\hat{\nabla}J(\theta)$  with the true gradient  $\nabla J(\theta)$ .

**Theorem 1** *Let Assumptions 1-3 be in effect. Then, under Assumption 4, it is true that*

$$\sup_{\theta \in \Theta} \|\nabla J(\theta) - \hat{\nabla}J(\theta)\| \leq B_\Theta \mu \sqrt{p} G / (1 - \gamma).$$

*Alternatively, under Assumption 5, it is true that*

$$\sup_{\theta \in \Theta} \|\nabla J(\theta) - \hat{\nabla}J(\theta)\| \leq B_\Theta \mu^2 (p + 4)^2 H / (1 - \gamma).$$

Assumption 4 is key for the former part of Theorem 1, as the difference between the gradients of the two point  $Q$ -function with respect to the action  $a$  is bounded. The latter part of Theorem 1 is a byproduct of [22, Lemma 3]. Alternatively, the relation between the estimated gradient with two points and the true gradient could be characterized by [22, Lemma 3]. Requiring the same assumptions as detailed here except for the bounded objective, this result is not advantageous as it results in a bound of order  $\mathcal{O}(p^{3/2})$ , contrary to the  $\mathcal{O}(p^{1/2})$  bound we provide for this specific case.

By construction of Algorithm 1, we have that  $g_t$  is an unbiased estimate of the smoothed gradient  $\hat{\nabla}J(\theta)$  defined in (7). Next we will show that the second-order moment of  $g_t$  is bounded. Doing so will allow us to use analysis similar to [8] in order to characterize the rate of Algorithm 1.

**Lemma 3** *Let Assumptions 1-3 be in effect. Then, the second-order moment of the quasi-gradient  $g_t$  in Algorithm 1 is bounded. In particular, it is true that*

$$\mathbb{E}[\|g_t\|^2] \leq \frac{2B_\Theta^2}{(1 - \gamma)^2} \left( (p + 4)^2 L^2 + \frac{p\sigma^2}{\mu^2 N} \right) =: V,$$

where  $\infty > \sigma^2 \geq \mathbb{E}\{|w_{t,n}^+ - w_{t,n}^-|^2 | s_t, \mathbf{u}, \theta_t, T_Q\}$ , deterministic and same for all  $t$  and  $n$ .

We are now ready to state our main theorem, which establishes the convergence rate of Algorithm 1 with respect to the true gradient of  $J(\theta)$ .

**Theorem 2** *Let Assumptions 1-3 be in effect. Further let the objective  $J(\theta)$  be  $L_J$ -smooth and  $G_J$ -Lipschitz continuous. Then, under Assumption 4, it is true that*

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla_\theta J(\theta_t)\|^2] \leq \begin{cases} \mathcal{O}(p^2 T^{-1/4}), & \text{if } (N, \alpha, \mu) = (1, T^{-3/4}, T^{-1/4}) \\ \mathcal{O}(p^2 T^{-1/2}), & \text{if } (N, \alpha, \mu) = (T, T^{-1/2}, T^{-1/2}). \end{cases}$$

*Alternatively, under Assumption 5, it holds that*

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla_\theta J(\theta_t)\|^2] \leq \begin{cases} \mathcal{O}(p^2 T^{-1/3}), & \text{if } (N, \alpha, \mu) = (1, T^{-2/3}, T^{-1/6}) \\ \mathcal{O}(p^2 T^{-1/2}), & \text{if } (N, \alpha, \mu) = (T^{1/2}, T^{-1/2}, T^{-1/4}). \end{cases}$$

Theorem 2 requires that the objective function  $J(\theta)$  is smooth and Lipschitz. This is easily satisfied under *all* Assumptions 1-4, whenever that the adopted policy parameterization is sufficiently well-behaved. A smooth policy serves this purpose, for instance. Further, for simplicity, we have presented Theorem 2 for the unconstrained setting, where  $\Theta = \mathbb{R}^d$ . In practice, though, we may need to constrain the parameters. In this case, Theorem 2 continues to hold (*almost*), but with the squared gradient norm being replaced by the merit function  $\|\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t\|^2/\alpha^2$ ; see, e.g., [9].

## 5 Numerical Results

In this section, we empirically evaluate ZDPG and compare its performance to vanilla Policy Gradient (PG) [30] and Policy Gradient with Baseline (PG-B) [11] with varying levels of system noise and MC variance reduction. First, we formally describe the Markov Decision Process we choose to solve, then we present our results and discuss the key takeaways.

### 5.1 Problem Setting: Navigating Around an Obstacle with Unknown Agent Dynamics

For our empirical evaluation of ZDPG, we consider a two-dimensional continuous state-action navigation problem, where a point agent aims to avoid a spherical (for simplicity) obstacle and converge fast to some fixed target while satisfying its motion control dynamics. This goal is achieved by appropriately shaping the associated reward function. Let  $s^* \in \mathbb{R}^2$  represent the target the point agent wishes to reach. Let  $s^c \in \mathbb{R}^2$  and  $r \in \mathbb{R}_+$  denote the center and radius of the spherical obstacle the agent wishes to avoid. Then, we define the reward  $R : \mathbb{R}^2 \rightarrow \mathbb{R}$  as a combination of two artificial potential functions: An attractive potential  $\phi^{att} : \mathbb{R}^2 \rightarrow \mathbb{R}_+$  with a maximum at the target, and a repulsive artificial potential  $\phi^{rep} : \mathbb{R}^2 \rightarrow \mathbb{R}_+$  with a minimum at the center of the object. Specifically, we set the attractive potential to be the negative Euclidean distance to the goal location  $s^*$  squared, that is,  $\phi^{att}(s) := -\|s - s^*\|_2^2$ , where  $s$  is the location (state) of the agent. Similar to [14], we define the repulsive potential as  $\phi^{rep}(s) := 1 - 1/\beta(s)$ , where

$$\beta(s) := \left(1 - \left(\frac{1 + r^4}{r^4}\right) \cdot \frac{(\|s - s^c\|^2 - r^2)^2}{1 + (\|s - s^c\|^2 - r^2)^2}\right)^{\frac{1}{2}(1 - \text{sign}(\|s - s^c\| - r))}.$$

Completely characterized by the state of the agent, the reward is equal to the sum of these attractive and repulsive potentials, that is,  $R(s) = \phi^{att}(s) + \phi^{rep}(s)$ . Also, the motion dynamics of the point agent are described by the controlled random walk

$$s_{t+1} = As_t + \eta \frac{\pi_{\theta}(s_t)}{\|\pi_{\theta}(s_t)\|} + n_t, \quad (8)$$

where  $A \in \mathbb{R}^{2 \times 2}$  is the motion state transition matrix of the agent,  $\{n_t\}$  is a random process modeling state process noise, and  $\eta > 0$  (together with policy normalization) imposes implicit resource constraints on the agent motion. Here, we assume that, except from the agent control vector (second term on the right-hand side of (8)), the dynamics of the agent are *unknown*, an assumption which conforms with the standard policy gradient setting. Lastly, we consider a linear, deterministic policy to determine optimal directions that the point agent will follow given its location to its prescribed destination. In particular, we adopt a Radial-Basis-Function (RBF) expansion as our (universal) policy parameterization, i.e.,

$$\pi_{\theta}(s) = \sum_{i=1}^d \theta_i \kappa(s, \bar{s}_i), \quad (9)$$

where  $\kappa(s, s') = \exp(-\|s - s'\|_2^2 / 2\sigma_{\kappa}^2)$  defines the RBF kernel,  $\theta_i \in \mathbb{R}^2, i = 1, \dots, d$  are the parameter vectors to be learned,  $\sigma_{\kappa}^2$  determines the variance of the kernel, and  $\bar{s}_i, i = 1, \dots, d$  are the associated RBF centers.

For all simulations, we select the target of the agent to be located at  $s^* = (-5, -5)$ , and the obstacle to be centered at  $s^c = (0, 0)$  with a radius of  $r = 2.5$ . We define the potential with a slightly larger radius ( $r = 3$ ) to ensure that the agent does not collide with the boundary of the obstacle. We select the centers of the RBF kernels to be spaced 0.25 units from each other populating the  $[-10, 10] \times [-10, 10]$  grid. The initial state distribution is uniform on  $[-10, 10] \times [-10, 10]$ .

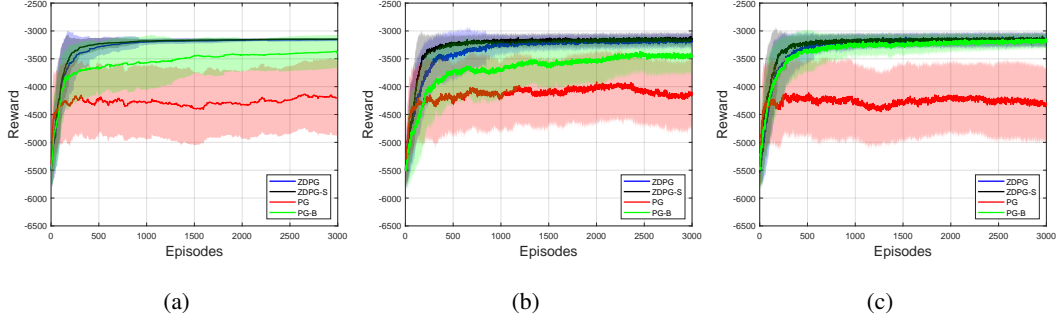


Figure 1: Average reward per episode with confidence bounds over 50 trials. Learning rate set to  $\alpha = 10^{-7}$  for all simulations. (a) Noiseless case with no MC variance reduction ( $N = 1$ ), ZDPG and ZDPG-S smoothing parameters set to 0.05 and 0.5 respectively. Action noise for PG and PG-B set to  $\sigma_a^2 = 0.025$ . (b) Noisy case ( $\omega = 0.01$ ) with no MC reduction ( $N = 1$ ), ZDPG and ZDPG-S smoothing parameters set to 0.25 and 0.5 respectively. Action noise for PG and PG-B set to  $\sigma_a^2 = 0.025$ . (c) Noisy case (same as (b)) with MC reduction ( $N = 10$ ).

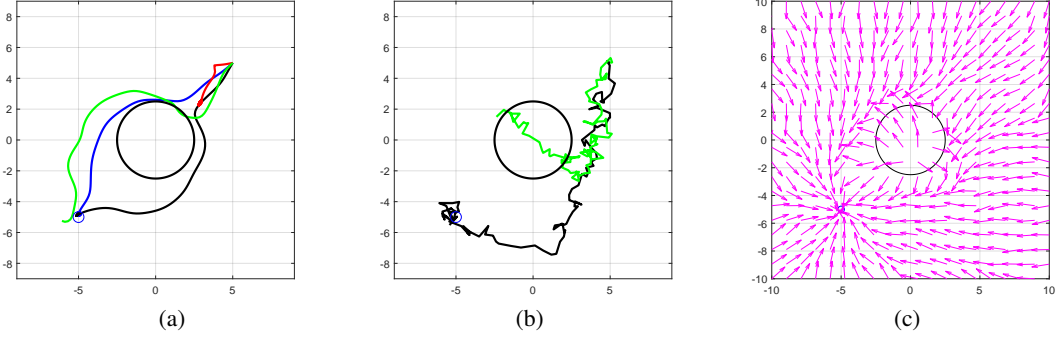


Figure 2: Sample trajectories with fixed length  $T = 100$  of learned policies corresponding to noiseless ((a):  $N = 1$ ,  $\omega = 0$ ) and noisy ((b):  $N = 10$ ,  $\omega = 0.1$ ) settings (c) Learned field plot for ZDPG-S with noise level  $\omega = 0.1$  shows that the variance in Figure 1 comes only from the system noise.

## 5.2 Results

The learning curves shown in Figure 1 illustrate the performance of ZDPG and its symmetric counterpart ZDPG-S (based on (6)), compared with PG and PG with a *stochastic* baseline (PG-B), with discount factor  $\gamma = 0.8$  in the case with and without system noise ( $n_t = \mathbf{0}$  and  $n_t \sim \mathcal{N}(\mathbf{0}, \omega \cdot I_2)$ , respectively). Also, we set  $A = I_2$ . For PG and PG-B, we let  $\tilde{\pi}$  denote the stochastic policy where the action is selected by  $a_t \sim \tilde{\pi}_\theta(s_t)$ . In particular, we let  $\tilde{\pi}$  be a Gaussian policy with mean equal to  $\pi_\theta(s)$  [c.f. (9)] and a standard choice of covariance matrix  $\Sigma = \sigma_a^2 \cdot I_2$ . For PG-B, the baseline is fairly chosen and is analogous as compared to ZDPG in that it scales the update direction  $\nabla_\theta \log \tilde{\pi}_\theta(s)$  by a two-point difference ( $\hat{Q}(s, a) - \tilde{V}(s)$ ), where  $\tilde{V}(s)$  is a single-rollout estimate of the associated value function at state  $s$ . After each policy update, we evaluate the system by generating a rollout of fixed length  $T = 20$  from a fixed starting point  $s^0 = (5, 5)$ . As done commonly in practice (as a heuristic), we evaluate the stochastic policies by setting the covariance matrix equal to zero thereby making the evaluation policy deterministic.

Depending on the level of the system noise, ZDPG(-S) consistently outperforms PG-B (and also PG), both in mean performance and the associated variance. Interestingly, the improvement of ZDPG over PG-B is sharper when the agent dynamics are noisy (also see our Supplementary Material, Section E). In fact, we observe that, either with *or* without system noise, the perturbation of the initial state with a deterministic policy indeed provides more meaningful information for policy improvement than in the setting of a stochastic policy (also see Figure 2). One should also note that PG-B performs better in the noisy setting with MC variance reduction (1 (c)) than the noiseless setting (1 (a)). Indeed, variance reduction improves the performance of all methods, except for vanilla policy gradient. We emphasize though that we evaluate stochastic policies using only their means. Our empirical results corroborate the claim that randomness in the policy is *not* necessary to solve MDPs within the PG-based RL framework.



## 6 Conclusions and Future Work

In this work, we introduced the ZDPG algorithm, which is based on an efficient two-point zeroth-order approximation of the deterministic policy gradient, for the infinite horizon Markov decision problem. Using random horizons to evaluate the zeroth-order estimates of the state-action function, we showed that low-dimensional perturbations in the action space can effectively replace the need for randomized policies in the classical policy gradient setting. We also presented finite sample complexity results for ZDPG, which improve upon the state of the art by up to two orders of magnitude. Finally, we corroborated our results with a numerical evaluation on a navigation problem which showcased the advantages of ZDPG over both standard PG and PG with a baseline. For future work, variance reduced ZDPG could be realized with more sample efficient variance reduction techniques, such as gradient averaging or using a critic network to evaluate zeroth order state action function.

## References

- [1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.
- [2] Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. *arXiv preprint arXiv:1806.02450*, 2018.
- [3] Gal Dalal, Balázs Szörényi, Gagan Thoppe, and Shie Mannor. Finite sample analyses for td (0) with function approximation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [4] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.
- [5] Mark Eisen, Clark Zhang, Luiz FO Chamon, Daniel D Lee, and Alejandro Ribeiro. Learning optimal resource allocations in wireless systems. *IEEE Transactions on Signal Processing*, 67(10):2775–2790, 2019.
- [6] Maryam Fazel, Rong Ge, Sham M Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.
- [7] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.
- [8] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- [9] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2):267–305, 2016.
- [10] Ilaria Giannoccaro and Pierpaolo Pontrandolfo. Inventory management in supply chains: a reinforcement learning approach. *International Journal of Production Economics*, 78(2):153–161, 2002.
- [11] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.
- [12] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.
- [13] Dionysios S Kalogerias, Mark Eisen, George J Pappas, and Alejandro Ribeiro. Model-free learning of optimal ergodic policies in wireless systems. *arXiv preprint arXiv:1911.03988*, 2019.
- [14] Dionysios S Kalogerias and Athina P Petropulu. Mobi-cliques for improving ergodic secrecy in fading wiretap channels under power constraints. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1578–1591. IEEE, 2014.
- [15] Dionysios S Kalogerias and Warren B Powell. Zeroth-order stochastic compositional algorithms for risk-aware learning. *arXiv preprint arXiv:1912.09484*, 2019.

- [16] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.
- [17] Vijay R Konda and John N Tsitsiklis. Onactor-critic algorithms. *SIAM journal on Control and Optimization*, 42(4):1143–1166, 2003.
- [18] Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *arXiv preprint arXiv:1910.08412*, 2019.
- [19] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [20] Dhruv Malik, Ashwin Pananjady, Kush Bhatia, Koulik Khamaru, Peter L Bartlett, and Martin J Wainwright. Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. *Journal of Machine Learning Research*, 21(21):1–51, 2020.
- [21] Hesameddin Mohammadi, Armin Zare, Mahdi Soltanolkotabi, and Mihailo R Jovanović. Convergence and sample complexity of gradient methods for the model-free linear quadratic regulator problem. *arXiv preprint arXiv:1912.11899*, 2019.
- [22] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.
- [23] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.
- [24] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirota, and Marcello Restelli. Stochastic variance-reduced policy gradient. *arXiv preprint arXiv:1806.05618*, 2018.
- [25] Santiago Paternain, Juan Andrés Bazerque, Austin Small, and Alejandro Ribeiro. Stochastic policy gradient ascent in reproducing kernel hilbert spaces. *arXiv preprint arXiv:1807.11274*, 2018.
- [26] Matteo Pirota, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2-3):255–283, 2015.
- [27] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.
- [28] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.
- [29] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML’14*, page I–387–I–395. JMLR.org, 2014.
- [30] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.
- [31] Gerald Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.
- [32] Anirudh Vemula, Wen Sun, and J Andrew Bagnell. Contrasting exploration in parameter and action space: A zeroth-order optimization perspective. *arXiv preprint arXiv:1901.11503*, 2019.
- [33] Lin F Yang and Mengdi Wang. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019.
- [34] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Fred Juang. Deep reinforcement learning based resource allocation for v2v communications. *IEEE Transactions on Vehicular Technology*, 68(4):3163–3173, 2019.
- [35] Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Başar. Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383*, 2019.

## Supplementary Material

### A Proof of Lemma 2

Since the estimate  $\hat{Q}$  is unbiased, it suffices to show that  $\mathbb{E}[|\hat{Q}|^2]$  is bounded. Indeed, we have

$$\begin{aligned}\mathbb{E}[|\hat{Q}|^2] &= \mathbb{E}\left[\left|\sum_{t=0}^T R(s_t, a_t = \pi_\theta(s_t))\right|^2\right] \leq \mathbb{E}\left[\left(\sum_{t=0}^T |R(s_t, a_t = \pi_\theta(s_t))|\right)^2\right] \\ &\leq \mathbb{E}[(T+1)^2 U_R^2] = U_R^2 \frac{1+\gamma}{(1-\gamma)^2},\end{aligned}$$

where the last equality follows from the fact that  $T$  is a geometric random variable with probability of success  $1 - \gamma$ . Enough said.  $\square$

### B Proof of Theorem 1

Let Assumptions 1-3 and also Assumption 4 be in effect. Also, recall the definitions of the gradients, namely

$$\nabla J(\theta) = \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)}],$$

and

$$\hat{\nabla} J(\theta) = \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\nabla_\theta \pi_\theta(s) \nabla_a Q_\mu^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)}].$$

By definition, and then by Jensen, we have

$$\begin{aligned}\|\nabla J(\theta) - \hat{\nabla} J(\theta)\| &= \left\| \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)}] - \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\nabla_\theta \pi_\theta(s) \nabla_a Q_\mu^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)}] \right\| \\ &= \left\| \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} - \nabla_\theta \pi_\theta(s) \nabla_a Q_\mu^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)}] \right\| \\ &\leq \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} - \nabla_\theta \pi_\theta(s) \nabla_a Q_\mu^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} \right\| \right].\end{aligned}$$

Grouping terms, then applying Cauchy-Schwarz, we obtain

$$\begin{aligned}\|\nabla J(\theta) - \hat{\nabla} J(\theta)\| &\leq \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_\theta \pi_\theta(s) (\nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} - \nabla_a Q_\mu^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)}) \right\| \right] \\ &\leq \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_\theta \pi_\theta(s) \right\| \cdot \left\| \nabla_a Q^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} - \nabla_a Q_\mu^{\pi_\theta}(s, a)|_{a=\pi_\theta(s)} \right\| \right].\end{aligned}$$

By Assumption 1, we can bound  $\nabla_\theta \pi_\theta(s)$  by  $B_\Theta$ . As such, the remaining expression can be bounded by invoking Assumption 4, yielding

$$\begin{aligned}\|\nabla J(\theta) - \hat{\nabla} J(\theta)\| &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_a (Q^{\pi_\theta}(s, a) - Q_\mu^{\pi_\theta}(s, a)) \right\|_{a=\pi_\theta(s)} \right] \\ &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_a (Q^{\pi_\theta}(s, a) - \mathbb{E}_{\mathbf{u}} [Q^{\pi_\theta}(s, a + \mu \mathbf{u})]) \right\|_{a=\pi_\theta(s)} \right] \\ &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_a (\mathbb{E}_{\mathbf{u}} [Q^{\pi_\theta}(s, a) - Q^{\pi_\theta}(s, a + \mu \mathbf{u})]) \right\|_{a=\pi_\theta(s)} \right] \\ &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \mathbb{E}_{\mathbf{u}} [\nabla_a (Q^{\pi_\theta}(s, a) - Q^{\pi_\theta}(s, a + \mu \mathbf{u})]) \right\|_{a=\pi_\theta(s)} \right] \\ &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \mathbb{E}_{\mathbf{u}} \left[ \left\| \nabla_a (Q^{\pi_\theta}(s, a) - Q^{\pi_\theta}(s, a + \mu \mathbf{u})) \right\|_{a=\pi_\theta(s)} \right] \right] \\ &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} [\mathbb{E}_{\mathbf{u}} [G \|\mu \mathbf{u}\|]] \\ &\leq B_\Theta \mu \sqrt{p} G / (1 - \gamma),\end{aligned}$$

where the final inequality comes from [22, Lemma 1].

If, alternatively, Assumption 5 instead of Assumption 4, holds, then by [22, Lemma 3], we obtain

$$\begin{aligned}\|\nabla J(\theta) - \hat{\nabla} J(\theta)\| &\leq B_\Theta \mathbb{E}_{s \sim \rho^{\pi_\theta}} \left[ \left\| \nabla_a (Q^{\pi_\theta}(s, a) - Q_\mu^{\pi_\theta}(s, a)) \right\|_{a=\pi_\theta(s)} \right] \\ &\leq B_\Theta \mu^2 (p+4)^2 H / (1 - \gamma).\end{aligned}$$

This completes the proof.  $\square$

### C Proof of Lemma 3

First, by definition, we may write

$$(1 - \gamma)^2 \mathbb{E}[\|g_t\|^2] = \mathbb{E} \left[ \left\| \nabla_{\theta} \pi_{\theta_t}(s_t) \frac{\hat{Q}_t^+ - \hat{Q}_t^-}{\mu} \mathbf{u} \right\|^2 \right] =: \mathbb{E}[\|g'_t\|^2],$$

where

$$\hat{Q}_t^{\pm} = Q^{\pi_{\theta_t}}(\pm) + \frac{1}{N} \sum_{n=1}^N w_{t,n}^{\pm}.$$

By Cauchy-Schwarz, and by exploiting the above, we readily get

$$\begin{aligned} \mathbb{E}[\|g'_t\|^2] &\leq B_{\Theta}^2 \mathbb{E} \left[ \left\| \frac{Q^{\pi_{\theta_t}}(+)-Q^{\pi_{\theta_t}}(-)}{\mu} + \frac{1}{N} \sum_{n=1}^N \frac{w_{t,n}^+ - w_{t,n}^-}{\mu} \right\|^2 \|\mathbf{u}\|^2 \right] \\ &= B_{\Theta}^2 \mathbb{E} \left[ \left| \frac{Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t) + \mu \mathbf{u}) - Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t))}{\mu} + \frac{1}{N} \sum_{n=1}^N \frac{w_{t,n}^+ - w_{t,n}^-}{\mu} \right|^2 \|\mathbf{u}\|^2 \right] \\ &\leq 2B_{\Theta}^2 \mathbb{E} \left[ \left| \frac{Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t) + \mu \mathbf{u}) - Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t))}{\mu} \right|^2 \|\mathbf{u}\|^2 \right. \\ &\quad \left. + \left| \frac{1}{N} \sum_{n=1}^N \frac{w_{t,n}^+ - w_{t,n}^-}{\mu} \right|^2 \|\mathbf{u}\|^2 \right]. \end{aligned} \quad (10)$$

For the first term on the right-hand side of (10), we have

$$\mathbb{E} \left[ \left| \frac{Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t) + \mu \mathbf{u}) - Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t))}{\mu} \right|^2 \|\mathbf{u}\|^2 \right] \leq L^2 \mathbb{E}[\|\mathbf{u}\|^4] \leq L^2(p+4)^2. \quad (11)$$

For the second term on the right-hand side of (10), we may write

$$\begin{aligned} \mathbb{E} \left[ \left| \frac{1}{N} \sum_{n=1}^N \frac{w_{t,n}^+ - w_{t,n}^-}{\mu} \right|^2 \|\mathbf{u}\|^2 \right] &= \frac{1}{\mu^2 N^2} \mathbb{E} \left[ \mathbb{E} \left[ \left| \sum_{n=1}^N w_{t,n}^+ - w_{t,n}^- \right|^2 \|\mathbf{u}\|^2 \middle| s_t, \mathbf{u}, \theta_t, T_Q \right] \right] \\ &= \frac{1}{\mu^2 N^2} \mathbb{E} \left[ \|\mathbf{u}\|^2 \mathbb{E} \left[ \left| \sum_{n=1}^N w_{t,n}^+ - w_{t,n}^- \right|^2 \middle| s_t, \mathbf{u}, \theta_t, T_Q \right] \right]. \end{aligned}$$

Naturally, we need to focus on the (inner) conditional expectation above. By construction, it is true that, for every  $t$ , the differences  $w_{t,n}^+ - w_{t,n}^-$ ,  $n \in \mathbb{N}_N^+$  are *conditionally* independent and identically distributed relative to  $s, \mathbf{u}, \theta_t$  and  $T_Q$ , and *with zero mean*. This implies that

$$\begin{aligned} \mathbb{E} \left[ \left| \sum_{n=1}^N w_{t,n}^+ - w_{t,n}^- \right|^2 \middle| s_t, \mathbf{u}, \theta_t, T_Q \right] &= \sum_{n=1}^N \mathbb{E} [ |w_{t,n}^+ - w_{t,n}^-|^2 | s_t, \mathbf{u}, \theta_t, T_Q ] \\ &= N \mathbb{E} [ |w_{t,1}^+ - w_{t,1}^-|^2 | s_t, \mathbf{u}, \theta_t, T_Q ], \\ &\leq N 2 \mathbb{E} [ (w_{t,1}^+)^2 + (w_{t,1}^-)^2 | s_t, \mathbf{u}, \theta_t, T_Q ] \\ &\leq N 4 U_R^2 \frac{1 + \gamma}{(1 - \gamma)^2} =: N \sigma^2, \end{aligned}$$

where in the last line we have used Lemma 2 twice. As a result, we obtain that

$$\mathbb{E} \left[ \left| \frac{1}{N} \sum_{n=1}^N \frac{w_{t,n}^+ - w_{t,n}^-}{\mu} \right|^2 \|\mathbf{u}\|^2 \right] \leq \frac{\sigma^2 p}{\mu^2 N}, \quad (12)$$

since  $\mathbb{E}[\|\mathbf{u}\|^2] \equiv p$ . Combining (11) and (12), we end up with the uniform bound

$$\mathbb{E}[\|g_t\|^2] \leq \frac{2B_{\Theta}^2}{(1 - \gamma)^2} \left( L^2(p+4)^2 + \frac{\sigma^2 p}{\mu^2 N} \right),$$

thus completing the proof.  $\square$

## D Proof of Theorem 2

First, it is not hard to see that the stochastic ascend direction  $g_t$  constructed by Algorithm 1 is an unbiased estimate of the quasi-gradient  $\hat{\nabla} J$ . Indeed, with  $s_t \sim (1 - \gamma)\rho^{\pi_{\theta_t}}$ , we may carefully write

$$\begin{aligned}
(1 - \gamma)\mathbb{E}[g_t | \theta_t] &= \mathbb{E} \left[ \nabla_{\theta} \pi_{\theta_t}(s_t) \frac{\hat{Q}_t^+ - \hat{Q}_t^-}{\mu} \mathbf{u} \middle| \theta_t \right] \\
&= \mathbb{E} \left[ \mathbb{E} \left[ \nabla_{\theta} \pi_{\theta_t}(s_t) \frac{\hat{Q}_t^+(s_t, \mathbf{u}, \theta_t, T_Q) - \hat{Q}_t^-(s_t, \theta_t, T_Q)}{\mu} \mathbf{u} \middle| s_t, \mathbf{u}, \theta_t, T_Q \right] \middle| \theta_t \right] \\
&\equiv \mathbb{E} \left[ \nabla_{\theta} \pi_{\theta_t}(s_t) \mathbb{E} \left[ \frac{\hat{Q}_t^+(s_t, \mathbf{u}, \theta_t, T_Q) - \hat{Q}_t^-(s_t, \theta_t, T_Q)}{\mu} \middle| s_t, \mathbf{u}, \theta_t, T_Q \right] \mathbf{u} \middle| \theta_t \right] \\
&= \mathbb{E} \left[ \nabla_{\theta} \pi_{\theta_t}(s_t) \frac{Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t) + \mu \mathbf{u}) - Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t))}{\mu} \mathbf{u} \middle| \theta_t \right] \\
&= \mathbb{E} \left[ \nabla_{\theta} \pi_{\theta_t}(s_t) \mathbb{E} \left[ \frac{Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t) + \mu \mathbf{u}) - Q^{\pi_{\theta_t}}(s_t, \pi_{\theta_t}(s_t))}{\mu} \mathbf{u} \middle| s_t, \theta_t \right] \middle| \theta_t \right] \\
&= \mathbb{E} [\nabla_{\theta} \pi_{\theta_t}(s_t) \nabla_a Q_{\mu}^{\pi_{\theta_t}}(s, a) |_{a=\pi_{\theta_t}(s_t)} | \theta_t] \\
&= (1 - \gamma) \mathbb{E}_{s_t \sim \rho^{\pi_{\theta_t}}} [\nabla_{\theta} \pi_{\theta_t}(s_t) \nabla_a Q_{\mu}^{\pi_{\theta_t}}(s, a) |_{a=\pi_{\theta_t}(s_t)} | \theta_t] \\
&= (1 - \gamma) \hat{\nabla} J(\theta_t).
\end{aligned}$$

Next, using the assumption that  $J(\theta)$  is  $L_J$ -smooth, we have

$$J(\theta_{t+1}) \geq J(\theta_t) + (\theta_{t+1} - \theta_t)^\top \nabla J(\theta_t) - L_J \|\theta_{t+1} - \theta_t\|^2$$

From Algorithm 1, we have  $\theta_{t+1} - \theta_t = \alpha g_t$ . Let  $\mathbb{E}_t$  denote conditional expectation taken relative to history up to iteration  $t - 1$ . Then, by taking the expectation on both sides, we obtain

$$\mathbb{E}_t[J(\theta_{t+1})] \geq J(\theta_t) + \alpha \hat{\nabla} J(\theta_t)^\top \nabla J(\theta_t) - L_J \alpha^2 \mathbb{E}_t[\|g_t\|^2].$$

Now add and subtract  $\alpha \nabla J(\theta_t)^\top \nabla J(\theta_t)$  to the right hand side. By grouping terms, we may write

$$\mathbb{E}_t[J(\theta_{t+1})] \geq J(\theta_t) + \alpha (\hat{\nabla} J(\theta_t) - \nabla J(\theta_t))^\top \nabla J(\theta_t) + \alpha \|\nabla J(\theta_t)\|^2 - L_J \alpha^2 \mathbb{E}_t[\|g_t\|^2].$$

Using Cauchy-Schwarz, we bound the term  $(\hat{\nabla} J(\theta_t) - \nabla J(\theta_t))^\top \nabla J(\theta_t)$  by  $-\|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\| \cdot \|\nabla J(\theta_t)\|$ . Further, because  $J(\theta)$  is assumed to be  $G_J$ -Lipschitz, we have

$$\mathbb{E}_t[J(\theta_{t+1})] \geq J(\theta_t) - \alpha G_J \|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\| + \alpha \|\nabla J(\theta_t)\|^2 - L_J \alpha^2 \mathbb{E}_t[\|g_t\|^2].$$

Taking the total expectation, invoking Lemma 3, summing up until  $T$ , and dividing by  $1/(\alpha T)$ , we obtain

$$\begin{aligned}
\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla J(\theta_t)\|^2 &\leq \frac{\Delta_0}{\alpha T} + L_J \alpha \frac{2B_{\Theta}^2}{(1 - \gamma)^2} (p + 4)^2 L^2 + L_J \alpha \frac{2B_{\Theta}^2}{(1 - \gamma)^2} \frac{p\sigma^2}{\mu^2 N} \\
&\quad + \frac{G_J}{T} \sum_{t=1}^T \mathbb{E} [\|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\|],
\end{aligned} \tag{13}$$

where  $\Delta_0 = J(\theta^*) - J(\theta_0)$  and  $\theta^* \in \arg \max J(\theta)$ .

Under Assumption 4, Theorem 1 implies that

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla J(\theta_t)\|^2 \leq \frac{\Delta_0}{\alpha T} + L_J \alpha \frac{2B_{\Theta}^2}{(1 - \gamma)^2} (p + 4)^2 L^2 + L_J \alpha \frac{2B_{\Theta}^2}{(1 - \gamma)^2} \frac{p\sigma^2}{\mu^2 N} + G_J \frac{B_{\Theta} \mu \sqrt{p} G}{1 - \gamma}.$$

If, alternatively, Assumption 5 holds, then from (13) we use the pertinent result from Theorem 1 to obtain

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E} \|\nabla J(\theta_t)\|^2 \leq \frac{\Delta_0}{\alpha T} + L_J \alpha \frac{2B_{\Theta}^2}{(1 - \gamma)^2} (p + 4)^2 L^2 + L_J \alpha \frac{2B_{\Theta}^2}{(1 - \gamma)^2} \frac{p\sigma^2}{\mu^2 N} + G_J \frac{B_{\Theta} \mu^2 (p + 4)^2 H}{1 - \gamma}.$$

To complete the proof, we set the number of Monte Carlo rollouts ( $N$ ), smoothing parameter ( $\mu$ ), and step size ( $\alpha$ ) in each of the expressions above, according to Table 1. Enough said.  $\square$

**Extensions for constrained policy parameterizations:** In such a case, we may consider a projected gradient step in Algorithm 1, i.e.,

$$\theta_{t+1} = \Pi_{\Theta}\{\theta_t + \alpha g_t\},$$

where  $\Theta \subset \mathbb{R}^d$  is also assumed to be closed and convex. Below we present two approaches for deriving meaningful convergence rates.

*Noisy projection approach:* In this approach, we may expand the product  $(\theta_{t+1} - \theta_t)^\top \nabla J(\theta_t)$  into two relevant terms as

$$\begin{aligned} (\theta_{t+1} - \theta_t)^\top \nabla J(\theta_t) &= (\theta_{t+1} + \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t)^\top \nabla J(\theta_t) \\ &= (\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t)^\top \nabla J(\theta_t) \\ &\quad + (\theta_{t+1} - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\})^\top \nabla J(\theta_t). \end{aligned}$$

For the first term, we invoke the key property [9, Lemma 1]

$$(\Pi_{\Theta}\{\theta + \alpha g\} - \theta)^\top g \geq \frac{1}{\alpha} \|\Pi_{\Theta}\{\theta + \alpha g\} - \theta\|^2, \quad \forall (\theta, g) \in \Theta \times \mathbb{R}^d, \quad (14)$$

yielding

$$(\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t)^\top \nabla J(\theta_t) \geq \frac{1}{\alpha} \|\theta_t - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\}\|^2.$$

For the second term, we may further expand as

$$\begin{aligned} &(\theta_{t+1} - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\})^\top \nabla J(\theta_t) \\ &= (\theta_{t+1} - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\})^\top \nabla J(\theta_t) \\ &\quad + (\Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\})^\top \nabla J(\theta_t), \end{aligned}$$

which implies that

$$\begin{aligned} &\mathbb{E}_t\{(\theta_{t+1} - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\})^\top \nabla J(\theta_t)\} \\ &= (\mathbb{E}_t\{\theta_{t+1}\} - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\})^\top \nabla J(\theta_t) \\ &\quad + (\Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\})^\top \nabla J(\theta_t) \\ &\geq -\|\nabla J(\theta_t)\| \|\mathbb{E}_t\{\theta_{t+1}\} - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\}\| \\ &\quad - \alpha \|\nabla J(\theta_t)\| \|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\| \\ &\geq -G_J \|\mathbb{E}_t\{\theta_{t+1}\} - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\}\| \\ &\quad - \alpha G_J \|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\|. \end{aligned}$$

Consequently, we obtain the lower bound

$$\begin{aligned} \mathbb{E}_t\{(\theta_{t+1} - \theta_t)^\top \nabla J(\theta_t)\} &\geq \frac{1}{\alpha} \|\theta_t - \Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\}\|^2 \\ &\quad - \alpha G_J \|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\| \\ &\quad - G_J \|\mathbb{E}_t\{\theta_{t+1}\} - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\}\|. \end{aligned}$$

Exploiting smoothness as in the unconstrained case discussed above, we end up with the complexity estimate

$$\begin{aligned} &\frac{1}{T} \sum_{t=1}^T \mathbb{E} \frac{1}{\alpha^2} \|\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t\|^2 \\ &\leq \frac{\Delta_0}{\alpha T} + L_J \alpha \frac{2B_{\Theta}^2}{(1-\gamma)^2} (p+4)L^2 + L_J \alpha \frac{2B_{\Theta}^2}{(1-\gamma)^2} \frac{p\sigma^2}{\mu^2 N} \\ &\quad + \frac{G_J}{T} \sum_{t=1}^T \mathbb{E} [\|\hat{\nabla} J(\theta_t) - \nabla J(\theta_t)\|] \\ &\quad + \frac{G_J}{\alpha T} \sum_{t=1}^T \mathbb{E} [\|\mathbb{E}_t[\Pi_{\Theta}\{\theta_t + \alpha g_t\}] - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\}\|]. \end{aligned}$$

Note that, in the above, whenever it is actually the case that  $\theta_t + \alpha g_t \in \Theta$  *conditioned on*  $\theta_t$ , it follows that (note that  $\Theta$  is closed convex)

$$\begin{aligned}\Theta \ni \mathbb{E}_t[\Pi_\Theta\{\theta_t + \alpha g_t\}] &= \mathbb{E}_t[\theta_t + \alpha g_t] \\ &= \theta_t + \alpha \hat{\nabla} J(\theta_t),\end{aligned}$$

which implies that

$$\mathbb{E}_t[\Pi_\Theta\{\theta_t + \alpha g_t\}] - \Pi_\Theta\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} = 0.$$

This will happen when  $g_t$  is (or can be made) uniformly not too large given  $\theta_t$ , and at the same time the stepsize  $\alpha$  is sufficiently small. Exactly such a behavior has been observed in our simulations.

*Mini-batch approach:* Alternatively, we may proceed to derive a slightly more pessimistic complexity estimate by writing

$$\begin{aligned}(\theta_{t+1} - \theta_t)^\top \nabla J(\theta_t) &= (\theta_{t+1} - \theta_t)^\top (\nabla J(\theta_t) - \hat{\nabla} J(\theta_t) + \hat{\nabla} J(\theta_t)) \\ &= (\theta_{t+1} - \theta_t)^\top (\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)) + (\theta_{t+1} - \theta_t)^\top \hat{\nabla} J(\theta_t).\end{aligned}$$

For the first term, Cauchy-Schwarz trivially gives

$$(\theta_{t+1} - \theta_t)^\top (\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)) \geq -\alpha \|g_t\| \|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|.$$

For the second term, we may further expand as

$$\begin{aligned}(\theta_{t+1} - \theta_t)^\top \hat{\nabla} J(\theta_t) &= (\theta_{t+1} - \theta_t)^\top (\hat{\nabla} J(\theta_t) + g_t - g_t) \\ &= (\theta_{t+1} - \theta_t)^\top (\hat{\nabla} J(\theta_t) - g_t) + (\theta_{t+1} - \theta_t)^\top g_t,\end{aligned}$$

where again invoking (14) yields

$$(\theta_{t+1} - \theta_t)^\top g_t \geq \frac{1}{\alpha} \|\Pi_\Theta\{\theta_t + \alpha g_t\} - \theta_t\|^2,$$

whereas it also holds that

$$\begin{aligned}(\theta_{t+1} - \theta_t)^\top (\hat{\nabla} J(\theta_t) - g_t) &= (\Pi_\Theta\{\theta_t + \alpha g_t\} - \theta_t)^\top (\hat{\nabla} J(\theta_t) - g_t) \\ &= (\Pi_\Theta\{\theta_t + \alpha g_t\} - \Pi_\Theta\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\})^\top (\hat{\nabla} J(\theta_t) - g_t) \\ &\quad + (\Pi_\Theta\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} - \theta_t)^\top (\hat{\nabla} J(\theta_t) - g_t).\end{aligned}$$

In regard, to the latter expression, it is then true that

$$\begin{aligned}\mathbb{E}_t[(\theta_{t+1} - \theta_t)^\top (\hat{\nabla} J(\theta_t) - g_t)] &= \mathbb{E}_t[(\Pi_\Theta\{\theta_t + \alpha g_t\} - \Pi_\Theta\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\})^\top (\hat{\nabla} J(\theta_t) - g_t)] \\ &\quad + 0 \\ &\geq -\alpha \mathbb{E}_t[\|g_t - \hat{\nabla} J(\theta_t)\|^2].\end{aligned}$$

Combining everything, we get

$$\begin{aligned}\mathbb{E}_t[(\theta_{t+1} - \theta_t)^\top \nabla J(\theta_t)] &\geq \frac{1}{\alpha} \|\Pi_\Theta\{\theta_t + \alpha g_t\} - \theta_t\|^2 \\ &\quad - \alpha \mathbb{E}_t[\|g_t\| \|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|] \\ &\quad - \alpha \mathbb{E}_t[\|g_t - \hat{\nabla} J(\theta_t)\|^2].\end{aligned}$$

Now, note that that we can write

$$\begin{aligned}
& \frac{1}{\alpha^2} \mathbb{E}[\|\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t\|^2] \\
&= \frac{1}{\alpha^2} \mathbb{E}[\|\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} \\
&\quad + \Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} - \theta_t\|^2] \\
&\leq 2\mathbb{E}[\|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|^2] + 2\frac{1}{\alpha^2} \mathbb{E}[\|\Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} - \theta_t\|^2] \\
&= 2\mathbb{E}[\|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|^2] \\
&\quad + 2\frac{1}{\alpha^2} \mathbb{E}[\|\Pi_{\Theta}\{\theta_t + \alpha \hat{\nabla} J(\theta_t)\} - \Pi_{\Theta}\{\theta_t + \alpha g_t\} \\
&\quad + \Pi_{\Theta}\{\theta_t + \alpha g_t\} - \theta_t\|^2] \\
&\leq 2\mathbb{E}[\|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|^2] \\
&\quad + 4\mathbb{E}[\|g_t - \hat{\nabla} J(\theta_t)\|^2] \\
&\quad + 4\frac{1}{\alpha^2} \mathbb{E}[\|\Pi_{\Theta}\{\theta_t + \alpha g_t\} - \theta_t\|^2].
\end{aligned}$$

Finally, following the usual steps as above, we obtain the complexity estimate

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \frac{1}{\alpha^2} \|\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t\|^2 \\
&\leq \frac{2}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|^2] + \frac{4}{T} \sum_{t=1}^T \mathbb{E}[\|g_t - \hat{\nabla} J(\theta_t)\|^2] \\
&\quad + \frac{\Delta_0}{\alpha T} + L_J \alpha \frac{2B_{\Theta}^2}{(1-\gamma)^2} (p+4)^2 L^2 + L_J \alpha \frac{2B_{\Theta}^2}{(1-\gamma)^2} \frac{p\sigma^2}{\mu^2 N} \\
&\quad + \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|g_t\| \|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|] + \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|g_t - \hat{\nabla} J(\theta_t)\|^2],
\end{aligned}$$

or, more compactly,

$$\begin{aligned}
& \frac{1}{T} \sum_{t=1}^T \mathbb{E} \frac{1}{\alpha^2} \|\Pi_{\Theta}\{\theta_t + \alpha \nabla J(\theta_t)\} - \theta_t\|^2 \\
&\leq \frac{\Delta_0}{\alpha T} + L_J \alpha \frac{2B_{\Theta}^2}{(1-\gamma)^2} (p+4)^2 L^2 + L_J \alpha \frac{2B_{\Theta}^2}{(1-\gamma)^2} \frac{p\sigma^2}{\mu^2 N} \\
&\quad + \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\|g_t\| \|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|] + \frac{2}{T} \sum_{t=1}^T \mathbb{E}[\|\nabla J(\theta_t) - \hat{\nabla} J(\theta_t)\|^2] \\
&\quad + \frac{5}{T} \sum_{t=1}^T \mathbb{E}[\|g_t - \hat{\nabla} J(\theta_t)\|^2],
\end{aligned}$$

This bound suggests that further variance reduction can be employed in order to make the term involving  $\mathbb{E}[\|g_t - \hat{\nabla} J(\theta_t)\|^2] = \mathbb{E}[\|g_t - \mathbb{E}[g_t|\theta_t]\|^2]$  small, for instance by exploiting mini-batching, similar in spirit to [9]. However, this comes at the expense of significantly higher training overhead.

## E Complementary Numerical Results

In section 5, we presented the learning curves for our algorithm implemented on a navigation problem with and without system noise. We showed that either with or without system noise, ZDPG and ZDPG-S enjoy a substantial advantage over PG with baseline (PG-B), and of course standard PG, especially without MC variance reduction, which is of importance in various practical settings. The



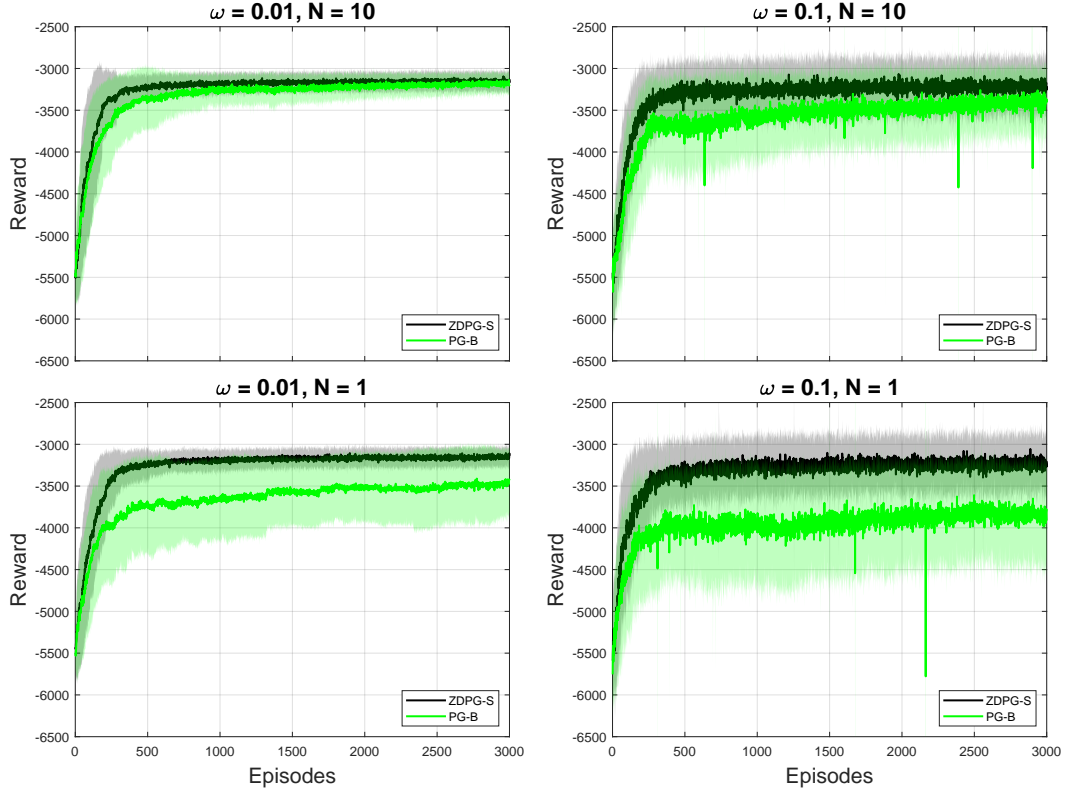


Figure 3: Average reward per episode of length  $T = 20$  with confidence bounds over 50 trials.

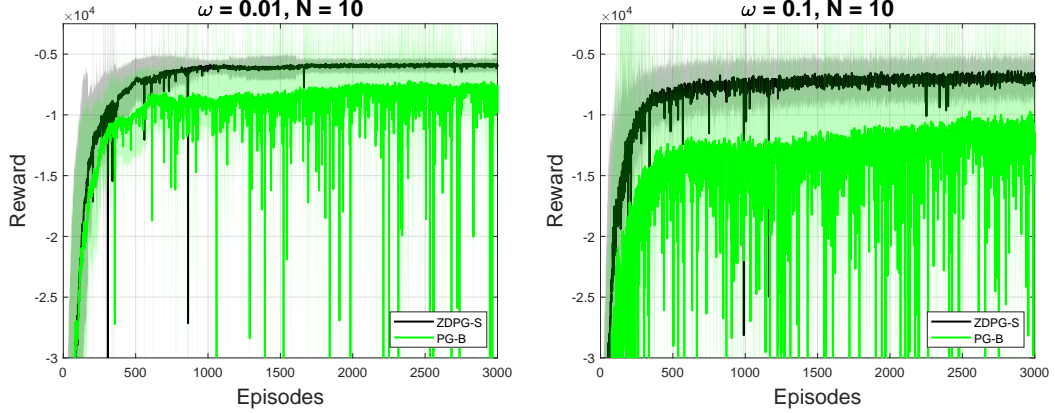


Figure 4: Average reward per episode of length  $T = 100$  with confidence bounds over 50 trials.

main purpose of this section is to confirm that the advantage of ZDPG does not disappear as the noise in the system further increases; in fact, it becomes even more drastic. We show this by comparing the learning curves on our navigation example for system noise variance  $\omega = 0.01$  and  $\omega = 0.1$ , for different values of Monte-Carlo variance reduction ( $N$ ), and policy evaluation rollout length ( $T$ ). In this set of simulations, we only compare ZDPG-S with PG-B, since these methods seem to be empirically superior to ZDPG and PG, respectively.

In Figure 3, the evaluation rollout length is set to  $T = 20$ . The smoothing parameter for ZDPG-S is  $\mu = 0.5$  for the  $\omega = 0.01$  system noise case and  $\mu = 0.6$  for the  $\omega = 0.1$  system noise case. Analogously, for PG-B, we select action variance as 0.025 for both levels of system noise as this value showed the most consistent performance throughout this line of experiments (by trial-and-error). For all simulations and for both methods, the learning rate was set to  $\alpha = 10^{-7}$ . Figure 3 (top) shows that, indeed, ZDPG-S remains advantageous over and consistently outperforms PG-B when

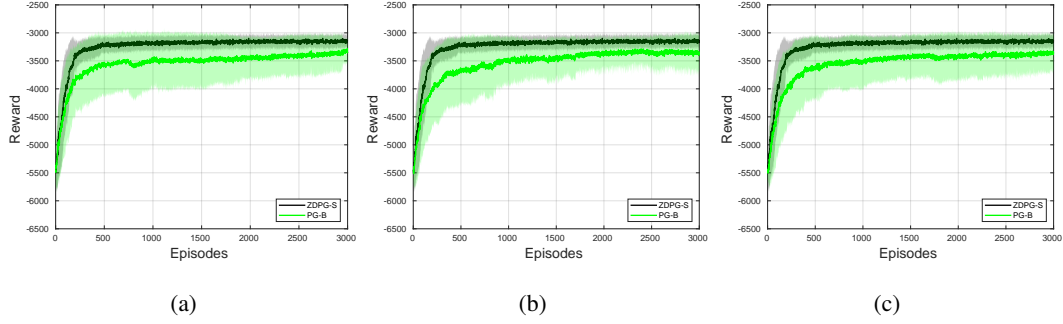


Figure 5: Average reward per episode of length  $T = 20$  with confidence bounds over 50 trials. System noise  $\omega = 0.01$  with no MC variance reduction ( $N = 1$ ). Variance reduction is applied *only* on value-function estimates to approximate the true (deterministic) baseline exclusively for PG-B, with (a) 20, (b) 200, and (c) 600 rollouts.

the system noise is increased. More interestingly, ZDPG-S shows significant advantage over PG-B in terms of noise resilience especially when there is no MC variance reduction at all (i.e.  $N = 1$ ), which, as also stated above, is a case of interest in most practical scenarios.

Figure 4 shows the learning curves for both methods when the evaluation length is set to  $T = 100$ . While the performance of both ZDPG-S and PG-B is more sensitive over such a longer evaluation horizon, the plots show that ZDPG-S experiences much more stability in solving the navigation problem for both levels of system noise. In contrast, PG-B becomes significantly unstable, resulting in trajectories that either do not necessarily reach the target (see Figure 2), or reach the target after a phase of prolonged, suboptimal wandering.

Because PG-B relies on the estimate  $(\hat{Q}(s, a) - \tilde{V}(s))$ , MC variance reduction (whenever  $N > 1$ ) is essentially applied to both parts of the involved two-point difference for the results presented in Figure 3 (and analogously for the two-point gradient estimates in ZDPG). In that way, the comparison between ZDPG and PG-B is fair for every  $N \geq 1$ . Still, it would be interesting to compare ZDPG-S with PG-B, when in the latter a more accurate version of the corresponding true value function is employed, resembling a standard (non-stochastic) PG with baseline. To do this, we conducted the same experiment as in Figure 3 for the low system noise case ( $\omega = 0.01$ ) and with no MC reduction ( $N = 1$ ), but where only the baseline  $\tilde{V}(s)$  is constructed as a *multiple-rollout* estimate of the associated value function at state  $s$  (i.e., an average). We note, though, that this procedure results in a PG-B algorithm which is much more wasteful in terms of resources, whereas at the same time its comparison with ZDPG is extremely unfair, in favor of ZDPG. This resulting plots are shown in Figure 5. It is very interesting to see that, even though the performance of PG-B slightly improves with the consideration of multiple-rollouts for the corresponding value function estimate, it still performs rather poorly compared with ZDPG-S, which of course demands just two system rollouts per episode.

Again, we would like to reiterate that the success of ZDPG comes from the non-heuristic use of deterministic policies, in contrast to random policies essential in log-trick-based standard (baseline) PG methods. At least with regards to our simulation setting, removing this level of randomness from the evolution of the system during training allows the agent to reach a solution quickly and consistently.