Sample Complexity of Estimating the Policy Gradient for Nearly Deterministic Dynamical Systems

Osbert Bastani
University of Pennsylvania, USA

Abstract

Reinforcement learning is a promising approach to learning robotics controllers. It has recently been shown that algorithms based on finite-difference estimates of the policy gradient are competitive with algorithms based on the policy gradient theorem. We propose a theoretical framework for understanding this phenomenon. Our key insight is that many dynamical systems (especially those of interest in robotics control tasks) are nearly deterministic—i.e., they can be modeled as a deterministic system with a small stochastic perturbation. We show that for such systems, finite-difference estimates of the policy gradient can have substantially lower variance than estimates based on the policy gradient theorem. Finally, we empirically evaluate our insights in an experiment on the inverted pendulum.

1 Introduction

The policy gradient is the workhorse of modern reinforcement learning. In particular, most state-of-the-art reinforcement learning algorithms aim to learn a control policy $\pi_\theta$ by estimating the policy gradient—i.e., the gradient $\nabla_\theta J(\theta)$ of the expected cumulative reward $J(\theta)$ with respect to the parameters $\theta$ of the control policy—in one of two ways: (i) numerically, e.g., using a finite-difference approximation (Kober et al., 2013; Mania et al., 2018), or (ii) by using the policy gradient theorem (Sutton et al., 2000) to construct estimates (Silver et al., 2014; Schulman et al., 2015a,b, 2017). However, there has been little work on theoretically understanding the tradeoffs between these two approaches, and our work aims to help fill this gap.

We are interested in applications to robotics control, which typically have continuous state and action spaces (Collins et al., 2005; Abbeel et al., 2007; Levine et al., 2016). For example, reinforcement learning can be used to learn controllers when the dynamics are unknown (Abbeel et al., 2007; Ross and Bagnell, 2012; Akametalu et al., 2014; Berkenkamp et al., 2017; Johannink et al., 2018). Understanding sample complexity is especially important in this application, since the goal is for robots to be able to learn based on real world experience, which can be very costly to obtain. Furthermore, having a theoretical understanding of sample complexity is important for developing safe reinforcement learning algorithms (Akametalu et al., 2014; Berkenkamp et al., 2017; Dean et al., 2018b).

We argue that near determinism is an important characteristic of dynamical systems relevant to robotics. More precisely, we study settings where the noise in the dynamics is “small” (i.e., sub-Gaussian with small constant). This setting captures robotics tasks such as grasping (Andrychowicz et al., 2018), quadcopters (Akametalu et al., 2014), walking (Collins et al., 2005), and driving (Montemerlo et al., 2008), where the dynamics are primarily deterministic but include small perturbations such as wind, friction, or slippage. We discuss this claim in detail below.

Main results. In the context of near determinism, we analyze the sample complexity of various algorithms for estimating the policy gradient $\nabla_\theta J(\theta)$. We study three algorithms: (i) an algorithm based on finite-differences, (ii) an algorithm based on the policy gradient theorem, and (iii) a model-based algorithm (i.e., it knows the system dynamics) that uses backpropagation to estimate the policy gradient. The model-based algorithm represents the best convergence rate we can hope to achieve using only random samples of the noise. We give details on these algorithms in Section 3.

Our key parameter of interest is the sub-Gaussian pa-
parameter $\sigma_\zeta$ of the system noise $\zeta$, which is small for nearly deterministic systems. Here, we also consider dependences on the estimation error $\epsilon$ and the dimension $d_\Theta$ of the parameter space; we state theorems giving dependences on all parameters in Section 4. We prove the following bounds on the sample complexity $n$ (i.e., the number of samples needed to get at most $\epsilon$ error with probability at least $1 - \delta$):

- For the model-based estimate, $n = \tilde{O}(\frac{\sigma_\zeta^2}{\epsilon^2})$.
- For the finite-differences estimate, $n = \tilde{O}(\frac{\sigma_\zeta^2d_\Theta}{\epsilon^4})$.
- For the estimate based on the policy gradient theorem, $n = \tilde{O}(1/\epsilon^2)$ and $n = \Omega(1/\epsilon)$.

Our key finding is that while both the model-based and finite-difference estimates become small as $\sigma_\zeta$ becomes small, the estimate based on the policy gradient theorem does not. Thus, for nearly deterministic dynamical systems, finite-difference algorithms perform significantly better. However, this improvement comes at a price—$n$ depends on $d_\Theta$, and furthermore quadratically more samples are needed to get to the same estimation error.

Finally, we focus on how many samples are needed to estimate the policy gradient on a single step. This understanding is already useful for applications such as safe reinforcement learning. Nevertheless, we discuss how our results connect to the problem of optimizing $J(\theta)$ in Section 4.

Motivation for near determinism. A common approach in robotics is to model the robot dynamics as deterministic (Levinson et al., 2011; Kuindersma et al., 2016). To account for stochasticity, either a stabilizing controller such as a PID controller is used (Levinson et al., 2011), or the robot’s trajectory is replanned at every step (Kwon et al., 1983; Kuindersma et al., 2016). An alternative approach is to assume that the dynamics are deterministic plus a bounded perturbation at each step, and then use robust control (Akametalu et al., 2014). Both approaches implicitly assume that the deterministic portion of the dynamics are a good approximation of the full dynamics. In general, most systems that have been successfully studied in reinforcement learning are nearly deterministic, including Atari games (Mnih et al., 2015), MuJoCo benchmarks (Todorov et al., 2012; Levine and Koltun, 2013), and simulated grasping tasks (Andrychowicz et al., 2018).

More importantly, we believe that it will be challenging to increase the sample efficiency of reinforcement learning in systems where the noise is high. Indeed, our analysis shows that noise can be greatly amplified by the dynamics, so if the noise is large, we believe there is very little hope for sample-efficient reinforcement learning. In these settings, we may need to rely on techniques such as transfer learning (Taylor and Stone, 2009), meta-learning (Finn et al., 2017), or learning to plan (Tamar et al., 2016) to achieve low sample complexity.

Related work. The theoretical work in reinforcement learning algorithms has primarily focused on $Q$-learning (Watkins and Dayan, 1992; Kearns and Singh, 2002; Kakade et al., 2003; Jin et al., 2018), especially for Markov decision processes (MDPs) with finite state and action spaces. There has been some work on understanding the sample complexity of reinforcement learning with function approximation—e.g., for fitted value iteration (Munos and Szepesvári, 2008), for fitted policy iteration (Antos et al., 2008; Lazaric et al., 2012; Farahmand et al., 2015, 2016), fitted $Q$-iteration (Tosatto et al., 2017), and the TD(0) algorithm (Dalal et al., 2018). For robotics tasks, where state and action spaces are typically continuous, the most successful approaches are predominantly based on policy gradient estimation (Collins et al., 2005; Kober et al., 2013), for which there has been relatively little work. In this direction, (Kakade et al., 2003) has analyzed the sample complexity of algorithms based on the policy gradient theorem, but they do not study the dependence of the sample complexity on the magnitude of the system noise. Furthermore, their work assumes finite state and action spaces and bounded rewards, and they do not consider finite-difference algorithms.

There has been work characterizing a key design choice of finite-difference algorithms—i.e., the distribution of perturbations used to numerically estimate the policy gradient (Roberts and Tedrake, 2009). They measure the performance of different choices using the signal-to-noise ratio. In contrast, our goal is to understand the sample complexity of different algorithms for nearly deterministic systems.

There has recently been work on understanding the sample complexity of learning controllers; however, they focus on linear dynamical systems, and on different algorithms—e.g., temporal difference learning (Tu and Recht, 2018b) or model-based algorithms (Dean et al., 2018a; Tu and Recht, 2018a). There has also been work in this setting studying the possibility of reducing variance by controlling the noise in the dynamics (Malik et al., 2019); in the setting we study, we cannot control the noise.

There has been recent work comparing approaches based on exploration in the action space (based on the policy gradient theorem) to exploration in the state space (based on finite difference methods) (Venula
et al., 2019). Our focus on nearly deterministic systems enables us to obtain qualitatively different insights compared to theirs. In particular, they find that approaches based on finite differences perform better for problems with a long time horizon. However, we analyze a more realistic model, and find that this insight no longer holds. Instead, approaches based on finite differences outperform approaches based on the policy gradient theorem for nearly deterministic systems.

Our analysis differs in three key ways. First, they assume an upper bound \( J(\theta) \leq J_{\text{max}} \), which is a very strong assumption. Second, their analysis does not model stochastic dynamics. Instead, they assume that \( J(\theta) \) is deterministic, but they can only obtain observations \( J(\theta) + \zeta \), where \( \zeta \) is i.i.d. noise. In contrast, our analysis considers both stochastic dynamics, as well as how noise is propagated through the dynamics. This distinction substantially complicates our analysis, but is necessary for us to understand the implications of near determinism (since we need to understand how the dynamics can amplify noise). Finally, unlike their work, we provide lower bounds for our main results.

**Connection to optimizing \( J(\theta) \).** Estimating the policy gradient can be used in conjunction with stochastic gradient descent to optimize \( J(\theta) \). There is a large body of work on understanding the convergence rate of stochastic gradient descent (Robbins and Monro, 1951; Spall et al., 1992; Bottou and Bousquet, 2008; Moulines and Bach, 2011), of which policy gradient algorithms are a special case. Indeed, (Vemula et al., 2019) uses these techniques to bound the complexity of optimizing \( J(\theta) \).

There are several reasons why we focus on understanding the sample complexity of a single gradient step rather than the sample complexity of optimization. First, they rely on the strong assumption that \( J(\theta) \) is bounded—i.e., \( J(\theta) \leq J_{\text{max}} \) for some \( J_{\text{max}} \in \mathbb{R}_+ \). Second, it would be much more difficult to derive lower bounds on optimization—existing lower bounds are for the setting where the objective \( f \) coming from a very general function family, and these bounds do not apply when \( f \) is restricted to be the objective of a reinforcement learning problem. In contrast, for sample complexity, we derive matching (or almost matching) upper and lower bounds. Third, the sample complexity of estimating \( \nabla \theta J(\theta) \) is of intrinsic interest—for example, it is an important prerequisite for safe reinforcement learning algorithms (Akametalu et al., 2014; Berkenkamp et al., 2017; Dean et al., 2018b). Finally, focusing on sample complexity simplifies our key insight. In particular, consider the the completely deterministic setting—optimizing a deterministic function using gradient descent may still take many steps, but “estimating” the gradient only requires a single sample.

Additionally, we note that sample complexity is directly related to the complexity of optimizing \( J(\theta) \). In particular, the bounds in Vemula et al. (2019) all depend directly on the variance \( \sigma^2 \) of the observations \( J(\theta) + \zeta \). Our proof bounds the sample complexity of estimating \( \nabla J(\theta) \) by bounding the sub-Gaussian parameter of \( J(\theta) \), which is an upper bound on the variance of \( J(\theta) \). Thus, smaller sample complexity translates to smaller complexity of optimizing \( J(\theta) \).

Finally, our focus on estimating the gradient does not address the problem of exploration. In terms of optimization, gradient estimates can be used in conjunction with gradient descent to efficiently find local minima, whereas exploration is needed to find global minima. Understanding the sample complexity of exploration is an important but orthogonal problem that we leave to future work.

## 2 Preliminaries

We consider a dynamical system with states \( S \subseteq \mathbb{R}^{d_S} \), actions \( A \subseteq \mathbb{R}^{d_A} \), and transitions

\[
s_{t+1} = f(s_t, a_t) + \zeta_t \quad \text{where} \quad \zeta_t \sim p(\zeta),
\]

where \( f : S \times A \to S \) is deterministic and \( \zeta \in \mathbb{R}^{d_\zeta} \) is a random perturbation. We consider deterministic control policies \( \pi_\theta : S \to A \) with parameters \( \theta \in \Theta \subseteq \mathbb{R}^{d_\theta} \). Except in the case of the model-based policy gradient algorithm, we assume that both \( f \) and \( p \) are unknown.

We separate \( f \) from \( p \) since we are interested in settings where \( \zeta \) is small. Also, we that assume \( \zeta_i \) is independent of \( s_t \) and \( a_t \). This assumption enables us to substantially simplify the model-based policy gradient (since we avoid taking a derivatives of \( p \)), and it also simplifies our analyses of other algorithms.

We are interested in controlling the system over a finite horizon \( T \in \mathbb{N} \)—given a reward function \( R : S \times A \to \mathbb{R} \), the goal is to find the policy \( \pi_\theta \) that maximizes the expected cumulative reward

\[
J(\theta) = \mathbb{E}_{p_\theta(\alpha)} \left[ \sum_{t=0}^{T-1} R(s_t, a_t) \right],
\]

where \( p_\theta(\alpha) \) is the distribution over rollouts \( \alpha = ((s_0, a_0), \ldots, (s_{T-1}, a_{T-1})) \) when using \( \pi_\theta \), and where we assume the initial state \( s_0 \in S \) is deterministic and known. Note that \( \alpha \) is determined by \( \theta \) and \( \zeta \), so an expectation over \( p_\theta(\alpha) \) is equivalent to one over \( p(\zeta) \). We are interested in estimating the policy gradient

\[
D(\theta) = \nabla \theta J(\theta)
\]
so we can perform gradient ascent on $\theta$. As usual, let

$$Q^t_\theta(s, a) = \mathbb{E}_{\pi(\zeta)} \left[ R(s, a) + V^t_{\theta}(f(s, a) + \zeta) \right],$$

$$V^t_{\theta}(s) = Q^t_{\theta}(s, \pi_\theta(s)),$$

for $t \in \{0, 1, ..., T-1\}$, where $V^T_{\theta}(s) = 0$, denote the $Q$ function and value function, respectively (Sutton and Barto, 2018). In particular, $J(\theta) = V^0_{\theta}(s_0)$.

**Remark 2.1.** Our results straightforwardly extend to dynamical systems with time varying dynamics and rewards. Also, we can relax our assumption that the initial state $s_0$ is deterministic—i.e., to handle an initial state distribution $p_0$, we can modify the dynamics on the first step to be $s_1 = s_0 + \zeta_0$, where $s_0 = 0$ and $\zeta_0 \sim p_0$. Furthermore, our results can be extended to the case where the noise $\zeta$ appears nonlinearly in the transitions, as long as it can be reparameterized (Kingma and Welling, 2014)—i.e., the transitions can be written in the form $s' = f(s, \zeta, a)$, where $\zeta \sim p(\zeta)$ i.i.d. for some $p(\zeta)$. We require that $f$ is Lipschitz in $\zeta$. Most kinds of noise considered in practice can be expressed in this form, though it may not satisfy the Lipschitz condition. Finally, our results can be extended to handle Martingale difference noise sequences by using the Azuma-Hoeffding inequality in place of the Hoeffding inequality.

### 3 Policy Gradient Algorithms

We now describe the policy gradient estimation algorithms that we consider.

**Model-based policy gradient.** When $f$ is known, we can estimate the policy gradient as

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\pi(\zeta)} \left[ \nabla_{\theta} \tilde{J}(\theta; \tilde{\zeta}) \right],$$

$$\tilde{J}(\theta; \tilde{\zeta}) = \sum_{t=0}^{T-1} R(s_t, a_t),$$

since a rollout $\alpha$ is determined by $\tilde{\zeta}$. In particular, we have estimator $\nabla_{\theta} J(\theta) \approx \bar{D}_{\text{MB}}(\theta)$, where

$$\bar{D}_{\text{MB}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \tilde{J}(\theta; \tilde{\zeta}^{(i)})$$

where $\tilde{\zeta}^{(i)} \sim p(\tilde{\zeta})$ i.i.d. for $i \in [n]$.

**Policy gradient theorem.** The policy gradient theorem is formulated for stochastic policies—i.e., $\pi_\theta(a \mid s)$ is the probability of taking action $a$ in state $s$. We assume a distribution $p_\zeta(\xi)$ of action perturbations that does not depend on $\theta$—i.e., $a_t = \pi_\theta(s_t) + \xi_t$, where $\xi_t \sim p_\zeta(\xi)$, then we have $\bar{p}_\theta(a \mid s) = p_\zeta(a - \pi_\theta(s))$.

The following are the modified $Q$ and value functions:

$$\tilde{Q}^t_{\theta}(s, a) = R(s, a) + \mathbb{E}_{\pi(\zeta)} \left[ \tilde{V}^{t+1}_{\theta}(f(s, a) + \zeta) \right],$$

$$\tilde{V}^t_{\theta}(s) = \mathbb{E}_{\pi_\theta(s)} \left[ \tilde{Q}^t_{\theta}(s, a) \right],$$

where $\tilde{V}^T_{\theta}(s) = 0$ as before. Then, the following is the policy gradient theorem (Sutton et al., 2000):

**Theorem 3.1.** Letting $\bar{\pi}_\theta(\alpha)$ be the distribution over rollouts when using $\pi_\theta$, we have

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\bar{\pi}_\theta(\alpha)} \left[ \sum_{t=0}^{T-1} \tilde{Q}^t_{\theta}(s_t, a_t) \nabla_{\theta} \log \bar{\pi}_\theta(a_t \mid s_t) \right].$$

The key challenge to using Theorem 3.1 to estimate $\nabla_{\theta} J(\theta)$ is to estimate $\tilde{Q}^t_{\theta}(s_t, a_t)$. The simplest approach is to estimate it using a single rollout (Williams, 1992):

$$\tilde{Q}^t_{\theta}(s, a) = \mathbb{E}_{\bar{\pi}_\theta(\alpha)} \left[ \tilde{Q}^t_{\theta}(s, a) \right],$$

$$\tilde{Q}^t_{\theta}(s, a) = \sum_{i=1}^{T-1} R(s_t, a_t).$$

A common technique to reduce variance is to normalize $\tilde{Q}^t_{\theta}(s, a)$ by subtracting the value function (Schulman et al., 2015b). In particular, the advantage function $\tilde{A}^t_{\theta}(s, a) = \tilde{Q}^t_{\theta}(s, a) - \tilde{V}^t_{\theta}(s)$ measures the advantage of using action $a$ instead of using $\bar{\pi}_\theta$ in state $s$ at time $t$. Then, we have

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\bar{\pi}_\theta(\alpha)} \left[ \sum_{t=0}^{T-1} \tilde{A}^t_{\theta}(\alpha) \nabla_{\theta} \log \bar{\pi}_\theta(a_t \mid s_t) \right].$$

Unlike $\tilde{Q}^t_{\theta}(s, a)$, we cannot estimate $\tilde{A}^t_{\theta}(s, a)$ using a single rollout. One approach is to estimate $f^t_{\theta}(s) \approx \tilde{V}^t_{\theta}(s)$, and then estimate $\tilde{A}^t_{\theta}(s, a)$ using $f^t_{\theta}(s)$. We assume that our estimate of $\tilde{V}^t_{\theta}(s)$ is exact—in particular, we consider the following estimator $\nabla_{\theta} J(\theta) \approx \bar{D}_{\text{PG}}(\theta)$:

$$\bar{D}_{\text{PG}}(\theta) = \frac{1}{n} \sum_{t=0}^{T-1} \sum_{i=1}^{n} \tilde{A}^t_{\theta}(\alpha^{(i)}) \nabla_{\theta} \log \bar{\pi}_\theta(a_t^{(i)} \mid s_t^{(i)}),$$

$$\tilde{A}^t_{\theta}(\alpha^{(i)}) = \tilde{Q}^t_{\theta}(\alpha^{(i)}) - \tilde{V}^t_{\theta}(s_t^{(i)}),$$

where $\alpha^{(i)} \sim p_\theta(\alpha)$ i.i.d. for each $i \in [n]$.

**Remark 3.2.** A common approach is to use an estimator $\hat{Q}^t_{\theta}(s, a)$ of the $Q$ function in place of $\tilde{Q}^t_{\theta}(s, a)$. This approach reduces variance, but may introduce bias. For instance, for dynamical systems with continuous actions, the deterministic policy gradient (DPG) algorithm uses this approach Silver et al. (2014). We
consider the algorithm described above for two reasons. First, our focus is on estimating the policy gradient, rather than understanding the sample complexity of Q-learning, which is required to analyze DPG. Second, it is hard to prove bounds for DPG since it relies on the derivative of the Q function, which cannot be bounded without additional assumptions. For example, suppose we train a random forest \( \hat{Q}_\theta^{(i)}(s, a) \). Even if this model achieves good accuracy, its gradient would be zero nearly everywhere since this model is piecewise constant; thus, it would not be useful in the context of the DPG algorithm.

**Finite-difference policy gradient.** We can use finite-differences to estimate \( \nabla_\theta J(\theta) \).

**Theorem 3.3.** For any \( f: \mathcal{X} \to \mathbb{R} \) (where \( \mathcal{X} \subseteq \mathbb{R}^d \)) where \( \nabla f \) is \( L_{\nabla f} \)-Lipschitz continuous, \(^1\)

\[
\nabla f(x) = \sum_{k=1}^{d} \frac{f(x + \lambda \nu^{(k)}) - f(x - \lambda \nu^{(k)})}{2\lambda} \cdot \nu^{(k)} + \Delta
\]

where \( \nu^{(k)} = \delta_k \) (where \( \delta_k \) is the Kronecker delta), and \( \Delta \in \mathbb{R} \) satisfies \( ||\Delta|| \leq L_{\nabla f} d\lambda \). We give a proof in Appendix E. Then, the finite difference approximation of the policy gradient is

\[
\nabla_\theta J(\theta) \approx \sum_{k=1}^{d} \frac{J(\theta + \lambda \nu^{(k)}) - J(\theta - \lambda \nu^{(k)})}{2\lambda} \cdot \nu^{(k)}.
\]

We can estimate \( J(\theta) \) using samples \( \tilde{\zeta} \sim p(\tilde{\zeta}) \), which yields the estimator \( \nabla_\theta J(\theta) \approx \hat{D}_{FD}(\theta) \), where

\[
\hat{D}_{FD}(\theta) = \sum_{k=1}^{d} \left[ \frac{1}{n} \sum_{i=1}^{n} \tilde{J}(\theta + \lambda \nu^{(k)}; \tilde{\zeta}^{(k,i)}) - \frac{1}{n} \sum_{j=1}^{n} \tilde{J}(\theta - \lambda \nu^{(k)}; \tilde{\zeta}^{(k,j)}) \right] \cdot \nu^{(k)}
\]

where \( \tilde{\zeta}^{(k,i)}, \tilde{\zeta}^{(k,j)} \sim p(\tilde{\zeta}) \) i.i.d. for \( k \in [m] \) and \( i, j \in [n] \). Note that we use separate samples \( \zeta^{(k,i)} \) and \( \zeta^{(k,j)} \) to estimate \( J(\theta + \lambda \mu^{(k)}) \) and \( J(\theta - \lambda \mu^{(k)}) \), respectively. If we are using a simulator, then we can reduce variance by using the same samples to estimate both terms.

**Remark 3.4.** Typically, rather than choose a fixed set of basis vectors \( \mu^{(1)}, \ldots, \mu^{(k)} \), finite-difference algorithms choose random vectors from a spherically symmetric distribution—e.g., \( \nu \sim N(0, \sigma^2 I_{d\theta}) \) (Spall et al., 1992; Mania et al., 2018). Our choice of a fixed basis simplifies our analysis.

### 4 Main Results

**Sample complexity.** Recall that the policy gradient \( \nabla_\theta J(\theta) \) must be estimated from sampled rollouts \( \zeta \sim p_\theta(\zeta) \). Our goal is to understand the tradeoffs in sample complexity of estimating \( \nabla_\theta J(\theta) \) between various different reinforcement learning algorithms.

**Definition 4.1.** Let \( X \) be a random vector, and let \( p_X^{(n)} = n^{-1} \sum_{i=1}^{n} x^{(i)} \), where \( x^{(1)}, \ldots, x^{(n)} \sim p_X(x) \) i.i.d. The sample complexity of \( n_X(\epsilon, \delta) \) of \( X \) is the smallest \( n \in \mathbb{N} \) such that

\[
\Pr_{x^{(1)}, \ldots, x^{(n)} \sim p_X(x)} \left[ \| p_X^{(n)} \| \leq \epsilon \right] \leq \delta.
\]

We are interested in the sample complexity \( n_D(\epsilon, \delta) \) of \( \hat{D}(\zeta) - \nabla_\theta J(\theta) \), where \( \hat{D}(\zeta) \) is an estimate of \( \nabla_\theta J(\theta) \) using a single rollout \( \zeta \sim p_\theta(\zeta) \).

**Assumptions.** We let \( f_\theta(s) = f(s, \pi_\theta(s)) \) and \( R_\theta(s) = R(s, \pi_\theta(s)) \). Similarly, for a stochastic policy \( \pi_\theta(s) + \xi \) (where \( \xi \sim p(\xi) \)), we let \( f_\theta(s, \xi) = f(s, \pi_\theta(s) + \xi) \) and \( R_\theta(s) = \mathbb{E}_{\xi \sim p(\xi)} [R(s, \pi_\theta(s) + \xi)] \). Next, to ensure convergence, we make regularity assumptions about the dynamics and our control policy; see Appendix F & G for definitions.

**Assumption 4.2.** We assume that \( f, R, \pi_\theta, f_\theta, \hat{f}_\theta, R_\theta \) are Lipschitz continuous and are twice continuously differentiable with Lipschitz continuous first derivative.

**Remark 4.3.** This standard assumption is needed to ensure that we can estimate the gradient using finite differences. It is somewhat strong—e.g., it rules out commonly used quadratic rewards. In practice, the state space is often compact, in which case the Lipschitz continuity assumption becomes redundant. However, we cannot handle discontinuous rewards or dynamics (including piecewise constant rewards). In these cases, the policy gradient may diverge near the discontinuities; thus, the sample complexity of estimating this gradient may diverge as well. In principle, we could handle discontinuities as long as the policy visits these discontinuities with zero probability.

Finally, for any function \( h \), we let \( L_h \) denote its Lipschitz constant and \( \bar{L}_h = \max \{ L_{\nabla h}, L_h, 1 \} \).

**Assumption 4.4.** We assume that \( p(\xi) \) is \( \sigma_\xi \)-subgaussian.

This assumption is required for proving concentration—e.g., it is typically assumed in the context of safe reinforcement learning (Akametalu et al., 2014; Berkenkamp et al., 2017). In practice, perturbations due to noise are often bounded (which implies the noise is sub-Gaussian), especially for our setting of interest—e.g., forces due to wind.
friction, or slippage have bounded magnitude. We are interested in settings where \( \sigma_{\xi} \) is small.

**Definition 4.5.** A system is nearly deterministic if \( \sigma_{\xi} \ll 1 \).

In particular, we are interested in the dependence of the sample complexity on \( \sigma_{\xi} \).

**Main theorems.** For the model-based policy gradient, we have:

**Theorem 4.6.** For \( \delta \leq 1/2 \), the sample complexity of \( \tilde{D}_{MB}(\theta) - \nabla_{\theta} J(\theta) \) satisfies

\[
\sqrt{n_{MB}(\epsilon, \delta)} = \tilde{O} \left( T^{8} L_{R_{\alpha}} L_{f_{\alpha}}^{6} \sigma_{\xi} d_{A}/\epsilon \right)
\]

\[
\sqrt{n_{MB}(\epsilon, \delta)} = \Omega \left( T^{L_{f_{\alpha}}^{-3}} \sigma_{\xi}/\epsilon \right).
\]

For the policy gradient based on Theorem 3.1:

**Theorem 4.7.** For the choice \( p_{\xi}(\xi) = \mathcal{N}(\xi \mid 0, \sigma_{\xi}^{2} I_{d_{A}}) \), \( D_{PG}(\theta) - \nabla_{\theta} J(\theta) \) has sample complexity

\[
\sqrt{n_{PG}(\epsilon, \delta)} = \tilde{O} \left( T^{6}(L_{R} + L_{R_{\alpha}}) L_{f_{\alpha}} L_{\pi}^{T} d^{4}/\epsilon \right),
\]

where \( d = \max\{d_{S}, d_{A}\} \), for \( \epsilon \) sufficiently small—i.e., \( \epsilon = \Omega(T^{6}(L_{R} + L_{R_{\alpha}}) L_{f_{\alpha}} L_{\pi}^{T} d^{4}) \). Next,

\[
\sqrt{n_{PG}(\epsilon, \delta)} \geq \sqrt{n_{\xi}(\epsilon/L_{f_{\alpha}}^{-2}, \delta)}
\]

\[
\sqrt{n_{PG}(\epsilon, \delta)} = \Omega \left( \min \left\{ L_{f_{\alpha}}^{-2}/\epsilon^{1/2}, 1/\delta^{1/2} \right\} \right).
\]

The first lower bound holds for any \( p_{\xi}(\xi) \) that is everywhere differentiable on \( \mathbb{R} \) and satisfies 
\( \lim_{\xi \rightarrow \pm \infty} \xi \cdot p_{\xi}(\xi) = 0 \), where \( n_{\xi} \) is the sample complexity of estimating \( \mathbb{E}_{p_{\xi}(\xi)}[\xi \cdot \nabla_{\xi} \log p_{\xi}(\xi)] \) using samples from \( p_{\xi} \). The second lower bound holds for \( p_{\xi}(\xi) = \mathcal{N}(0, \sigma_{\xi}^{2}) \), for any \( \sigma_{\xi} \in \mathbb{R}_{+} \).

We have shown two lower bounds—one for an arbitrary distribution \( p_{\xi} \) (in terms of a sample complexity \( n_{\xi} \) related to \( p_{\xi} \)), and one for the specific choice where \( p_{\xi} \) is Gaussian (as is the case in our upper bound).

Note, also that our upper bound depends on choosing the action noise to have variance \( \sigma_{\xi} \). In principle, the first lower bound holds even if \( p_{\xi} \) depends on the problem parameters; however, then \( n_{\xi} \) may depend on these parameters as well. The second lower bound is independent of the the action noise \( \sigma_{\xi} \), so it holds even if \( \sigma_{\xi} \) depends on the problem parameters.

**Remark 4.8.** Note that the upper and lower bounds have a gap on the order of \( \epsilon^{1/2} \). We believe that this gap is due to limitations in our analysis. In particular, our lower bounds depend on a lower bound on the tail of the \( \chi_{n}^{2} \) distribution, which has exponential tails. In contrast, our other lower bounds depend on Gaussian tails, which are doubly exponential. Intuitively, since the \( \chi_{n}^{2} \) distribution has a longer tail, it should not have lower sample complexity.

**Remark 4.9.** Note that the second lower bound contains a dependence on \( \delta^{-1/2} \), which is unusual. However, this term only has a role if the first term in the minimum is very large. Furthermore, the first term depends as usual on \( \log(1/\delta) \) (which is not shown since we omit log factors).

**Remark 4.10.** Actor-critic approaches reduce variance by using function approximation to obtain lower variance estimates of the advantage \( \hat{A}_{\theta}^{(t)} \) (Schulman et al., 2015b). However, our lower bounds hold even if the advantage is known exactly. Thus, while actor-critic approaches can reduce variance, they do not affect our main insight that these estimates remain noisy for nearly deterministic dynamical systems.

For the finite-difference policy gradient:

**Theorem 4.11.** The sample complexity of \( \tilde{D}_{FD}(\theta) - \nabla_{\theta} J(\theta) \) satisfies

\[
\sqrt{n_{FD}(\epsilon, \delta)} = \tilde{O} \left( T^{9} L_{R_{\alpha}} L_{f_{\alpha}}^{5} \sigma_{\xi} d_{A} \sqrt{d_{\phi}/\epsilon^{2}} \right)
\]

\[
\sqrt{n_{FD}(\epsilon, \delta)} = \Omega \left( T^{3} L_{f_{\alpha}}^{5} \sigma_{\xi} d_{\phi}/\epsilon^{2} \right).
\]

The first bound (i.e., the upper bound) holds for a choice \( \lambda = O(\epsilon/T^{5} L_{R_{\alpha}}^{5} L_{f_{\alpha}} d_{A}) \). The second bound (i.e., the lower bound) holds for any \( \lambda \in \mathbb{R}_{+}^{T} \), \( \epsilon \leq 1 \), and \( \delta \leq 1/2 \).

Note that our upper bound is for the choice \( \lambda = O(\epsilon) \), but our lower bound holds for arbitrary \( \lambda \).

**Remark 4.12.** In an abuse of notation, in Theorem 4.11, we have ignored the fact that \( n_{FD} \) must always be at least \( 2d_{\phi} \); in particular, it does not go to zero as \( \sigma_{\xi} \) goes to zero. This discrepancy in Theorem 4.11 arises because there is an implicit assumption that we use when inverting Hoeffding’s inequality that \( n \geq 1 \)—more precisely, Hoeffding’s inequality gives a bound of the form

\[
\Pr[\hat{\mu}_{n}^{(n)} \geq \epsilon] \leq \delta,
\]

where \( \hat{\mu}_{n}^{(n)} \) is an estimate of \( \mu_{X} = \mathbb{E}[X] \) using \( n \) samples, and \( \delta \geq e^{-n\sigma^{2}/(2\epsilon^{2})} \). Solving for \( n \) yields \( n \geq 2\sigma^{2} \log(1/\delta)/\epsilon^{2} \). However, if \( \sigma = 0 \), then \( \delta \) is not well defined, so it does not mean we can get an estimate of \( \mu_{X} \) using \( n = 0 \) samples; instead, we need to take \( n = 1 \). In our proof of Theorem 4.11, we apply Hoeffding’s inequality \( 2d_{\phi} \) times (since we estimate the gradient of each component separately), so we need \( n \geq 2d_{\phi} \).

**Proof strategy.** We give a high-level overview of our proof strategy, focusing on Theorem 4.6. Our proof
proceeds in two steps. First, we prove an upper bound
\[ |\hat{D}_{MB}(\theta) - \nabla_{\theta} J(\theta)| \leq AE + B, \] (1)
where \( E = T^{-1} \sum_{t=0}^{T-1} \| \xi_t \| \) and \( A, B \in \mathbb{R}_+ \) do not depend on \( \xi \). This step uses induction based on the recursive structure of \( V_\phi \). Second, we prove Lemma G.7; we state a simplified version:

**Lemma 4.13.** Let \( X \) be a \( \sigma_X \)-sub-Gaussian random vector over \( \mathbb{R}^d \), and let \( Y \) be a random vector over \( \mathbb{R}^d \) satisfying \( \| Y \| \leq A \| X \| + B \), where \( A, B \in \mathbb{R}_+ \). Then \( Y \) is \( \sigma_Y \)-sub-Gaussian, where \( \sigma_Y = \bar{O}(A\sigma_X + B) \).

Combined with (1), we conclude that \( \hat{D}_{MB}(\theta) - \nabla_{\theta} J(\theta) \) is sub-Gaussian, from which we can use Hoeffding’s inequality (see Lemma G.3) to complete the proof. For the lower bound, we construct a system where \( J(\theta) \) is Gaussian. The proof of Theorem 4.7 follows similarly, except we need to use analogous results for sub-exponential random variables. In particular, we prove Lemma H.7, an analog of Lemma G.7. The proof of Theorem 4.11 also follows similarly, but we need to account for the bias in the finite-difference estimate of \( \nabla_{\theta} J(\theta) \) from Theorem 3.3.

5 Discussion

**Dependence on \( \sigma_\xi \).** Both \( n_{MB} \) and \( n_{FD} \) scale linearly in \( \sigma_\xi \). Thus, the corresponding algorithms perform very well when \( \sigma_\xi \) is small. In contrast, \( n_{PG} \) does not become small when \( \sigma_\xi \) becomes small. Intuitively, if \( p_\xi \) is wide, then the action noise adds uncertainty to \( D_{PG}(\theta) \). On the other hand, if \( p_\xi \) is narrow, then \( \nabla_{\theta} \log \tilde{p}_\xi(a|s) = \nabla_{\theta} \log p_\xi(a|s) \) becomes large—in particular, \( p_\xi \) must change rapidly for some values of \( \xi \), and must have large gradient at such values of \( \xi \).

A key point is that in the first lower bound for \( n_{PG} \) (i.e., for arbitrary \( p_\xi \)), even though we do not know its explicit dependence on \( \epsilon, \delta, T, \) and \( L_{f_0} \), we know that it is completely independent of \( \sigma_\xi \). Thus, regardless of how \( p_\xi \) is chosen (e.g., even if it chosen based on the parameter problems), the sample complexity does not become small as \( \sigma_\xi \) becomes small.

**Full determinism (\( \sigma_\xi = 0 \)).** When \( \sigma_\xi = 0 \), we have \( n_{MB} = 1 \) (i.e., we only need a single sample to estimate \( \nabla_{\theta} J(\theta) \)) and \( n_{FD} = 2d_\phi \) (i.e., we need two samples to estimate the derivative of each parameter, taking \( \lambda \) small enough to get \( \epsilon \) error). For the case of \( n_{PG} \), our lower bound in Theorem 4.7 still holds—the dynamical system we use to obtain the lower bound has no noise in the dynamics. In particular, a large number of samples are still needed to obtain good estimates (i.e., possibly exponential in \( T \)).

**Dependence on \( \epsilon \).** Both \( n_{MB} \) and \( n_{PG} \) depend quadratically on \( \epsilon \) (ignoring the gap between the upper and lower bounds for \( n_{PG} \)). In contrast, \( n_{FD} \) depends quartically on \( \epsilon \). This gap arises because according to Theorem 3.3, the finite-differences error of \( \hat{D}_{FD}(\theta) \) (assuming there is no noise) depends linearly on \( \lambda \). Thus, we must choose \( \lambda = O(\epsilon) \) to obtain error at most \( \epsilon \). If the dynamical system and control policy are both linear, then this error goes away, so the dependence on \( \epsilon \) becomes quadratic.

**Dependence on \( d_\phi \).** Only \( n_{FD} \) depends on \( d_\phi \)— whereas the other two algorithms make use of the fact that we can compute \( \nabla_{\theta} \pi_\theta \), the finite-difference approximation ignores this ability.

**Dependence on \( T \).** All of the sample complexities depend exponentially on \( T \). As we show in our lower bounds, this dependence is unavoidable—it arises from the fact that the dynamics cause the state (and therefore the rewards) to grow exponentially large in \( T \). A common assumption made in prior work is that the rewards are bounded uniformly by \( R_{max} \in \mathbb{R}_+ \) (Kearns and Singh, 2002; Kakade et al., 2003). Intuitively, our results indicate that without stronger assumptions, \( R_{max} \) may be exponentially large. In practice, rewards for continuous control tasks are often quadratic, and can indeed be exponentially in magnitude.

An important aspect is that estimation is substantially easier when the current policy is good. In our bounds, the base of the exponential dependence is always \( L_{f_0} \). If the initial policy \( \pi_\theta \) provides relatively stable control, then we may expect that \( L_{f_0} \leq 1 \)—i.e., the states remain bounded in magnitude. Then, we have \( L_{f_0} = 1 \), so our bounds no longer depend exponentially on \( T \). This insight suggests the importance of good initialization for fast estimation.

Indeed, policy gradient estimators can have high variance in practice. As an example, consider the cart-pole problem with continuous action space, with random initial state and where the reward function is the negative distance to origin. We empirically estimated that the MSE of the model-based policy gradient estimator using \( n = 1 \) on a randomly initialized policy for this benchmark is \( 3.5 \times 10^7 \). This error is substantially reduced when the policy is stable—for a trained cart-pole policy, we estimate that the MSE of the model-based policy gradient estimator is just \( 5.2 \times 10^{-2} \).

6 Experiments

We empirically evaluated the effect of \( \sigma_\xi \) on the performance of the different algorithms.

**Dynamical system.** We use the inverted pendulum (Tedrake, 2018) (specifically, using the dynamics from OpenAI Gym (Brockman et al., 2016)), which
Sample Complexity of Estimating the Policy Gradient

Figure 1: The cumulative expected reward $J(\theta)$ as a function of the number of gradient steps $i \in \{1, 2, ..., 1000\}$. The black, purple, blue, green, orange, and red curves correspond to $\sigma_\zeta = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, respectively. The x-axis is the number of gradient steps taken (or equivalently, the number of rollouts), and the y-axis is $-J(\theta)$.

Our goal is to control the system over a horizon of 50 steps, from a fixed start state $s_0 = (\vartheta_0, 0)$, where $\vartheta_0 = 0.05$. For the control policy, we used a neural network $\pi_\vartheta$ with a single hidden layer with 100 neurons, ReLU activations, and linear outputs. As usual, we randomly initialize the weights; to reduce variance, we initialized the policy to have a reasonably high reward by running our model-based algorithm until $J(\theta) \geq 100$.

**Algorithms.** We use stochastic gradient descent in conjunction with each of the three estimation algorithms. On each gradient step, we use a single sample to estimate the gradient, and we take 1000 gradient steps. We modify the finite-difference algorithm to use a single random sample $\nu \sim \text{Uniform}(S^{d_\theta-1})$ (i.e., the uniform distribution on the unit sphere in $\mathbb{R}^{d_\theta}$), rather than summing over the $d_\theta$ basis vectors $\nu^{(k)}$. This choice may improve the dependence of the sample complexity on $d_\theta$; however, it should not affect dependence on $\sigma_\zeta$, which is our parameter of interest.

For the algorithm based on the policy gradient theorem, we use action noise $\xi \sim \mathcal{N}(0, \sigma_\xi I_{d_\lambda})$. For each choice of $\sigma_\zeta$, we used cross-validation to identify the optimal hyperparameters: the learning rate $\nu$ (for all algorithms), the parameter $\lambda$ (for the finite-differences algorithm), and the action noise $\sigma_\xi$ (for the algorithm based on the policy gradient theorem).

**Results.** Average the results of each algorithm over 20 runs; the algorithms have very high variance, so we discard runs that do not converge. In

Note that unlike the other two algorithms, the finite-difference algorithm actually uses 2000 sampled rollouts (since it uses two per gradient step). However, this detail does not affect our insights regarding the relative convergence rate of different algorithms for different $\sigma_\zeta$.

Our key finding is that the learning curves for the model-based and finite-differences are ordered based on the choice of $\sigma_\zeta$—i.e., the curves tend to converge more quickly for smaller choices of $\sigma_\zeta$. This effect is most apparent in the curves for the finite-differences algorithms, where curves for smaller $\sigma_\zeta$ (black and blue) converge much faster than those for larger $\sigma_\zeta$ (red and orange). In contrast, the learning curves for the policy gradient based algorithm do not have strong dependence on $\sigma_\zeta$. For example, the fastest curve to converge (at least initially) for the policy gradient based algorithm is for our second-largest choice $\sigma_\zeta = 10^{-2}$ (orange), whereas the slowest to converge is for $\sigma_\zeta = 10^{-4}$ (blue). These results mirror our theoretical insights.

Finally, as expected, the model-based algorithm converges most quickly, followed by the finite-differences and policy gradient theorem based algorithms.

7 Conclusion

We have analyzed the sample complexity of algorithms for estimating the policy gradient for nearly deterministic dynamical systems. Future work includes leveraging these results in safe reinforcement learning algorithms, and understanding the sample complexity of optimizing $J(\theta)$. 
Acknowledgements
This work was supported by NSF CCF-1910769.

References


