Mollification Effects of Policy Gradient Methods

Tao Wang [1] **Sylvia Herbert** [1] **Sicun Gao** [1]

Abstract

Policy gradient methods have enabled deep reinforcement learning (RL) to approach challenging continuous control problems, even when the underlying systems involve highly nonlinear dynamics that generate complex non-smooth optimization landscapes. We develop a rigorous framework for understanding how policy gradient methods mollify non-smooth optimization landscapes to enable effective policy search, as well as the downside of it: while making the objective function smoother and easier to optimize, the stochastic objective deviates further from the original problem. We demonstrate the equivalence between policy gradient methods and solving backward heat equations. Following the ill-posedness of backward heat equations from PDE theory, we present a fundamental challenge to the use of policy gradient under stochasticity. Moreover, we make the connection between this limitation and the uncertainty principle in harmonic analysis to understand the effects of exploration with stochastic policies in RL. We also provide experimental results to illustrate both the positive and negative aspects of mollification effects in practice.

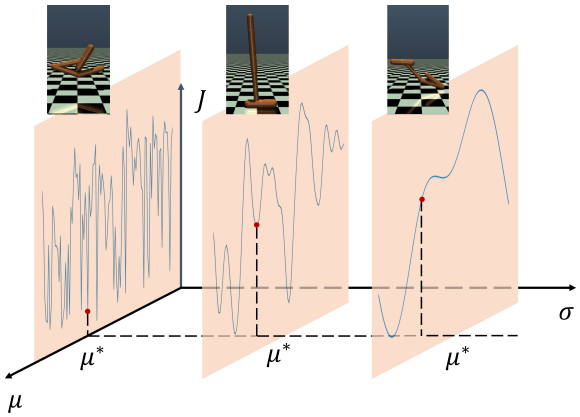


Figure 1. The Gaussian kernel in the policy gradient mollifies the optimization landscape. However, when the variance σ^2 is too small, the landscape remains highly non-smooth. Conversely, if the variance is too large, the Gaussian kernel over-smooths the landscape, eliminating the optimal solution. Both of these lead to failures in the hopper stand task. Details are avaliable in Section 6.

1. Introduction

Deep reinforcement learning (RL), especially methods based on policy gradients (Silver et al., 2014; Lillicrap et al., 2015; Schulman et al., 2017), has been successfully used to solve challenging nonlinear control problems (Gu et al., 2016; Kim et al., 2022). The approach considers control design as an optimization problem over the parameter space of the policy, and thus the effectiveness of policy gradient methods depend heavily on the *optimization landscape* in the policy space. Although the global convergence of policy gradient methods has been established for restricted classes of systems such as linear dynamics or tabular state spaces (Fazel et al., 2018; Agarwal et al., 2021), its observed effectiveness for nonlinear systems is not well understood. In fact, it has been shown that in many control settings, especially those with chaotic dynamics, the RL formulation can generate highly non-smooth optimization landscapes (Suh et al., 2022; Wang et al., 2023) that should challenge the use of gradient-based optimization methods including policy gradient. Existing attempts on bridging this gap have mostly focused on the effectiveness of exploration (Haarnoja et al., 2018a; Cai et al., 2020). However, while exploration is a common element in every search-based algorithm, it alone does not fully explain the success of RL over other schemes in high-dimensional spaces.

Given that the true gradient of the objective function may not exist in many robotics systems, we aim to understand the effectiveness of policy gradient methods through the analytic perspectives of partial differential equations (PDEs) and stochastic dynamical systems. We view the Gaussian noise introduced in the stochastic policies as a smoothing kernel that *mollifies* the objective function. Mollification is a concept in analysis that corresponds to smoothing the sharp features of a non-smooth function while remaining close to

[1]University of California, San Diego, La Jolla, USA. Correspondence to: Tao Wang <taw003@ucsd.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

the original one. In effect, regardless of whether the original landscape is smooth or not, policy gradient methods can be consistently applied to estimate the gradient of the *mollified objective*. We draw on PDE theory and establish the equivalence between policy gradient methods and the Cauchy problem for heat equations. In particular, we show that training a policy by stochastic policy gradient algorithms is equivalent to performing gradient ascent for the solution of the corresponding heat equation in the spatio-temporal domain. In control problems where deterministic control policies are expected as the outcome, the RL learning process corresponds to a time-reversed heat process. However, the backward Cauchy problem for heat equations is ill-posed in terms of the stability of the solution, which becomes less smooth as time decreases (Höllig, 1983; Kabanikhin, 2008); this suggests that reducing the variance of stochastic policy can result in a more non-smooth objective. This effect is especially pronounced when the MDP is chaotic, causing the true value function to contain significant high-frequency components.

Importantly, the results illustrate a fundamental limitation of policy gradient methods. While the mollification smooths the optimization landscape, it unavoidably changes the original objective. This trade-off can be precisely formulated by the uncertainty principle from harmonic analysis, as we visualize in Figure 1. That is, when the variance of the stochastic policy is too large, the mollified function will deviate too much from the true objective, which makes the estimated gradient no longer informative. On the other hand, if the variance is too small, the mollification effect is weak and hence leads to another highly-oscillating landscape. In either case, the training process becomes unstable, suggesting the existence of an optimal variance for the stochastic policy in policy gradient methods.

Equipped with the theoretical results, we conduct experiments to illustrate how our framework can be applied to explain both the successes and failures in practice. In particular, from the view of mollification, we can characterize a class of control problems where RL algorithms consistently face challenges: the region of attraction for the optimal policy is extremely small and thus can be entirely eliminated by the Gaussian kernel in stochastic policies. It also explains why policy gradient methods always encounter difficulties when solving quadrotor-related problems and a detailed discussion is presented in Section 6.

The contributions of this paper are summarized as follows:

• We establish a framework that builds the connection between policy gradient methods and heat equations to study their mollification effects. It provides an explanation why policy gradient can still yield effective ascent directions even when the optimization landscape is fractal.

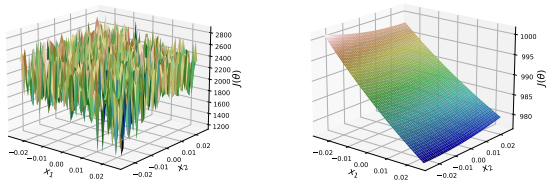
• We analyze the fundamental trade-off of policy gradient from the perspective of harmonic analysis. In particular, we demonstrate that the variance of stochastic policies should be carefully balanced. The training process loses stability if it is either too small or too large.

• Numerical results are presented to substantiate our theoretical results. We show that policy gradient methods exhibit limited effectiveness in specific MDPs, such as the quadrotor system. In these cases, the policy landscape has a spike-like structure around the optimal policy, which is then filtered by the mollification effect of policy gradient.

2. Related Work

Optimization Landscapes in RL. When the state space in an RL problem is assumed to be finite, the corresponding policy landscape is smooth (though may be non-convex). This allows for various gradient-based algorithms to converge towards the optimal policy (Agarwal et al., 2021; Bhandari & Russo, 2021; Xiao, 2022; Zeng et al., 2023). Similar results can be obtained for some simple continuous state-space MDPs, such as in linear-quadratic regulator (LQR) (Fazel et al., 2018), linear-quadratic-Gaussian control (LQG) (Tang et al., 2021) and robust control (Zhang et al., 2021). For general control settings, however, even the smoothness of objective function is not guaranteed. In particular, it has been shown that the variance of the gradient estimator in chaotic systems is very likely to explode (Parmas et al., 2018; Metz et al., 2021) due to the long chain of non-linear computations. This phenomenon is a reflection of the fractal structure in both value and policy landscapes (Wang et al., 2023). There has been work on mitigating the effect of non-smooth landscapes, such as local smoothing via Gaussian kernels that act as low-pass filters to block high-frequency components (Parmas et al., 2018; Suh et al., 2022; Zhang et al., 2023), as well as reparameterization techniques (Parmas, 2018; Parmas & Sugiyama, 2021). Our work contributes to this line by providing a theoretical analysis of the strengths and limitations of the smoothing effect produced by policy gradient methods in the case of fractal landscapes.

Policy Gradient over Non-Smooth Landscapes. It is observed that policy gradient methods can still provide good solutions when the optimization landscape is non-smooth or fractal (Lillicrap et al., 2015; Schulman et al., 2015a). A widely-accepted explanation for its effectiveness is that it encourages the policy to keep exploring undiscovered, high-reward actions (Schulman et al., 2017; Cai et al., 2020) that are out of the current policy distribution, thereby increasing the likelihood of discovering better policies. Various exploration schemes can be found in actor-critic approaches (Haarnoja et al., 2018a), Q-learning methods (Sutton &



(a) (b)

Figure 2. Fractal landscapes occur in chaotic MDPs. For instance, the objective landscape of the double pendulum system as shown in (a) has a fractal structure, in contrast to the non-chaotic single pendulum system in (b). Both systems are controlled by deterministic neural network policies.

Barto, 1998; Jin et al., 2018) and traditional control algorithms (Shkolnik et al., 2009).

Most deep RL algorithms including policy gradient can be considered as generalized policy iteration methods (Williams, 1992; Sutton et al., 1999; Kakade, 2001; Silver et al., 2014; Schulman et al., 2015a; 2017). Typically, they consist of two interacting stages: value approximation and policy improvement. Q-learning is one of the popular methods in model-free settings (Watkins & Dayan, 1992; van Hasselt et al., 2016; Hester et al., 2018), aiming to estimate the Q-function to guide future actions. Actor-critic is another framework that interactively updates the value function and current policy (Konda & Tsitsiklis, 1999; Peters & Schaal, 2008; Fujimoto et al., 2018; Haarnoja et al., 2018b). There are also works studying value approximation from both theoretical and practical sides (Schaul et al., 2015; Wang et al., 2019; Jin et al., 2020). In this paper, we will focus on the policy improvement stage and presume that the value function is exact.

Mollification in Stochastic Optimization. Policy gradient is not the only domain where stochasticity demonstrates its strength in mollifying the potentially non-smooth objective functions. Algorithms in stochastic optimization, as well as those in zeroth-order optimization, benefit from the mollification effect. (Wierstra et al., 2014; Shamir, 2015; Wang et al., 2018; Boţ & Böhm, 2020). Their ability to generalize the non-smoothness in objective functions, which is intractable for classical methods, motivates a line of applications in machine learning (Agarwal et al., 2012; Lin et al., 2022; Chen et al., 2023).

3. Preliminaries

We consider infinite-horizon MDPs with state space $\mathcal{S} \subseteq \mathbb{R}^n$ and action space $\mathcal{A} = \mathbb{R}^m$. The initial state s_0 is sampled from distribution ρ_0, and at each step $k \geq 0$, the action a_k at state s_k is obtained from the (stochastic or deterministic) policy $\pi_\theta(\cdot|s_k)$, parameterized by $\theta \in \mathbb{R}^N$. The objective of the RL problem is to maximize the performance metric:

$$J(\theta) = \mathbb{E}_{a_k \sim \pi_\theta(\cdot|s_k), s_0 \sim \rho_0}[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k)] \quad (1)$$

$$= \int_{\mathcal{S}} \rho^{\pi_\theta}(s) \int_{\mathcal{A}} Q^{\pi_\theta}(s,a)\pi_\theta(a|s)\, \mathrm{d}a\mathrm{d}s \quad (2)$$

where $\gamma \in (0,1)$ is the discount factor and $R(s,a)$ is the reward function. We will assume that the state space $\mathcal{S}$ is compact and the objective function $J(\theta)$ is bounded. In the integral form above, $\rho^\pi(\cdot)$ is the discounted visitation density under π (assuming it exists) and Q^π is the Q-function of π.

Policy gradient methods. The policy gradient theorem provides an explicit form of the gradient of the performance objective over the policy parameters that can be estimated by state-action samples (Sutton et al., 1999):

$$\nabla_\theta J(\theta) \propto \int_{\mathcal{S}} \rho^\pi(s) \int_{\mathcal{A}} Q^\pi(s,a) \nabla_\theta \pi_\theta(a|s)\, \mathrm{d}a\mathrm{d}s, \quad (3)$$

Dropping θ from π_θ in ρ^π and Q^π here is intentional, to highlight the core of the theorem in showing that the gradient operator ∇_θ only needs to be applied to the $\pi_\theta(a|s)$ part of the integrand (Sutton et al., 1999). It follows that $\nabla_\theta J(\theta)$ can be estimated by

$$\nabla_\theta J(\theta) \simeq \hat{\mathbb{E}}_{s,a\sim\pi_\theta}[A^{\pi_\theta}(s,a)\nabla_\theta \pi_\theta(a|s)] \quad (4)$$

where $A^{\pi_\theta}(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$ is the advantage function and V^{π_θ} is the value function of π_θ (Schulman et al., 2015b), and $\hat{\mathbb{E}}$ denotes sampled mean, which converges to the integral of (3) under some necessary smoothness assumptions.

Fractal landscapes in RL. One fundamental assumption for policy gradient methods to work is that the objective function $J(\theta)$ *does have a gradient*, so that (4) can provide a close estimate of it. This assumption, however, may not hold when the underlying MDP is chaotic and has a positive maximal Lyapunov exponent (MLE). Indeed, we have the following result that characterizes the fractal structure in the policy space:

Proposition 3.1. *(Wang et al., 2023) Assume that the dynamics, reward function and policy are all Lipschitz continuous with respect to their input variables. Let π_θ be a deterministic policy and $\lambda(\theta)$ denote the MLE of the system. Suppose that $\lambda(\theta) > -\log\gamma$, then*

- *$V^{\pi_\theta}(\cdot)$ is $\frac{-\log\gamma}{\lambda(\theta)}$-Hölder continuous;*
- *$J(\cdot)$ is $\frac{-\log\gamma}{\lambda(\theta)}$-Hölder continuous.*

Note that a function $f(x)$ is α-Hölder continuous at x if $|f(x)-f(x')| \leq K|x-x'|^\alpha$ for some constant $K > 0$ when $|x - x'|$ is small, which typically indicates non-smoothness when $\alpha < 1$. It reduces to Lipschitz continuity when $\alpha = 1$. In general, the true gradient $\nabla J(\theta)$ may not exist if the MLE is positive. This prompts us to understand the mechanism of (4) from the perspective of mollification, rather than focusing solely on gradient estimation.

Cauchy problem for heat equations. The heat equation, as known as the diffusion equation, describes the evolution of the distribution of temperature in the space $\mathbb{R}^m$ (Evans, 2010):

$$\begin{cases} 2u_t - \Delta u = 0, & (x,t) \in \mathbb{R}^m \times (0,\infty) \\ u = g, & (x,t) \in \mathbb{R}^m \times \{0\}. \end{cases} \tag{5}$$

where $x \in \mathbb{R}^m$ is the position and $t \in [0,\infty)$ is the time, $u(x,t)$ is the temperature at $(x,t) \in \mathbb{R}^m$, $\Delta u = u_{x_1x_1} + ... + u_{x_mx_m}$ is the Laplacian of u and g is the initial distribution of temperature at $t = 0$. The solution to this PDE system is given by

$$u(x,t) = \int_{\mathbb{R}^m} g(z)\Phi(x-z,t)\,\mathrm{d}z, \tag{6}$$

where $\Phi(x-z,t) = \frac{1}{(2\pi t)^{m/2}} e^{-\|x-z\|^2/2t}$ is the heat kernel. The solution $u(x,t)$ becomes smooth as t increases, *regardless of* how non-smooth the original function g is. This is due to the mollification effect of the heat kernel $\Phi(x-z,t)$, as shown in Figure 3 (a).

4. The Dynamics of Policy Improvement

In this section, we will establish the connection between policy gradient methods and the heat equation. We will focus on analyzing policies that are parameterized as isotropic Gaussian distributions, i.e. $\pi_\theta(a|s) = \mathcal{N}(\mu(s), \sigma^2\mathcal{I}_m)$ for fixed state $s \in \mathcal{S}$, where $\mu(s) \in \mathbb{R}^m$ and $\sigma^2\mathcal{I}_m$ is the covariance matrix with nonzero scalar $\sigma > 0$ and $\mathcal{I}_m$ being the $m \times m$ identity matrix. Consider the inner integral of (3), namely

$$L_s(\theta) := \int_{\mathcal{A}} Q^\pi(s,a)\pi_\theta(a|s)\,\mathrm{d}a, \tag{7}$$

which is a *mollification* of Q^π in the action space and hence is smooth with respect to θ even in the case of chaotic MDPs. This is because the policy density function $\pi_\theta(a|s)$ works as a smoothing kernel that mollifies the (often non-smooth) landscape of $Q^\pi(s,a)$ in $\mathcal{A}$. Since (7) is where mollification occurs, we mainly focus on $L_s(\theta)$.

To see its connection with the Cauchy problem for heat equations, let us first ignore the policy parameterization and

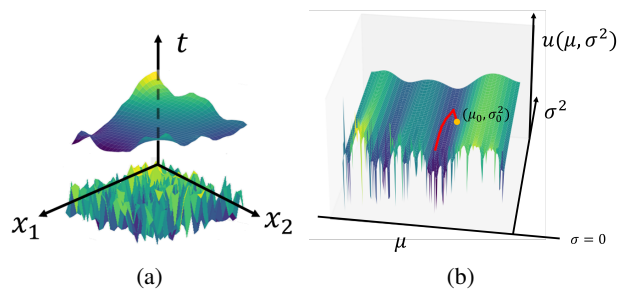


Figure 3. (a) The heat equation smooths the initial temperature distribution as t increases; (b) The gradient flow of $u(\mu,\sigma^2)$ in the solution space.

only take the mean and variance of π_θ into account, i.e., assume $L_s(\theta) = L_s(\mu,\sigma^2)$ for now, which further has

$$L_s(\mu,\sigma^2) = \int_{\mathbb{R}^m} Q^\pi(s,z)\Phi(\mu - z, \sigma^2)\,\mathrm{d}z.$$

Thus, if we consider μ and σ^2 as position and time, respectively, then $L_s(\mu,\sigma^2)$ is exactly the solution of (5). Indeed, we have the formal connection between them:

Proposition 4.1. *Let $L_s(\mu,\sigma^2)$ is given by (7) where $\pi_\theta \sim \mathcal{N}(\mu,\sigma^2\mathcal{I}_m)$, then $L_s(\mu,\sigma^2)$ is the solution of the heat equation (5).*

This connection enables us to analyze the dynamics of policy gradient methods from the view of PDE theory.

Smoothing by mollication. Essentially, the heat equation mollifies the solution through the Gaussian kernel, which acts as a low-pass filter and blocks high-frequency components. Consider the one-dimensional case in which we expand the initial condition $g(x) = \sum_{k=-\infty}^{\infty} B_k e^{ikx}$ in Fourier series (here we assume its existence for the purpose of illustration), the solution of (5) is given by

$$\tilde{u}(x,t) = \sum_{k=-\infty}^{\infty} B_k e^{-k^2 t} e^{ikx}$$

where $t \geq 0$. Therefore, for any frequency $k \in \mathbb{Z}$, the magnitude of this frequency $|B_k e^{-k^2 t}|$ decays exponentially fast as t increases, especially for higher frequencies (larger k). It means that the heat equation mollifies functions by suppressing their high-frequency components that cause fractal structures in the optimization landscape.

Mollified optimization landscape. Understanding that policy gradient methods are equivalent to performing gradient descent on the solution $u(\mu,\sigma^2)$ in (5) over the (μ,σ^2)-space allows us to better predict the behavior of policy gradient in the solution space of the heat equation (Figure 3(b)). In particular, an important property of the heat equation is the strong maximum principle:

Proposition 4.2 (Strong Maximum Principle (Evans, 2010)). *Let u be the solution of (5). Suppose that $(x_0, t_0) \in \mathbb{R}^d \times (0, \infty)$, then for any $\delta_1 > 0$ and $\delta_2 \in (0, t_0)$, exactly one of the following statements is true:*

(I) There exists $(x', t') \in \mathcal{B}(x_0, \delta_1) \times (t_0 - \delta_2, t_0)$ such that $u(x', t') > u(x, t)$;

(II) u is constant in $\overline{\mathcal{B}(x_0, \delta_1)} \times [0, t_0]$.

This results states that for any (x_0, t_0), it cannot be a strict local maximum of $u(x, t)$ when $t_0 > 0$. Following the equivalence shown in Proposition 4.1, we see that for any mean-variance pair (μ, σ^2) with some positive t, there always exists at least one increasing direction in a neighborhood of (μ, σ^2). It also explains why policy gradient algorithms can improve the performance even in the case that the true optimization landscape is fractal. The convergence modes of policy gradient follow from the connection:

Theorem 4.3. *Let π_θ be an isotropic Gaussian policy $\mathcal{N}(\mu, \sigma^2 \mathcal{I}_m)$. Suppose that $\nabla_{(\mu,\sigma^2)} L_s(\mu, \sigma^2) = 0$, then either π_θ is deterministic, or π_θ is stochastic and θ is not a strict local maximum of $L_s(\theta)$. Then exactly one of the following statements holds:*

(I) π_θ is deterministic;

(II) π_θ is stochastic and θ is not a strict local maximum of $L_s(\theta)$.

Proof. Suppose that the policy is stochastic, i.e., $\sigma > 0$. According to Proposition 4.2, for any $\delta_1, \delta_2 > 0$, $L_s(\mu, \sigma^2)$ is either constant in $\overline{\mathcal{B}(\mu_0, \delta_1)} \times [0, \sigma_0^2]$, or there exists another $[\mu', \sigma'^2] \in \mathcal{B}(\mu, \delta_1) \times (\sigma^2 - \delta_2, \sigma^2)$, which implies that π_θ cannot be a strict local maximum of $L_s(\theta)$. □

Similar results can be obtained for the gradient-descent case using the Strong Minimum Principle. In other words, if policy gradient converges toward a strict local maximum of $L_s(\mu, \sigma^2)$, we know that the final policy must be deterministic. It confirms the appropriate use of policy gradient methods in continuous control problems, where the control policy used in practice should be deterministic.

Policy parameterization. In policy gradient methods, the mean of the Gaussian policy is typically parameterized through some differentiable representation, namely $\mu = \mu(s_0; \zeta)$ where $\zeta \in \mathbb{R}^{N_0}$ denotes the parameters in the representation, such as weights and biases in a neural network. In this case, we have $\theta = [\zeta, \sigma^2]$ as the policy parameters. Note that

$$\nabla_\zeta L_s(\zeta, \sigma^2) = \nabla_\mu u(\mu(s_0; \zeta) \left.\frac{\partial \mu}{\partial \zeta}\right|_{s=s_0},$$

where $\frac{\partial \mu}{\partial \zeta}$ is the Jacobian matrix of μ with respect to the parameter ζ, and the full gradient is given as

$$\nabla_\theta L_s(\theta) = [\nabla_\mu u(\mu(\zeta), t) \left.\frac{\partial \mu}{\partial \zeta}\right|_{s=s_0}, u_{\sigma^2}(\mu(\zeta), \sigma^2))]^T. \tag{8}$$

Therefore, the degeneracy of the parameterization $\mu(s_0; \zeta)$ plays a crucial role in analyzing the gradient flow of (8): given an arbitrary initial state s_0, the Jacobian $\left.\frac{\partial \mu(s;\zeta)}{\partial \zeta}\right|_{s=s_0}$ is a linear transformation from the action space $\mathcal{A} = \mathbb{R}^m$ to the representation parameter space $\mathbb{R}^{N_0}$. This mapping is non-degenerate if it is injective, i.e., $\ker \left.\frac{\partial \mu}{\partial \zeta}\right|_{s=s_0} = \{\mathbf{0}\}$, which establishes a one-to-one correspondence between the gradient flow in Theorem 4.3 and in the new parameterization space $\mathbb{R}^{N_0}$:

Theorem 4.4. *Let $\pi_\theta(\cdot|s_0) \sim \mathcal{N}(\mu, \sigma^2 \mathcal{I}_d)$ be an isotropic Gaussian policy where $\mu = \mu(s_0; \zeta)$ is parameterized by $\zeta \in \mathbb{R}^{N_0}$. Also, assume that $\left.\frac{\partial \mu}{\partial \zeta}\right|_{s=s_0}$ is injective. Suppose that $\nabla_\theta L_{s_0}(\theta) = 0$ where $\theta = [\zeta, \sigma^2]^T$, then exactly one of the following statements holds:*

(I) π_θ is deterministic;

(II) π_θ is stochastic and θ is not a strict local maximum of $L_{s_0}(\theta)$.

This result also encourages the use of complex representations, such as neural networks, since the more parameters they contain, the more likely the Jacobian $\left.\frac{\partial \mu(x;\zeta)}{\partial \zeta}\right|_{x=s}$ is to be non-degenerate for arbitrary $s_0 \in \mathcal{S}$. For the whole gradient estimator $\nabla_\theta \hat{J}$, the component $\nabla_\zeta \hat{J}$ is given by

$$\begin{aligned}\nabla_\zeta \hat{J} &= \int_{\mathcal{S}} \rho^\pi(s) \nabla_\zeta L_s(\theta) \, \mathrm{d}s \\ &= \int_{\mathcal{S}} \rho^\pi(s) \nabla_\mu u(\mu(s; \zeta), t) \left.\frac{\partial \mu(x; \zeta)}{\partial \zeta}\right|_{x=s} \mathrm{d}s,\end{aligned}$$

where $\rho^\pi(s)$ acts as a weighting function on $\nabla_\theta L_s(\theta)$. It should note that the results in Theorem 4.4 no longer hold true, as the gradient directions at different states may cancel out. Therefore, while we can still have a gradient estimate even if the true policy objective is non-differentiable, there is no guarantee that policy gradient algorithms will converge towards some good solutions in general.

Anisotropic Gaussian distributions. Now let us consider the case of diagonal covariance matrices $\Sigma = \mathrm{diag}(r_1, ..., r_m)$ with $r_1, ..., r_m > 0$, which is a common practice in most policy gradient methods (Schulman et al., 2015a; 2017). Similar to heat equations, we will show that the strong maximum principle still holds for (9). Let $t \geq 0$ and consider the following parabolic equation

$$u_t - \frac{1}{2} \sum_{i=1}^{m} r_i \, u_{x_i x_i} = 0, \tag{9}$$

which has $u(\mu, t) = L_s(\mu, t\mathbf{r})$, where $\mathbf{r} = [r_1, ..., r_m]^T$. According to the maximum principle of parabolic equations (Evans, 2010), the same claims as in Proposition 4.2 hold for (9). Since any small neighborhood of the point $(\mu, 1)$ can always be embedded into a neighborhood of $(\mu, \mathbf{r})$, we have proved the following result:

Theorem 4.5. *Let π_θ be a Gaussian policy $\mathcal{N}(\mu, diag(\mathbf{r}))$ with diagonal covariance matrix where $\mathbf{r} = [r_1, ..., r_m]^T$ is independent of states. Suppose that $\nabla_{(\mu,\mathbf{r})} L_s(\mu, v) = 0$, then exactly one of the following statements holds:*

(I) $r_i = 0$ for some $i = 1, ..., m$;

(II) r_i are all positive and θ is not a strict local maximum of $L_s(\theta)$.

5. The Limitations of Mollification

In the previous section, we have shown that policy gradient methods are closely related to the heat equation whose solution mollifies the initial condition and hence enables gradient-based algorithms. In this section, we will move to analyze the downsides of mollification effects and describe the fundamental trade-off of policy gradient methods between smoothing and approximating.

5.1. Convergence to deterministic policies

In control tasks, the policy gradient is expected to converge towards some deterministic policy, which prompts us to ask whether there exists a smooth gradient flow in a neighborhood of $\mathbb{R}^m \times \{0\}$ in the (μ, σ^2)-space. Unfortunately, the answer is *NO* in the general case: as the heat equation makes the solution smoother and smoother as t increases, moving backwards in time will make $u(x, t)$ less smooth, as illustrated in Figure 3 (a). In other words, the gradient flow can be smoothly extended to $\mathbb{R}^m \times \{0\}$ only if $\lim_{t\to 0^+} \int_{\mathbb{R}^m} g(z) \frac{\partial \Phi(x-z,t)}{\partial t}\, dz$ exists for every $x \in \mathbb{R}^m$, which is true for smooth functions and is discussed in Appendix D. However, the solution can become highly non-smooth when approaching $t = 0$ if g is non-differentiable and even fractal. For instance, the following result shows how complex the fractal landscape can be in the one-dimensional case:

Proposition 5.1. *[Theorem 1.2, (Posey & Vaughan, 1986)] Let $\eta \in C(\mathbb{R})$ be nowhere differentiable in $[a, b]$, then the set of its local maximum (minimum) is dense in $[a, b]$.*

It suggests that in the case of fractal landscapes, there are so many local maximum points that it is challenging to determine where policy gradient converges towards. The gradient flow also loses stability as the policy approaches a deterministic limit, and even a small perturbation may result in a totally different trajectory in the policy space.

In fact, reaching the initial condition $u(x, 0)$ from a future time $t = T > 0$ involves solving the backward heat equation (Renardy & Rogers, 1993):

$$\begin{cases} 2u_t + \Delta u = 0, & (x,t) \in \mathbb{R}^m \times (-\infty, T) \\ u = g_T, & (x,t) \in \mathbb{R}^m \times \{T\}. \end{cases} \tag{10}$$

where $g_T = u(x, T)$ is the solution at time $t = T$. It is well-known in PDE theory that (10) is *ill-posed* (Kabanikhin, 2008). The physical intuition is that we cannot reverse diffusion, which is an information-losing process over time. Indeed, a PDE system is called ill-posed if it does not have a solution, or the solution is not unique, or if the solution does not change continuously with respect to the initial (terminal) conditions.

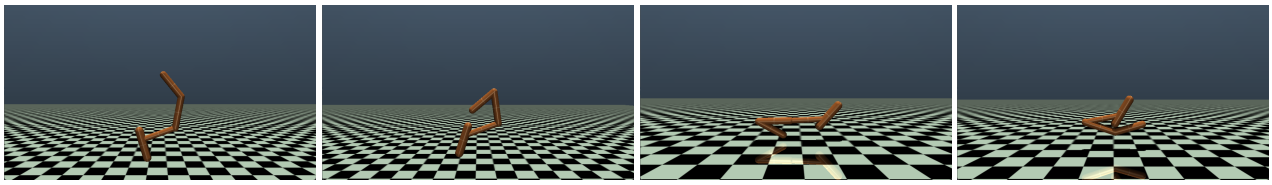
To better understand the ill-posedness of (10), the following theorem states that an arbitrarily small perturbation of the terminal condition g_T can preclude the existence of the solution:

Theorem 5.2 (Ill-posedness around deterministic policies)**.** *For any $\sigma > 0$, terminal condition $g_{\sigma^2} \in L^2(\mathbb{R}^m)$ and $\epsilon > 0$, there exists $g'_{\sigma^2} \in L^2(\mathbb{R}^m)$ with $\|g_{\sigma^2} - g'_{\sigma^2}\|_{L^2(\mathbb{R}^m)} < \epsilon$ and the solution of (10) does not exist for the terminal condition $u(\mu, \sigma^2) = g'_{\sigma^2}$.*

The complete proof can found in Appendix C. This result suggests that policy gradient methods are inherently ill-posed when reducing the variance. In this sense, even a small perturbation of the current state may significantly change the limit to which it converges.

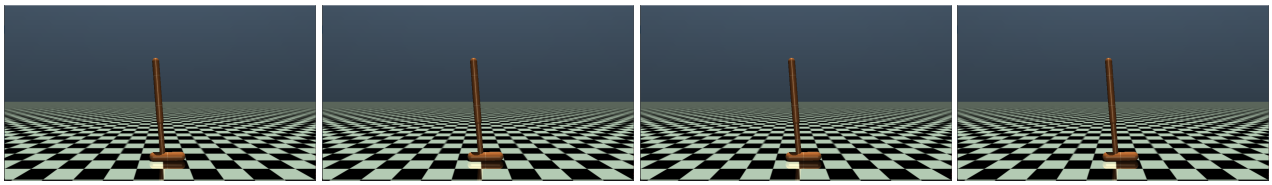
Existence of optimal variance. It has shown that as the variance σ^2 decreases, the mollified surrogate objective becomes less smooth and eventually converges to the fractal landscape of deterministic policy. On the other hand, note that the term $Q^\pi(s, a)\pi_\theta(a|s)$ is a random variable with $a \sim \pi_\theta(\cdot|s)$, the variance of $Q^\pi(s, a)\pi_\theta(a|s)$ will grows as well and the training process becomes more random as σ increases. For instance, consider the gradient-ascent algorithm $X_{k+1} = X_k + \delta G_k$ where G_k is the estimated gradient and $\delta > 0$ is the stepsize. Suppose that we just obtained X_k, the conditional variance of the next parameter is given as $Var(X_{k+1}|X_k) = \delta^2\, Var(G_k)$ (here we used $Var(X_k|X_k) = 0$), which means that the uncertainty in the gradient will propagate into the optimization parameters, and eventually affect the stability of the training curve. Combining these two facts together hints that there should exist an optimal value of σ for the policy gradient and we conclude with the following assertion:

Remark 5.3. *For chaotic MDPs where the optimization landscapes are fractal, there exists an optimal variance σ^* for the Gaussian policy that minimizes the uncertainty in training.*



(a) (b) (c) (d)

Figure 4. Hopper stand: the hopper failed to learn standing when $\sigma = 0.005$.



(a) (b) (c) (d)

Figure 5. Hopper stand: the hopper successfully learned to stand when $\sigma = 0.05$.

5.2. The uncertainty principle

The uncertainty principle, also known as the Heisenberg uncertainty principle, is a fundamental underlying law in harmonic analysis. It states that a function and its Fourier transform cannot be localized at the origin simultaneously. First, let us present the definition of Fourier transform:

Definition 5.4 (Fourier transform)**.** *Let $\phi \in \mathcal{S}(\mathbb{R}^d)$ belong to the space of rapidly decreasing functions on $\mathbb{R}^d$ that consists of all indefinitely differentiable functions f on $\mathbb{R}^d$ such that $\sup |x^\alpha (\frac{\partial}{\partial x})^\beta f(x)| < \infty$ for all multi-index α and β, then the Fourier transform of ϕ is defined as*

$$\mathcal{F}(\xi) = \int_{\mathbb{R}^d} \phi(x) e^{-2\pi i \langle x, \xi \rangle}\, dx, \quad \xi \in \mathbb{R}^d,$$

where $\langle x, \xi \rangle$ is the inner product of x and ξ.

An important property connecting Fourier transform and convolution is that $\mathcal{F}(\phi_1 * \phi_2) = \mathcal{F}(\phi_1)\mathcal{F}(\phi_2)$ for any $\phi_1, \phi_2 \in \mathcal{S}(\mathbb{R}^d)$. As mentioned in Section 4, $\hat{\phi}$ describes the frequency and thus the non-smoothness of ϕ. The rigorous formulation of the uncertainty principle is given as follows:

Proposition 5.5 (Uncertainty Principle (Stein & Shakarchi, 2003))**.** *If $\phi \in \mathcal{S}(\mathbb{R}^d)$ satisfies $\int_{\mathbb{R}^d} |\phi(x)|^2 \, \mathrm{d}x = 1$, then*

$$\left(\int_{\mathbb{R}^d} |x|^2 |\phi(x)|^2 \, \mathrm{d}x\right)\left(\int_{\mathbb{R}^d} |\xi|^2 |\hat{\phi}(\xi)|^2 \, \mathrm{d}\xi\right) \geq \frac{d^2}{16\pi^2}, \quad (11)$$

where $\hat{\phi}$ is the Fourier transform of ϕ and $\mathcal{S}(\mathbb{R}^d)$ denotes the space of rapid decreasing functions. The equality holds when ϕ is a Gaussian function.

To see how it is related to the mollification, let us consider the probability density function $\phi(x) \in \mathcal{S}(\mathbb{R}^d)$. Let $g(a) = Q^\pi(s_0, a)$ for simplicity, the convolution $g * \phi$ is close to g when ϕ concentrates at $x = 0$, i.e., the quantity $\int_{\mathbb{R}^d} |x|^2 |\phi(x)|^2 \, \mathrm{d}x$ is small. On the other hand, $g * \phi$ is smooth when $\hat{\phi}$ concentrates at $\xi = 0$, which is equivalent to having small $\int_{\mathbb{R}^d} |\xi|^2 |\hat{\phi}(\xi)|^2 \, \mathrm{d}\xi$. However, the uncertainty principle prohibits us to achieve these two things at the same time, which leads to a fundamental limitation: when the policy gradient mollifies and explores the landscape, it inevitably increases the risk of over-smoothing in the meantime. In particular, if the region of attraction of the optimal policy is small, the Gaussian kernel in policy gradient may completely eliminate that region and the problem will become unsolvable (see *Planar quadrotor balance* experiment in Section 6 for details). Therefore, policy gradient methods are inherently limited when reducing the variance to zero, fundamentally challenging their applications in control and robotics.

6. Experiments

In this part, we will apply the theory established in previous sections to explain when policy gradient methods can/cannot solve certain control problems. The controls are linearly parameterized as $u = -Ks + b$ in the planar quadrotor task, and $u = -Ks$ in the double pendulum example. The optimal solution to those problems are obtained using the LQR method in optimal control. In the hopper stand task, we use a 2-layer neural network as controller. More details of experiments are provided in Appendix F and G.

Hopper stand. We begin with a standard example in the OpenAI GYM documentation (Brockman et al., 2016). As shown in Figure 6(e), the policy landscape for a randomly

initialized policy is fractal due to the chaoticness in the underlying dynamics. To penalize any deviation from the balanced standing position, we apply a negative quadratic cost so that the total reward is always non-positive. Here we conduct 4 sets of parallel experiments with different standard deviations for the Gaussian policy, and each set is repeated across five random seeds.

In Figure 6 (a), the resulted variance of the return sequence $\{J(\theta_k)\}_{k=0}^{100}$ is even greater when $\sigma = 0.005$. This is attributed to the weak mollification effect, resulting in an optimization landscape very close to the fractal landscape generated by the deterministic policy. It shows that both $\sigma = 0.05$ and $\sigma = 0.5$ achieve nice and stable performance than the other two standard deviations, which agrees with the statement in Section 5 that the variance of stochastic policies cannot be too large or too small. The behaviors of the hopper in simulation are presented in Appendix G, and we can observe that the policy gradient with a standard deviation of $\sigma = 0.05$ discovers a policy that successfully stabilizes the hopper.

In Figure 6 (f), after 100 epochs, the trained policy entered into a smooth region which corresponds to a stable dynamics of the hopper, compared to the fractal landscape around the initial policy θ_0 in (e). It demonstrates that policy gradient is capable of guiding chaotic environments towards stability in some tasks with a proper sampling variance.

Double pendulum stabilization. It is well-known that the double pendulum system can exhibit chaotic behavior which leads to a fractal optimization landscape. In this experiment, we adopt the dynamics from (Chang et al., 2019) and the initial policy $\theta_0 = [K_0, b_0]^T$ is selected to be close to the stabilizing region, but still generates an unstable trajectory. The initial state $s_0 = [-0.2, 0.2, 0, 0]^T$ and the reward function is quadratic. As depicted in Figure 7 (a), the trained policy successfully stabilizes the system towards the origin (upright position) after 50 epochs. As in Figure 7 (b), the Q-function landscape of the trained policy θ_{50} is smooth, indicating that the system is well-behaved. This example suggests a scenario where policy gradient methods can exhibit its full strength: fine-tuning an initial policy that is close enough to the desired policies.

Planar quadrotor balance. Now let us evaluate the hardness of the balance task of the planar quadrotor system (Tedrake), and we will understand it from the view of mollification effects. Consider the control system where the two control inputs u_1, u_2 are the propelling force provided by motors. The main difficulty of balancing the quadrotor horizontally via RL algorithms is that the policy must learn to generate equal control inputs at two propellers, otherwise the quadrotor will immediately flip and fall. Therefore, even a slight deviation from the balanced control results in a significant divergence in the trajectory, which means that the landscape around the optimal solution is highly non-smooth

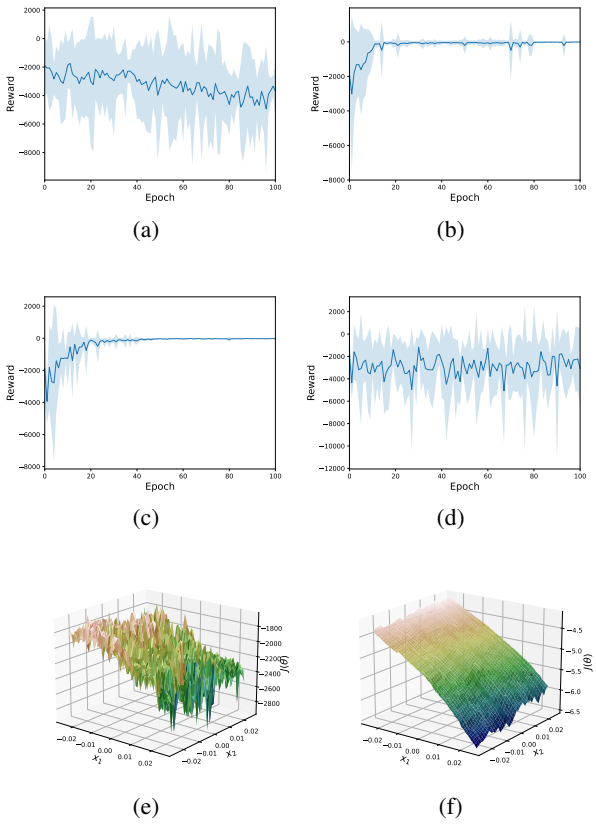


Figure 6. Hopper: Training curves with different standard deviations: (a) $\sigma = 0.005$; (b) $\sigma = 0.05$; (c) $\sigma = 0.5$; (d) $\sigma = 5$. The x-axis is the number of epochs and the y-axis is the total reward. We can see that training is successful when the variance is neither too small nor too large ($\sigma = 0.05$ and 0.5); (e) policy landscapes around the initial policy θ_0; (f) policy landscapes around the final policy θ_{100}. When generating the landscapes, $\sigma = 0.05$ is employed and we use the deterministic version of the policy for plotting.

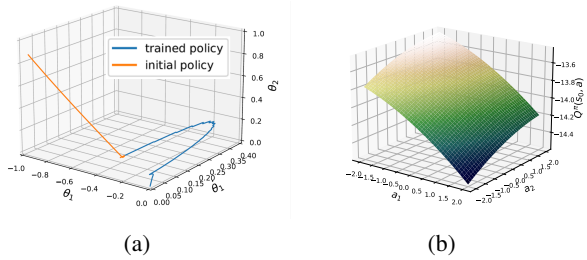
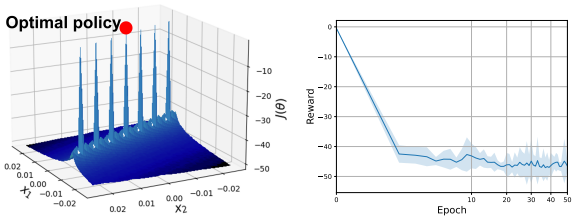


Figure 7. Double pendulum: (a) the trained policy successfully stabilizes the double pendulum system towards the origin, while the initial policy does not; (b) the Q-landscape $Q^{\pi}(s_0, a)$ of the trained policy in action space.

as shown in Figure 8 (a) despite the dynamics of quadrotor is *not chaotic*. In particular, the objective landscape has a spike at the origin (the optimal policy), which behaves like a high-frequency signal in contrast to the surrounding landscape.



(a) (b)

Figure 8. Planar quadrotor: (a) the policy landscape around the optimal policy θ^* is a spike, which is then easily eliminated by the Gaussian kernel in the policy gradient; (b) the training curve immediately falls after one epoch even the initial policy has already been optimal.

As a consequence, even if we run the policy gradient algorithm with the initial policy $\theta_0 = \theta^*$ (the optimal policy), the training curve immediately drops after one epoch and never returns to the optimal policy θ^* as shown in Figure 8 (b). This phenomenon is explained by the theoretical results in Section 5 that the optimal solution is a high-frequency noise and is filtered by the Gaussian kernel in policy gradient. Traditional control methods presume that the torque applied to the two motors must be equal, while model-free RL algorithms have to figure this out via trial and error, which leads to the aforementioned problem.

7. Concluding Remarks

A theoretical framework for understanding the mollification effect of policy gradient methods is proposed in this paper. From this perspective, we demonstrate that the mollification effect of policy gradient can be both advantageous and a bottleneck, depending on the practical cases in which it is applied and how the algorithm is implemented. Given that there are some control problems where policy gradient can almost filter the true solution as high-frequency noise, its strength should not be overestimated. Hence, a systematic understanding of the role RL should play in control and robotics needs to be established in the future.

Acknowledgements

This material is based on work supported by NSF Career CCF 2047034, NSF CCF 2112665 TILOS AI Institute, NSF CCF DASS 2217723, ONR YIP N00014-22-1-2292, and Amazon Research Award. We thank the anonymous reviewers for their valuable suggestions.

Impact Statement

This paper presents work to advance the theoretical understanding of the mechanisms and limitations of deep reinforcement learning algorithms. We believe the potential societal consequences are positive, as they will guide more principled use of machine learning techniques.

References

Agarwal, A., Negahban, S., and Wainwright, M. J. Stochastic optimization and sparse statistical recovery: An optimal algorithm for high dimensions. *arXiv preprint arXiv:1207.4421*, 2012.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Bhandari, J. and Russo, D. On the linear convergence of policy gradient methods for finite MDPs. *International Conference on Artificial Intelligence and Statistics*, 2021.

Boţ, R. I. and Böhm, A. Variable smoothing for convex optimization problems using stochastic gradients. *Journal of Scientific Computing*, 85(33), 2020.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. *Proceedings of the 37th International Conference on Machine Learning*, pp. 1283–1294, 2020.

Chang, Y., Roohi, N., and Gao, S. Neural Lyapunov control. *Advances in Neural Information Processing Systems*, 32: 3245–3254, 2019.

Chen, L., Xu, J., and Luo, L. Faster gradient-free algorithms for nonsmooth nonconvex stochastic optimization. *arXiv preprint arXiv:2301.06428*, 2023.

Evans, L. C. *Partial Differential Equations*. American Mathematical Society, 2010.

Fazel, M., Ge, R., Kakade, S. M., and Mesbahi, M. Global convergence of policy gradient methods for the linear quadratic regulator. *Proceedings of the 35th International Conference on Machine Learning*, pp. 1467–1476, 2018.

Fujimoto, S., van Hoof, H., and Meger, D. Addressing function approximation error in actor-critic methods. *Proceedings of the 35th International Conference on Machine Learning*, pp. 1587–1596, 2018.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. *arXiv preprint arXiv:1610.00633*, 2016.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, 2018a.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b.

Hester, T., Vecerik, M., Pietquin, O., Lanctot, M., Schaul, T., Piot, B., Horgan, D., Quan, J., Sendonaris, A., Osband, I., Dulac-Arnold, G., Agapiou, J., Leibo, J., and Gruslys, A. Deep Q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

Höllig, K. Existence of infinitely many solutions for a forward backward heat equation. *Transactions of the American Mathematical Society*, 278(1), 1983.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? *Advances in Neural Information Processing Systems*, 31, 2018.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. *Proceedings of Thirty Third Conference on Learning Theory*, pp. 2137–2143, 2020.

Kabanikhin, S. I. Definitions and examples of inverse and ill-posed problems. *J. Inv. Ill-Posed Problems*, 16:317–357, 2008.

Kakade, S. M. A natural policy gradient. *Advances in Neural Information Processing Systems 14*, pp. 1531–1538, 2001.

Kim, S., Sorokin, M., Lee, J., and Ha, S. Human Motion Control of Quadrupedal Robots using Deep Reinforcement Learning. *Proceedings of Robotics: Science and Systems*, 2022.

Konda, V. and Tsitsiklis, J. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12, 1999.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Lin, T., Zheng, Z., and Jordan, M. I. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *arXiv preprint arXiv:2209.05045*, 2022.

Metz, L., Freeman, C. D., Schoenholz, S. S., and Kachman, T. Gradients are not all you need. 2021.

Parmas, P. Total stochastic gradient algorithms and applications in reinforcement learning. *NeurIPS*, 2018.

Parmas, P. and Sugiyama, M. A unified view of likelihood ratio and reparameterization gradients. *International Conference on Artificial Intelligence and Statistics*, pp. 4078–4086, 2021.

Parmas, P., Rasmussen, C. E., Peters, J., and Doya, K. Pipps: Flexible model-based policy search robust to the curse of chaos. *Proceedings of the 35th International Conference on Machine Learning*, pp. 4065–4074, 2018.

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

Posey, E. E. and Vaughan, J. E. Extrema and nowhere differentiable functions. *The Rocky Mountain Journal of Mathematics*, 16(4):661–668, 1986.

Renardy, M. and Rogers, R. C. *An introduction to partial differential equations*. Springer, 1993.

Rudin, W. *Functional Analysis*. McGraw-Hill, Inc., 1991.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1312–1320, 2015.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1889–1897, 2015a.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shamir, O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *arXiv preprint arXiv:1507.08752*, 2015.

Shkolnik, A., Walter, M., and Tedrake, R. Reachability-guided sampling for planning under differential constraints. *IEEE International Conference on Robotics and Automation*, pp. 2859–2865, 2009.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. Deterministic policy gradient algorithms. *Proceedings of the 31st International Conference on Machine Learning*, pp. 387–395, 2014.

Stein, E. M. and Shakarchi, R. *Fourier Analysis: An Introduction*. Princeton University Press, 2003.

Suh, H., Simchowitz, M., Zhang, K., and Tedrake, R. Do differentiable simulators give better policy gradients? *Proceedings of the 39th International Conference on Machine Learning*, 162:20668–20696, 2022.

Sutton, R. S. and Barto, A. *Reinforcement Learning: an Introduction*. MIT Press, 1998.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems 12*, pp. 1057–1063, 1999.

Tang, Y., Zheng, Y., and Li, N. Analysis of the optimization landscape of linear quadratic gaussian (LQG) control. *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, pp. 599–610, 2021.

Tedrake, R. *Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832)*.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double Q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), 2016.

Wang, T., Herbert, S., and Gao, S. Fractal landscapes in policy optimization. *arXiv preprint arXiv:2310.15418*, 2023.

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. Stochastic zeroth-order optimization in high dimensions. pp. 1356–1365, 2018.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Watkins, C. J. C. H. and Dayan, P. Q-learning. *Machine Learning*, 8:279–292, 1992.

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., and Schmidhuber, J. Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980, 2014.

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

Xiao, L. On the convergence rates of policy gradient methods. *arXiv preprint arXiv:2201.07443*, 2022.

Zeng, S., Doan, T. T., and Romberg, J. Connected superlevel set in (deep) reinforcement learning and its application to minimax theorems. *arXiv preprint arXiv:2303.12981*, 2023.

Zhang, K., Hu, B., and Başar, T. Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit regularization and global convergence. *SIAM Journal on Control and Optimization*, 59(6):4081–4109, 2021.

Zhang, S., Jin, W., and Wang, Z. Adaptive barrier smoothing for first-order policy gradient with contact dynamics. *Proceedings of Machine Learning Research*, pp. 41219–41243, 2023.

A. Convolution

We introduce some fundamental concepts in Fourier analysis. Let $L^1(\mathbb{R}^N) = \{f : \mathbb{R}^N \to \mathbb{R}^N : \int_D |f| < \infty\}$ denote the space of integrable functions, and let $L^\infty(\mathbb{R}^N) = \{f : \mathbb{R}^N \to \mathbb{R}^N : |f(x)| \leq M \textit{ a.e. } x \in \mathbb{R}^N \textit{ for some } M > 0\}$ denote the space of essentially bounded measurable functions.

Notation. To avoid any possible ambiguity, we collect the notations below for later reference. Unless mentioned otherwise, let $U \subset \mathbb{R}^N$ be an open set, and we have:

- $C_c^\infty(U)$: The space of infinitely differentiable functions $\phi : U \to \mathbb{R}$ with compact support in U (i.e., the set $\{\theta \in \mathbb{R}^N : \phi(\theta) \neq 0\}$ is a compact subset of U).
- $L^1_{loc}(U)$: The space of locally integrable functions on U ($v \in L^1_{loc}(U)$ means that for any compact set $K \subset U$, ϕ is integrable on K).
- $\mathcal{B}(x, r)$: The open ball of radius $r > 0$ centered at x.
- $\mathcal{S}(\mathbb{R}^d)$: The space of rapid decreasing functions that consists of all indefinitely differentiable functions f on $\mathbb{R}^d$ such that $\sup |x^\alpha (\frac{\partial}{\partial x})^\beta f(x)| < \infty$ for all multi-index α and β.

The convolution operator is defined as:

Definition A.1 (Convolution)**.** *Suppose that $\phi_1 \in L^1(\mathbb{R}^\mathbb{N})$ and $\phi_2 \in L^\infty(\mathbb{R}^\mathbb{N})$, then their convolution $\phi_1 * \phi_2 \in L^\infty(\mathbb{R}^\mathbb{N})$ is defined by*

$$(\phi_1 * \phi_2)(x) = \int_{\mathbb{R}^\mathbb{N}} \phi_1(y)\phi_2(x - y)\, dy.$$

An important property of convolution is that for any variable x_i, the partial derivative satisfies

$$\frac{\partial(\phi_1 * \phi_2)}{\partial x_i} = \frac{\partial \phi_1}{\partial x_i} * \phi_2 = \phi_1 * \frac{\partial \phi_2}{\partial x_i}$$

which implies that $\phi_1 * \phi_2$ is smooth as long as one of ϕ_1 and ϕ_2 is smooth. In particular, we have $\nabla(\phi_1 * \phi_2) = \phi_1 * \nabla\phi_2$ when ϕ_2 is smooth.

B. The Initial State Distribution ρ_0

Here we briefly discuss why the distribution of initial states does not affect the smoothness of the original objective too much. To demonstrate this, let us rewrite the value function $V^{\pi_\theta}(s)$ as $V(s; \theta)$ to emphasize its dependence on θ. Then, the objective function is given by

$$J(\theta) = \int V(s; \theta)\rho_0(s)\, \mathrm{d}s.$$

Thus, if $J(\theta)$ is differentiable, we will be able to push the differentiation inside the integral as ρ_0 is independent of θ, i.e.,

$$\frac{\mathrm{d}J}{\mathrm{d}\theta} = \int \frac{\partial V(s; \theta)}{\partial \theta} \rho_0(s)\, \mathrm{d}s. \tag{12}$$

Therefore, the smoothness of $J(\theta)$ is closely associated with the differentiability of $V(s, \theta)$ *regardless of* the distribution ρ_0. On the other hand, if the value function $V(s; \theta)$ is not differentiable in θ-space for all $s \in S$ where $S \subset \mathbb{R}^n$ is a set of positive measure, $\frac{\mathrm{d}J}{\mathrm{d}\theta}$ no longer exists as the integral on the right-hand side of (12) diverges.

C. Proof of Theorem 5.2

Proof. Without loss of generality, we assume $m = 1$. Since heat equations are linear, it suffice to prove the case of $g_T \equiv 0$ where $T = \sigma^2$.

Note that for any terminal condition $\phi \in L^2(\mathbb{R})$, the solution of (10) is given by

$$u'(x,t) = \frac{1}{\sqrt{2\pi(T-t)}} \int_{\mathbb{R}} \phi(z) e^{\frac{|x-z|^2}{2(T-t)}} \, \mathrm{d}z.$$

Now let $g'_T(x) = \frac{\epsilon}{2} e^{-\frac{|x|}{2}}$ so that $\|g'_T\| = \frac{\epsilon}{\sqrt{2}} < \epsilon$, and for any $t < T$ and $x \in \mathbb{R}$, we have

$$\begin{aligned}
\frac{1}{\sqrt{2\pi(T-t)}} \int_{\mathbb{R}} g'_T(z) e^{\frac{|x-z|^2}{2(T-t)}} \, \mathrm{d}z &= \frac{\epsilon}{2} \frac{1}{\sqrt{2\pi(T-t)}} \int_{\mathbb{R}} e^{\frac{|x-z|^2}{2(T-t)} - \frac{|z|}{2}} \, \mathrm{d}z \\
&\geq \frac{\epsilon}{2} \frac{1}{\sqrt{2\pi(T-t)}} \int_0^{\infty} e^{\frac{(z-x)^2}{2(T-t)} - \frac{z}{2}} \, \mathrm{d}z \\
&\geq \frac{\epsilon}{2} \frac{1}{\sqrt{2\pi(T-t)}} \int_0^{\infty} e^{\frac{x^2 - (x + \frac{T-t}{2})^2}{2(T-t)}} \, \mathrm{d}z \\
&= \infty,
\end{aligned}$$

which implies that the solution $u'(x,t)$ does not exist and we complete the proof. □

D. Lipschitz Continuous Objectives

In Section 5, we have seen that the existence of $\lim_{t \to 0} u(x,t)$ depends on the smoothness of initial condition g. However, for many problems such as finite state-space MDPs in RL, the objective function is locally Lipschitz continuous. In this section, we will first introduce the notion of weak derivatives from the distribution theory (Rudin, 1991), then prove the gradient estimated by policy gradient methods converges to the weak derivative of objective function. Consider the mollified objective function

$$F(\mu, \sigma) = f * p_\sigma(\mu). \tag{13}$$

where $p_\sigma(\mu)$ is the Gaussian distribution with mean μ and covariance $\sigma^2 \mathcal{I}$. The definition of weak derivatives that extends the notion of conventional derivatives:

Definition D.1. *(Weak derivative) Assume $U \subset \mathbb{R}^N$ is open. Suppose that $u, v \in L^1_{loc}(U)$ and $\beta \in \Lambda(N)$. We say that v is the βth-weak partial derivative of u, written as $D^\beta u = v$, if*

$$\int_U u D^\beta \phi \, dx = (-1)^{|\beta|} \int_U v \phi \, dx$$

for all $\phi \in C_c^\infty(U)$, where $D^\beta \phi$ gives the corresponding β^{th}-conventional partial derivative of ϕ and $\Lambda(N) = \mathbb{Z}^N$ is the set of multi-indices of dimension N, that is, every $\beta = (a_1, ..., a_N) \in \Lambda(N)$, $|\beta| = a_1 + ... + a_N$.

The following example gives an idea of how weak derivatives are related to strong derivatives:

Example D.2. *Consider the function $u(x) = |x|$ where $x \in U = (-1, 1)$, and define*

$$v(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0. \end{cases}$$

Now let us show $Du = v$ using the preceding definition, that is, for any $\phi \in C_c^\infty(U)$, we need to prove

$$\int_{-1}^{1} u \phi' \, dx = -\int_{-1}^{1} v \phi \, dx = \int_{-1}^{0} \phi \, dx - \int_0^1 \phi \, dx.$$

Applying the integration by parts, it yields

$$\begin{aligned}\int_{-1}^{1} u\phi'\,dx &= -\int_{-1}^{0} x\phi'\,dx + \int_{0}^{1} x\phi'\,dx \\ &= \phi(1) - \int_{0}^{1}\phi\,dx - \phi(-1) + \int_{-1}^{0}\phi\,dx \\ &= \int_{-1}^{0}\phi\,dx - \int_{0}^{1}\phi\,dx \\ &= -\int_{-1}^{1} v\phi\,dx\end{aligned}$$

where we use the fact that $\phi \in C_c^\infty(U)$ (thus $\phi(-1) = \phi(1) = 0$) at the third equality.

Then we will see that the gradient estimated by policy gradient methods converges to the weak gradient of the true objective function as $\sigma \to 0$ when $f(\theta)$ is globally Lipschitz continuous:

Theorem D.3. *Suppose that f is globally Lipschitz in $U \subset \mathbb{R}^N$ and uniformly bounded over $\mathbb{R}^N$ where $U \subset \mathbb{R}^N$ is a bounded open set. Let $D^{\alpha_i}u$ denote the first-order derivative with respect to x_i, then*

- *$f * p_\sigma \to f$, a.e.;*
- *$\frac{\partial(f*p_\sigma)}{\partial x_i} \to D^{\alpha_i} f$, a.e.;*

as $\sigma \to 0$.

Indeed, it explains why policy gradient methods work sufficiently well in finite-space problems (Agarwal et al., 2021). On the other hand, if the objective function is locally Lipschitz continuous but not uniformly bounded, the convergence is not guaranteed as the function diverges fast as $\|\theta\| \to \infty$ as shown in the following example:

Example D.4. *Consider the function $f(x) = e^{x^3}$ and $p_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$ is a Gaussian kernel. where $x \in \mathbb{R}$ and $\sigma > 0$. Apparently, f is locally Lipschitz continuous. However, the integral*

$$\int_{-\infty}^{\infty} f(x)\,p_\sigma(y-x)\,dx = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty} 3x^2 e^{(x^3 - \frac{(y-x))^2}{2\sigma^2})}\,dx = \infty.$$

Thus, $f * p_\sigma$ does not exist for all $\sigma > 0$.

It indicates a fundamental drawback of Gaussian kernel: even though the objective function f is smooth in a neighborhood of every policy parameter θ, the mollification $f * p_\sigma$ may blow up if f diverges exponentially fast at infinity. Therefore, a local gradient estimator may be a better choice in this case:

Definition D.5. *(Bump function) For any $\sigma > 0$, let $\eta_\sigma \in C^\infty(\mathbb{R}^N)$ be defined by*

$$\eta_\sigma(x) = \begin{cases} C\exp(\frac{1}{|x|^2-\sigma^2}), & |x| \le \sigma \\ 0, & |x| > \sigma. \end{cases} \tag{14}$$

where the constant $C > 0$ is selected so that $\int_{\mathbb{R}^N} \eta_\sigma dx = 1$.

Unlike Gaussian kernel which uses information from the entire space, bump function samples data from a small neighborhood of θ_0. However, as we mentioned in previous sections, Gaussian kernels suffer from two problems: first, the computational budget may not allow one to sample trajectories as many as they want, which leads to a significant sample bias when the variance is large; second, as a non-local estimator, Gaussian kernel samples data from the entire space so that the region of high loss/frequency may affect the estimate of local behavior. Therefore, using bump functions to estimate search directions may be a better option in such cases. The convergence result for locally Lipschitz continuous functions is established as follows:

Theorem D.6. *(Evans, 2010) Suppose that f is locally Lipschitz in some open set $U \subset \mathbb{R}^N$. Let $D^{\alpha_i} u$ denote the first-order derivative with respect to x_i, then*

- *$f * \eta_\sigma \to f$ almost everywhere;*
- *$\|D^{\alpha_i} f - \frac{\partial (f*\eta_\sigma)}{\partial x_i}\|_{L^2(U)} \to 0$;*

as $\sigma \to 0$.

E. Proofs Omitted in Section D

The following lemma will be helpful:

Lemma E.1. *(Lebesgue Differentiation Theorem, (Stein & Shakarchi, 2003)) Let $f : \mathbb{R}^N \to \mathbb{R}$ is locally integrable, then for a.e. $x_0 \in \mathbb{R}^N$, it has*

- *$\frac{1}{Vol(\mathcal{B}(x_0,r))} \int_{\mathcal{B}(x_0,r)} f \, dx \to f(x_0)$ as $r \to 0$;*
- *$\frac{1}{Vol(\mathcal{B}(x_0,r))} \int_{\mathcal{B}(x_0,r)} |f(x) - f(x_0)| \, dx \to 0$ as $r \to 0$;*

where $\mathcal{B}(x_0, r)$ is the open ball of radius r centered at x_0 and $Vol(\mathcal{B}(x_0, r))$ is the volume of the ball.

Specifically, a point x_0 at which $\frac{1}{Vol(\mathcal{B}(x_0,r))} \int_{\mathcal{B}(x_0,r)} |f(x) - f(x_0)| \, dx \to 0$ as $r \to 0$ is called a *Lebesgue point* of f.

Proof of Theorem D.3: (a) Suppose that $\theta_0 \in \mathbb{R}^N$ is a Lebesgue point of f. Let $|f(\theta)| \leq M$ for all $\theta \in \mathbb{R}^N$, then for any $\epsilon > 0$, there exists $K > 0$ such that

$$\int_{|\theta-\theta_0|\geq r} |f(\theta) - f(\theta_0)| \, p_\sigma(\theta_0 - \theta) \, d\theta \leq 2M \int_{|\theta-\theta_0|\geq r} p_\sigma(\theta_0 - \theta) \, d\theta < \frac{\epsilon}{2} \tag{15}$$

for all $r \geq K\sigma$ and $\sigma > 0$, which is possible because p_σ is a Gaussian density function. According to Lemma E.1, there exists $r_0 > 0$ such that for any $r < r_0$,

$$\frac{1}{Vol(\mathcal{B}(\theta_0, r))} \int_{|\theta-\theta_0|<r} |f(\theta) - f(\theta_0)| \, d\theta < \frac{\epsilon}{2} \frac{(2\pi)^{\frac{N}{2}}}{C(K+1)^N}$$

where $C = \frac{Vol(\mathcal{B}(x_0,r))}{r^N}$ depends only on the dimension N.

On the other hand, the integral over $\mathcal{B}(x_0, r)$ is bounded by

$$\begin{aligned}
\int_{|\theta-\theta_0|<r} |f(\theta) - f(\theta_0)| \, p_\sigma(\theta_0 - \theta) \, d\theta &= \frac{1}{(2\pi)^{\frac{N}{2}} \sigma^N} \int_{|\theta-\theta_0|<r} |f(\theta) - f(\theta_0)| \, e^{-\frac{|\theta-\theta_0|^2}{2\sigma^2 N}} \, d\theta \\
&\leq \frac{1}{(2\pi)^{\frac{N}{2}} \sigma^N} \int_{|\theta-\theta_0|<r} |f(\theta) - f(\theta_0)| \, d\theta \\
&\leq \frac{1}{(2\pi)^{\frac{N}{2}} \sigma^N} \frac{\epsilon}{2} \frac{(2\pi)^{\frac{N}{2}}}{(K+1)^N} r^N \\
&\leq \frac{\epsilon}{2} (\frac{r}{(K+1)\sigma})^N
\end{aligned}$$

In particular, plugging $r = (K+1)\sigma$ into the last inequality yields

$$\int_{|\theta-\theta_0|<r} |f(\theta) - f(\theta_0)| \, p_\sigma(\theta_0 - \theta) \, d\theta < \frac{\epsilon}{2} \tag{16}$$

when $\sigma < \frac{r_0}{K+1}$. Combining (15) and (16) yields

$$\int_{\mathbb{R}^N} |f(\theta) - f(\theta_0)| \, p_\sigma(\theta_0 - \theta) \, d\theta < \epsilon.$$

Since $|f * p_\sigma(\theta_0) - f(\theta_0)| \leq \int_{\mathbb{R}^N} |f(\theta) - f(\theta_0)| \, p_\sigma(\theta_0 - \theta) \, d\theta$, using the fact that Lebesgue points are almost everywhere in $\mathbb{R}^N$ completes the proof.

(b) According to the Rademacher's Theorem, f is differentiable almost everywhere in U where $U \subset \mathbb{R}^N$ is an open set. For each x_i, applying the definition of weak derivative yields

$$D^{\alpha_i}(f * p_\sigma)(\theta) = (D^{\alpha_i} f) * p_\sigma(\theta) \tag{17}$$

for a.e. $\theta \in U$ since weak derivatives coincide with strong derivatives when the function is smooth. Specifically, the right-hand side of (17) exists because f is globally Lipschitz continuous so that $D^{\alpha_i} f$ is bounded almost everywhere (and hence $D^{\alpha_i} f \in L^\infty(\mathbb{R}^N)$), which implies that the integral $\int_{\mathbb{R}^N} (D^{\alpha_i} f)(y - x) \, p_\sigma(x) \, dy$ exists.

Applying the result of part (a) completes the proof.

	Quadrotor	Double pendulum	Hopper
Batch	16	16	32
Epoch	50	50	100
Horizon	1000	1000	1000
Discount factor	0.99	0.99	0.99

Table 1. Hyperparameters for experiments.

Coordinate	Observation
s_2	Angle of the torso
s_3	Angle of the thigh joint
s_4	Angle of the leg joint
s_5	Angle of the foot joint
s_6	Horizontal velocity the torso
s_7	Vertical velocity the torso
s_8	Angular velocity of the angle of the torso
s_9	Angular velocity of the angle of the thigh hinge
s_{10}	Angular velocity of the angle of the leg hinge
s_{11}	Angular velocity of the angle of the foot hinge

Table 2. What the coordinates correspond to.

F. Details of experiments

The hyperparameters used in Section 6 is summarized in Table 1. Note that all gradients are normalized.

F.1. Hopper stand

Regarding the controller, we use a 2-layer neural network $u = W_2 \tanh W_1 s$ where the width of the hidden layer is 16. the activation function is *tanh*. The reward function $R(s,a) = -(s_2^2+s_3^2+s_4^2+s_5^2+0.1\,(s_6^2+s_7^2+s_8^2+s_9^2+s_{10}^2+s_{11}^2))-0.001\,|a|^2$ where the coordinates are specified as in Table 2. The stepsize for policy update in each epoch is $\delta = 1$.

F.2. Double pendulum

The dynamics of double pendulum (Figure 9 (b)) is solved from the manipulation equation

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} = \tau(q) + Bu$$

where

$$M(q) = \begin{bmatrix} I_1 + I_2 + m_2 l_1^2 + 2m_2 l_1 l_{c2}\cos(\theta_2) & I_2 + m_2 l_1 l_{c2}\cos(\theta_2) \\ I_2 + m_2 l_1 l_{c2}\cos(\theta_2) & I_2 \end{bmatrix}, \quad q = \begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix}, \quad B = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix},$$

$$C(q,\dot{q}) = \begin{bmatrix} -2m_2 l_1 l_{c2}\sin(\theta_2)\dot{\theta}_2 & -m_2 l_1 l_{c2}\sin(\theta_2)\dot{\theta}_2 \\ m_2 l_1 l_{c2}\sin(\theta_2)\dot{\theta}_1 & 0 \end{bmatrix},$$

$$\tau(q) = \begin{bmatrix} -m_1 g l_{c1}\sin(\theta_1) - m_2 g(l_1\sin(\theta_1) + l_{c2}\sin(\theta_1+\theta_2)) \\ -m_2 g l_{c2}\sin(\theta_1+\theta_2) \end{bmatrix}, \quad u = \begin{bmatrix}u_1 \\ u_2\end{bmatrix}$$

with $I_1 = I_2 = 0.1, m = 0.15, g = 9.81, l_1 = l_2 = 0.5, l_{c1} = l_{c2} = 0.25$. The controller $u = -Ks$ is linear and the initial feedback gain matrix is

$$K_0 = \begin{bmatrix} -20 & -20.0854 & -21.4826 & -10.0516 \\ -18.22 & -19.143 & -9.2905 & -6.6695 \end{bmatrix}.$$

The reward function is $R(s,a) = 5(\theta_1^2+\theta_2^2)+0.5(\dot{\theta}_1^2+\dot{\theta}_2^2)+0.00005|a|^2$. The stepsizes for simulation and policy updates are $\Delta t = 0.01$ and $\delta = 1$. The control inputs are saturated, i.e., $u = [\max\{-10, \min\{u_1, 10\}\}, \max\{-10, \min\{u_2, 10\}\}]^T$.

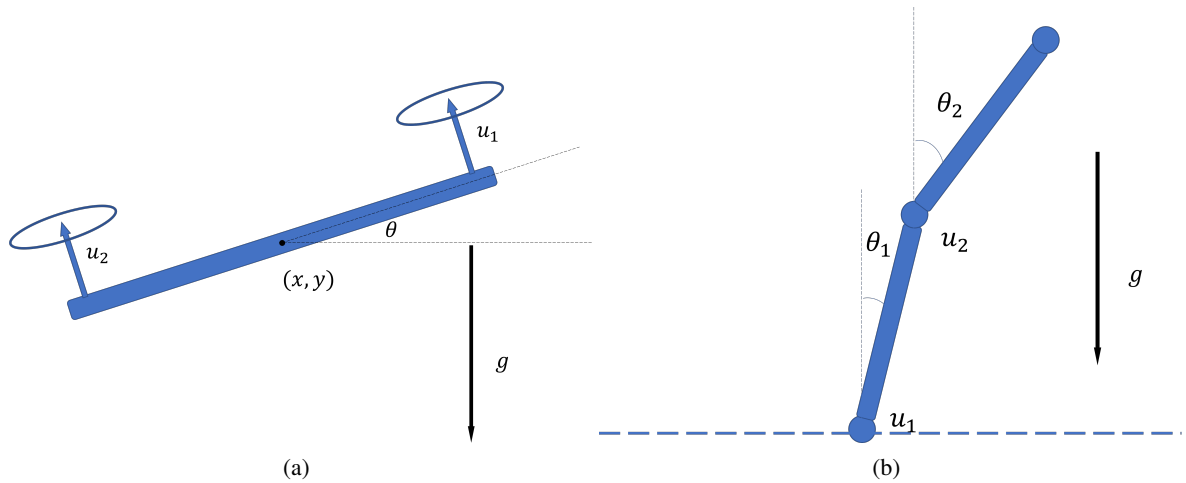


Figure 9. (a) Planar quadrotor; (b) double pendulum.

F.3. Planar quadrotor

The dynamics of the planar quadrotor (Figure 9 (a)) is

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{1}{m}(u_1 + u_2)\sin\theta \\
\dot{y}_1 &= x_2 \\
\dot{y}_2 &= \frac{1}{m}(u_1 + u_2)\cos\theta - mg \\
\dot{\theta} &= w \\
\dot{w} &= \frac{r}{I}(u_1 - u_2)
\end{aligned}$$

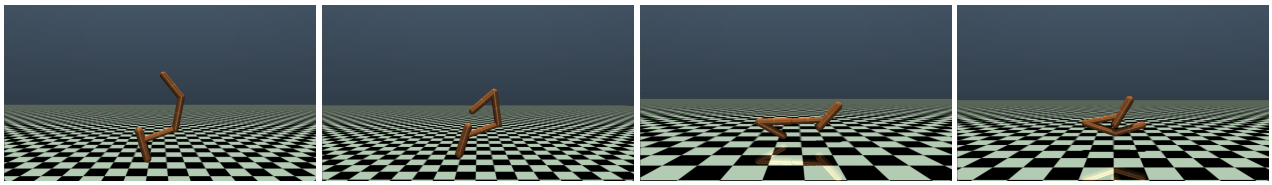
where $m = 1.0, I = 0.1, r = 0.5, g = 9.81$ and $u_1, u_2 \in \mathbb{R}$ are the control inputs. The reward function is $R(s, a) = -(x_1^2 + y_1^2 + \theta^2 + 0.1\,(x_2^2 + y_2^2 + w^2) + 0.0001\,|a|^2)$ and $\Delta t = 0.1$ is the stepsize of discretization. The controller $u = -Ks + b$ is linear and the initial policy is $u_0 = -K_0 s + b_0$ with

$$K_0 = \begin{bmatrix} -2.2361 & -3.3404 & 2.2361 & 2.69 & 13.5092 & 2.7752 \\ 2.2361 & 3.3404 & 2.2361 & 2.69 & -13.5092 & -2.7752 \end{bmatrix}, \quad b_0 = \begin{bmatrix} 4.905 \\ 4.905 \end{bmatrix}.$$

u_0 is the optimal policy solved by LQR methods. The standard deviation of Gaussian policy is fixed to $\sigma = 0.1$ and the stepsize for policy updates in each epoch is $\delta = 0.001$. The control inputs are saturated, i.e., $u = [\max\{0, \min\{u_1, 10\}\}, \max\{0, \min\{u_2, 10\}\}]^T$

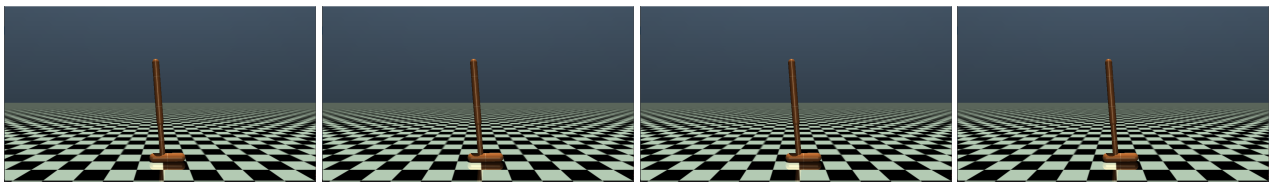
G. Behaviors of Hopper

The behaviors of the hopper with different variances are presented below.



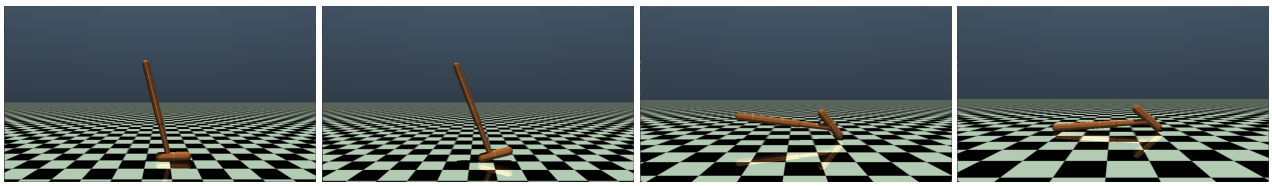
(a) (b) (c) (d)

Figure 10. $\sigma = 0.005$.



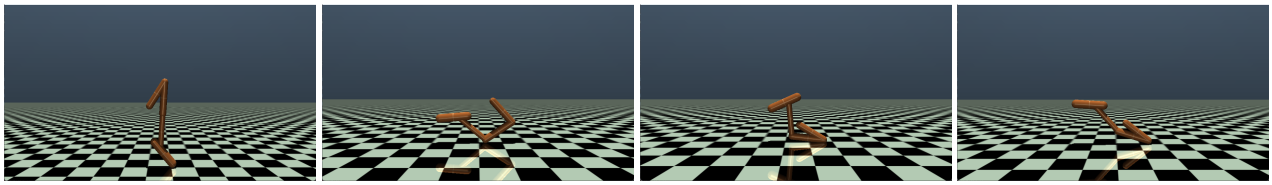
(a) (b) (c) (d)

Figure 11. $\sigma = 0.05$.



(a) (b) (c) (d)

Figure 12. $\sigma = 0.5$.



(a) (b) (c) (d)

Figure 13. $\sigma = 5$.