Policy Optimization with Stochastic Mirror Descent

Long Yang [1,*], Yu Zhang [2,*], Gang Zheng[1], Qian Zheng[1,3], Pengfei Li[1], Jianghang Huang [1], Gang Pan [1,†]

[1]College of Computer Science and Technology, Zhejiang University, China
[2]Netease Games AI Lab, Hangzhou, China
[3]School of Electrical and Electronic Engineering, Nanyang Technological University,Singapore
[1] {yanglong,gang_zheng, hzzhangyu,pfl,gpan}@zju.edu.cn [2]zhangyu15@corp.netease.com [3]zhengqian@ntu.edu.sg

Abstract

Improving sample efficiency has been a longstanding goal in reinforcement learning. This paper proposes VRMPO algorithm: a sample efficient policy gradient method with stochastic mirror descent. In VRMPO, a novel variance-reduced policy gradient estimator is presented to improve sample efficiency. We prove that the proposed VRMPO needs only $\mathcal{O}(\epsilon^{-3})$ sample trajectories to achieve an ϵ-approximate first-order stationary point, which matches the best sample complexity for policy optimization. Extensive empirical results demonstrate that VRMPO outperforms the state-of-the-art policy gradient methods in various settings.

Introduction

Policy gradient (Williams 1992; Sutton et al. 2000) is widely used to search the optimal policy in reinforcement learning (RL), and it has achieved significant successes in challenging fields such as playing Go (Silver et al. 2016, 2017) or robotics (Duan et al. 2016). However, policy gradient methods suffer from high sample complexity, since many existing popular methods require to collect a lot of samples for each step to update its parameters (Haarnoja et al. 2018; Yang et al. 2021; Xing et al. 2021; Yang et al. 2022), which partially reduces the effectiveness of the samples. Besides, it is still very challenging to provide a theoretical analysis of sample complexity for policy gradient methods instead of empirically improving sample efficiency.

To improve sample efficiency, this paper addresses how to design an efficient and convergent algorithm with stochastic mirror descent (SMD) (Nemirovskij and Yudin 1983). SMD keeps the advantage of low memory requirement and low computational complexity (Lei and Tang 2018), which implies SMD needs less samples to learn a model. However, the significant challenges of applying the existing SMD to RL are two-fold: **1)** The objective of policy-based RL is a typical non-convex function, Ghadimi et al. (2016) show that it may cause instability and even divergence when updating the parameter of a non-convex objective by SMD via a single sample. **2)** The large variance of policy gradient estimator is a critical bottleneck of improving sample efficiency for policy optimization with SMD. The non-stationary sampling process with the environment will lead to a large variance on the policy gradient estimator (Papini et al. 2018), which requires more samples to get a robust policy gradient and results in poor sample efficiency (Liu et al. 2018).

To address the above challenges, we provide a theory analysis of the dilemma of applying SMD to policy optimization. Result (18) shows that under the Assumption 1, deriving the algorithm directly via SMD can not guarantee the convergence for policy optimization. Furthermore, we propose a new algorithm MPO that keeps a provable convergence guarantee (see Theorem 2). Designing a new gradient estimator according to historical information of policy gradient is the key to MPO.

Then, we propose a variance-reduced mirror policy optimization algorithm (VRMPO): an efficient sample method via constructing a variance reduced policy gradient estimator. Concretely, we design an efficiently computable policy gradient estimator (see Eq.(26)) that utilizes fresh information and yields a more accurate estimation of the policy gradient, which is the key to improve sample efficiency. Theorem 3 illustrates that VRMPO needs $\mathcal{O}(\epsilon^{-3})$ sample trajectories to achieve an ϵ-approximate first-order stationary point (ϵ-FOSP). To our best knowledge, the proposed VRMPO matches the best sample complexity among the existing literature. Particularly, although SRVR-PG (Xu et al. 2020; Xu 2021) achieves the same sample complexity as VRMPO, our approach needs less assumptions than Xu et al. (2020); Xu (2021), and our VRMPO unifies SRVR-PG. Besides, empirical result shows VRMPO converges faster than SRVR-PG.

Background and Stochastic Mirror Descent

Reinforcement learning (RL) is often formulated as *Markov decision processes* (MDP) $M = (\mathcal{S}, \mathcal{A}, P, R, \rho_0, \gamma)$, where $\mathcal{S}$ is state space, $\mathcal{A}$ is action space; $P(s^{'}|s,a)$ is the probability of the state transition from s to $s^{'}$ under playing a; $R(\cdot,\cdot) : \mathcal{S} \times \mathcal{A} \to [-R_{\max}, R_{\max}]$ is the reward function, where $R_{\max}$ is a certain positive scalar. $\rho_0(\cdot) : \mathcal{S} \to [0, 1]$ is the initial state distribution and $\gamma \in (0, 1)$.

Policy $\pi_\theta(a|s)$ is a probability distribution on $\mathcal{S} \times \mathcal{A}$ with a parameter $\theta \in \mathbb{R}^p$. Let $\tau = \{s_t, a_t, r_{t+1}\}_{t=0}^{H_\tau}$ be a trajectory, where $s_0 \sim \rho_0(s_0)$, $a_t \sim \pi_\theta(\cdot|s_t), r_{t+1} = R(s_t, a_t)$, $s_{t+1} \sim P(\cdot|s_t, a_t)$, and H_τ is the finite horizon of τ. The

*L.Yang and Y.Zhang share equal contributions.
†G.Pan is the corresponding author.

expected return function $J(\theta)$ is defined as follows,

$$J(\theta) \stackrel{\text{def}}{=} \int_{\tau} P(\tau|\theta)R(\tau)\mathrm{d}\tau = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)], \quad (1)$$

where $P(\tau|\theta) = \rho_0(s_0)\prod_{t=0}^{H_\tau} P(s_{t+1}|s_t,a_t)\pi_\theta(a_t|s_t)$ is the probability of generating τ, $R(\tau) = \sum_{t=0}^{H_\tau}\gamma^t r_{t+1}$ is the accumulated discounted return. Let $\mathcal{J}(\theta) =: -J(\theta)$, the central problem of policy-based RL is to solve the problem:

$$\theta^\star = \arg\max_\theta J(\theta) \Longleftrightarrow \theta^\star = \arg\min_\theta \mathcal{J}(\theta). \quad (2)$$

Computing $\nabla J(\theta)$ analytically, we have

$$\nabla J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t\geq 0}\nabla_\theta \log\pi_\theta(a_t|s_t)R(\tau)\right]. \quad (3)$$

Let $g(\tau|\theta) = \sum_{t=0}^{H_\tau}\nabla_\theta\log\pi_\theta(a_t|s_t)R(\tau)$, which is an unbiased estimator of $\nabla J(\theta)$. Vanilla policy gradient (`VPG`) is a straightforward way to solve problem (2) as follows,

$$\theta \leftarrow \theta + \alpha g(\tau|\theta),$$

where α is step size.

Assumption 1. (Papini et al. 2018) *For each pair $(s,a)\in\mathcal{S}\times\mathcal{A}$, $\theta\in\mathbb{R}^p$, and all components i, j, there exists positive constants G, F such that:*

$$|\nabla_{\theta_i}\log\pi_\theta(a|s)| \leq G, \quad \left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\log\pi_\theta(a|s)\right| \leq F. \quad (4)$$

Assumption 1 implies $\nabla J(\theta)$ is L-Lipschiz (Papini et al. 2018, Lemma B.2), i.e.,

$$\|\nabla J(\theta_1) - \nabla J(\theta_2)\| \leq L\|\theta_1 - \theta_2\|, \quad (5)$$

where $L = R_{\max}H_\tau(H_\tau G^2 + F)/(1-\gamma)$, Besides, under Assumption 1, Shen et al. (2019) have shown the property:

$$\|g(\tau|\theta) - \nabla J(\theta)\|_2^2 \leq G^2 R_{\max}^2/(1-\gamma)^4 =: \sigma^2. \quad (6)$$

SMD and Bregman Gradient

Now, we review some basic concepts of stochastic mirror descent(SMD) and Bregman gradient.

Let's consider the stochastic optimization problem,

$$\min_{\theta\in D_\theta}\{f(\theta) = \mathbb{E}[F(\theta;\xi)]\}, \quad (7)$$

where $D_\theta \in \mathbb{R}^n$ is a nonempty convex compact set, ξ is a random vector whose probability distribution μ is supported on $\Xi \in \mathbb{R}^d$ and $F: D_\theta\times\Xi \to \mathbb{R}$. We assume that the expectation $\mathbb{E}[F(\theta;\xi)] = \int_\Xi F(\theta;\xi)\mathrm{d}\mu(\xi)$ is well defined and finite-valued for every $\theta\in D_\theta$.

Definition 1 (Proximal Operator)**.** *Let T be defined on a closed convex $\mathcal{X}$, and $\alpha > 0$. The proximal operator of T is*

$$\mathcal{M}^\psi_{\alpha,T}(z) = \arg\min_{x\in\mathcal{X}}\left\{T(x) + \frac{1}{\alpha}D_\psi(x,z)\right\}, \quad (8)$$

where $\psi(\cdot)$ is a continuously-differentiable, ζ-strictly convex function satisfies $\langle x-y, \nabla\psi(x)-\nabla\psi(y)\rangle \geq \zeta\|x-y\|^2, \zeta > 0$, $D_\psi(\cdot,\cdot)$ is Bregman distance: $\forall\, x,y\in\mathcal{X}$,

$$D_\psi(x,y) = \psi(x) - \psi(y) - \langle\nabla\psi(y), x-y\rangle.$$

Stochastic Mirror Descent (SMD). The SMD solves (7) by generating an iterative solution as follows,

$$\theta_{t+1} = \mathcal{M}^\psi_{\alpha_t,\ell(\theta)}(\theta_t) = \arg\min_{\theta\in D_\theta}\left\{\langle g_t,\theta\rangle + \frac{1}{\alpha_t}D_\psi(\theta,\theta_t)\right\}, \quad (9)$$

where $\alpha_t > 0$ is step-size, $\ell(\theta) = \langle g_t,\theta\rangle$ is the first-order approximation of $f(\theta)$ at θ_t, $g_t = g(\theta_t,\xi_t)$ is stochastic subgradient such that $g(\theta_t) = \mathbb{E}[g(\theta_t,\xi_t)] \in \partial f(\theta)|_{\theta=\theta_t}$, $\{\xi_t\}_{t\geq 0}$ represents a draw form distribution μ, and $\partial f(\theta) = \{g | f(\theta) - f(\omega) \leq g^\top(\theta-\omega), \forall\omega\in\mathbf{dom}(f)\}$. If we choose $\psi(x) = \frac{1}{2}\|x\|_2^2$, then $D_\psi(x,y) = \frac{1}{2}\|x-y\|_2^2$, since then iteration (9) is reduced to stochastic gradient decent (SGD).

Convergence Criteria: Bregman Gradient. Recall $\mathcal{X}$ is a closed convex set on $\mathbb{R}^n$, $\alpha > 0$, $T(x)$ is defined on $\mathcal{X}$. The Bregman gradient of T at $x\in\mathcal{X}$ is defined as:

$$\mathcal{G}^\psi_{\alpha,T}(x) = \alpha^{-1}(x - \mathcal{M}^\psi_{\alpha,T}(x)), \quad (10)$$

where $\mathcal{M}^\psi_{\alpha,T}(\cdot)$ is defined in Eq.(8). If $\psi(x) = \frac{1}{2}\|x\|_2^2$, according to Bauschke, Combettes et al. (2011, Theorem 27.1), then $x^\star$ is a critical point of T if and only if $\mathcal{G}^\psi_{\alpha,T}(x^\star) = \nabla T(x^\star) = 0$. Thus, Bregman gradient (10) is a generalization of standard gradient. Remark 1 provides us some insights to understand Bregman gradient as a convergence criterion.

Remark 1. *Let $T(\cdot)$ be a convex function, according to Bertsekas (2009, Proposition 5.4.7): $x^\star$ is a stationarity point of $T(\cdot)$ if and only if*

$$0 \in \partial(T + \delta_\mathcal{X})(x^\star), \quad (11)$$

where $\delta_\mathcal{X}(\cdot)$ is the indicator function on $\mathcal{X}$. Furthermore, if $\psi(x)$ is twice continuously differentiable, let $\tilde{x} = \mathcal{M}^\psi_{\alpha,T}(x)$, by the definition of $\mathcal{M}^\psi_{\alpha,T}(\cdot)$ (8), we have

$$\begin{aligned} 0 &\in \partial(T+\delta_\mathcal{X})(\tilde{x}) + \big(\nabla\psi(\tilde{x}) - \nabla\psi(x)\big) \\ &\overset{(*)}{\approx} \partial(T+\delta_\mathcal{X})(\tilde{x}) + \alpha\mathcal{G}^\psi_{\alpha,T}(x)\nabla^2\psi(x), \end{aligned} \quad (12)$$

Eq.($*$) holds due to Taylor expansion of $\nabla\psi(x)$ on first order. If $\mathcal{G}^\psi_{\alpha,T}(x) \approx 0$, Eq.(12) implies the origin point 0 is near the set $\partial(T+\delta_\mathcal{X})(\tilde{x})$, i.e., according to the criteria (11), $\tilde{x}$ is close to a stationary point. For the iteration (9), we focus on the time when it makes the $\mathcal{G}^\psi_{\alpha,T}(\theta_t)$ near origin point $\mathbf{0}$. Formally, we are satisfied with finding an ϵ-approximate first-order stationary point (ϵ-FOSP) θ_ϵ such that

$$\|\mathcal{G}^\psi_{\alpha,T(\theta_\epsilon)}(\theta_\epsilon)\|_2 \leq \epsilon. \quad (13)$$

Particularly, for policy optimization (2), we would choose $T(\theta) = \langle -\nabla J(\theta_t),\theta\rangle$.

Stochastic Mirror Policy Optimization

In this section, we solve the problem (2) via SMD. Firstly, we analyze the theoretical dilemma of applying SMD directly to policy optimization, and result shows that under the common Assumption 1, there still lacks a provable guarantee of solving (2) via SMD directly. Then, we propose a convergent mirror policy optimization algorithm (`MPO`).

Theoretical Dilemma

For each $k \in [1, N-1]$, $\tau_k = \{s_t, a_t, r_{t+1}\}_{t=0}^{H_{\tau_k}} \sim \pi_{\theta_k}$, and we receive the gradient information as follows,

$$-g(\tau_k|\theta_k) = -\sum_{t\geq 0} \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau_k)|_{\theta=\theta_k}. \quad (14)$$

According to (9), we define the update rule as follows,

$$\begin{aligned}\theta_{k+1} &= \mathcal{M}^{\psi}_{\alpha_k, \langle -g(\tau_k|\theta_k), \theta\rangle}(\theta_k) \quad (15)\\ &= \arg\min_\theta \left\{ \langle -g(\tau_k|\theta_k), \theta\rangle + \frac{1}{\alpha_k} D_\psi(\theta, \theta_k) \right\},\end{aligned}$$

where α_k is step-size. After $(N-1)$ episodes, we receive a collection $\{\theta_k\}_{k=1}^N$. Since $-J(\theta)$ is non-convex, according to Ghadimi, et al (2016), a standard strategy for analyzing non-convex optimization is to pick up the output $\tilde{\theta}_N$ from the following distribution (16) over $\{1, 2, \cdots, N\}$:

$$\mathbb{P}(\tilde{\theta}_N = \theta_k) = \frac{\zeta\alpha_k - L\alpha_k^2}{\sum_{i=1}^N (\zeta\alpha_i - L\alpha_i^2)}, k \in [1, N], \quad (16)$$

where step-size $\alpha_k \in (0, \zeta/L)$.

Theorem 1. (Ghadimi, et al (2016)) *Under Assumption 1, consider the sequence $\{\theta_k\}_{k=1}^N$ generated by (15), the output $\tilde{\theta}_N = \theta_k$ follows the distribution (16). Let $\ell(g, u) = \langle g, u\rangle$, $g_k = (\tau_k|\theta_k)$, Let $\Delta = J(\theta^\star) - J(\theta_1)$. Then,*

$$\mathbb{E}\left[\|\mathcal{G}^{\psi}_{\alpha_k, \ell(-g_k, \theta_k)}(\tilde{\theta}_N)\|_2^2\right] \leq \frac{\Delta + \sigma^2/\zeta \sum_{i=1}^N \alpha_i}{\sum_{i=1}^N (\zeta\alpha_i - L\alpha_i^2)}. \quad (17)$$

Unfortunately, the lower bound of (17) reaches

$$\frac{J(\theta^\star) - J(\theta_1) + \sigma^2/\zeta \sum_{i=1}^N \alpha_i}{\sum_{i=1}^N (\zeta\alpha_i - L\alpha_i^2)} \geq \frac{\sigma^2}{\zeta^2}, \quad (18)$$

which can not guarantee the convergence of (15), no matter how the step-size α_k is specified. Thus, under Assumption 1, updating parameters according to (15) and the output following (16) lacks a provable convergence guarantee.

Discussion 1 (Open Problems)**.** *Eq.(15) is a general rule that unifies many existing algorithms. If $\psi(\theta) = \frac{1}{2}\|\theta\|_2^2$, then (15) is* `VPG` *(Williams 1992). The update (15) is natural policy gradient (Kakade 2002) if we choose $\psi(\theta) = \frac{1}{2}\theta^\top F(\theta)\theta$, where $F(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) \nabla_\theta \log \pi_\theta(s, a)^\top]$ is Fisher information matrix. If ψ is Boltzmann-Shannon entropy, then D_ψ is KL divergence and update (15) is reduced to relative entropy policy search (Peters et al. 2010). Despite extensive works around above methods, existing works are scattered and fragmented in both theoretical and empirical aspects (Agarwal et al. 2020). Thus, it is of great significance to establish the fundamental theoretical convergence properties of iteration (15):*

What conditions guarantee the convergence of (15)?

This is an open problem. From the previous discussion, intuitively, the iteration (15) is a convergent scheme since particular mirror maps ψ can lead (15) to some popular empirically effective policy-based RL algorithms, but there still lacks a complete theoretical convergence analysis of (15).

Algorithm 1: MPO

1: **Initialize:** parameter θ_1, step-size$\alpha_k > 0$, $g_0 = 0$, parametric policy $\pi_\theta(a|s)$, and map ψ.
2: **for** $k = 1$ **to** N **do**
3: Generate a trajectory $\tau_k = \{s_t, a_t, r_{t+1}\}_{t=0}^{H_{\tau_k}} \sim \pi_{\theta_k}$, temporary variable $g_0 = 0$.

$$g_k \leftarrow \sum_{t=0}^{H_{\tau_k}} \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau_k)|_{\theta=\theta_k} \quad (21)$$

$$\hat{g}_k \leftarrow \frac{1}{k} g_k + (1 - \frac{1}{k})\hat{g}_{k-1} \quad (22)$$

$$\theta_{k+1} \leftarrow \arg\min_\omega \{\langle -\hat{g}_k, \omega\rangle + \frac{1}{\alpha_k} D_\psi(\omega, \theta_k)\} \quad (23)$$

4: **end for**
5: **Output** $\tilde{\theta}_N$ according to (16).

MPO: A Convergent Implementation

In this section, we propose a convergent mirror policy optimization (`MPO`) as follows, for each step k:

$$\begin{aligned}\theta_{k+1} &= \mathcal{M}^{\psi}_{\alpha_k, \langle -\hat{g}_k, \theta\rangle}(\theta_k)\\ &= \arg\min_{\theta\in\Theta} \{\langle -\hat{g}_k, \theta\rangle + \frac{1}{\alpha_k} D_\psi(\theta, \theta_k)\}, \quad (19)\end{aligned}$$

where $\hat{g}_k$ is an arithmetic mean of previous episodes' gradient estimate $\{g(\tau_i|\theta_i)\}_{i=1}^k$:

$$\hat{g}_k = \frac{1}{k} \sum_{i=1}^k g(\tau_i|\theta_i). \quad (20)$$

We present the details of an implementation of `MPO` in Algorithm 1. Eq.(22) is an incremental implementation of the average (20), thus, (22) enjoys a lower storage cost than (20).

For a given episode, the gradient flow (20)/(22) of `MPO` is slightly different from the traditional `VPG`, `REINFORCE` (Williams 1992), or `DPG` (Silver et al. 2014) whose gradient estimator (14) follows the current episode, while our `MPO` uses an arithmetic mean of all the previous policy gradients. The gradient estimator (14) is a natural way to estimate the term $-\nabla J(\theta_t) = -\mathbb{E}[\sum_{k=0}^{H_{\tau_t}} \nabla_\theta \log \pi_\theta(a_k|s_k) R(\tau_t)]$, i.e., using the current trajectory to estimate policy gradient.

Theorem 2 (Convergence of Algorithm 1)**.** *Under Assumption 1, and the total trajectories are $\{\tau_k\}_{k=1}^N$. Consider the sequence $\{\theta_k\}_{k=1}^N$ generated by Algorithm 1, and the output $\tilde{\theta}_N = \theta_n$ follows the distribution of (16). Let $0 < \alpha_k < \frac{\zeta}{L}$, $\ell(g, u) = \langle g, u\rangle$, $\hat{g}_k = \frac{1}{k}\sum_{i=1}^k g_i$, and $\Delta = J(\theta^\star) - J(\theta_1)$, where $g_i = \sum_{t=0}^{H_{\tau_i}} \nabla_\theta \log \pi_\theta(a_t|s_t) R(\tau_i)|_{\theta=\theta_i}$. Then the output $\tilde{\theta}_N = \theta_n$ satisfies*

$$\mathbb{E}[\|\mathcal{G}^{\psi}_{\alpha_n, \ell(-g_n, \theta_n)}(\theta_n)\|_2^2] \leq \frac{\Delta + \sigma^2/\zeta \sum_{k=1}^N \frac{\alpha_k}{k}}{\sum_{k=1}^N (\zeta\alpha_k - L\alpha_k^2)}. \quad (24)$$

For the proof, see Appendix A. Let $\alpha_k = \zeta/2L$, $\mathbb{E}[\|\mathcal{G}^{\psi}_{\alpha_n, \ell(-\hat{g}_n, \theta_n)}(\theta_n)\|^2] \leq \frac{4L\Delta + 2\sigma^2 \sum_{k=1}^N \frac{1}{k}}{N\zeta^2} = \mathcal{O}(\frac{\ln N}{N})$.

VRMPO: Variance Reduction Mirror Policy Optimization

In this section, we propose a variance reduction version of `MPO`: `VRMPO`. Inspired by the above work of (Nguyen et al. 2017a), we provide an efficiently computable policy gradient estimator; then, we prove that the `VRMPO` needs $\mathcal{O}(\epsilon^{-3})$ sample trajectories to achieve an ϵ-FOSP that matches the best sample complexity.

Methodology. For any initial θ_0, let $\{\tau_j^0\}_{j=1}^N \sim \pi_{\theta_0}$, we estimate the initial policy gradient as follows,

$$G_0 = -\hat{\nabla}_N J(\theta_0) \stackrel{\text{def}}{=} -\frac{1}{N}\sum_{j=1}^{N} g(\tau_j^0|\theta_0). \tag{25}$$

Let $\theta_1 = \theta_0 - \alpha G_0$, for each step $k \in \mathbb{N}^+$, let $\{\tau_j^k\}_{j=1}^N$ be the trajectories generated by π_{θ_k}, we define the policy gradient estimator G_k and update rule as follows,

$$G_k = G_{k-1} + \frac{1}{N}\sum_{j=1}^{N}\big(-g(\tau_j^k|\theta_k) + g(\tau_j^k|\theta_{k-1})\big), \tag{26}$$

$$\theta_{k+1} = \arg\min_{\theta}\{\langle G_k, \theta\rangle + \frac{1}{\alpha}D_\psi(\theta, \theta_k)\}. \tag{27}$$

In (26), $-g(\tau_j^k|\theta_k)$ and $g(\tau_j^k|\theta_{k-1})$ share the same trajectory $\{\tau_j^k\}_{j=1}^N$, which plays a critical role in reducing the variance of gradient estimator (Shen et al. 2019). Besides, it is different from (20), we admit a simple recursive formulation to conduct the gradient estimator, see (26), which captures the technique from `SARAH` (Nguyen et al. 2017a). For each step k, the term $\frac{1}{N}\sum_{j=1}^{N}\big(-g(\tau_j^k|\theta_k) + g(\tau_j^k|\theta_{k-1})\big)$ can be seen as an additional "noise" for the policy gradient estimate. A lot of practices show that conducting a gradient estimator with such additional "noise" enjoys a lower variance and speeding up the convergence (Reddi et al. 2016). More details are shown in Algorithm 2.

Theorem 3 (Convergence Analysis)**.** *Consider $\{\tilde{\theta}_k\}_{k=1}^K$ generated by Algorithm 2. Under Assumption 1, and let $\zeta > \frac{5}{32}$. For any positive scalar ϵ, let batch size of the trajectories of the outer loop $N_1 = \big(\frac{1}{8L\zeta^2} + \frac{1}{2(\zeta-\frac{5}{32})}\big(1+\frac{1}{32\zeta^2}\big)\big)\frac{\sigma^2}{\epsilon^2}$, $m-1 = N_2 = \sqrt{\big(\frac{1}{8L\zeta^2} + \frac{1}{2(\zeta-\frac{5}{32})}\big(1+\frac{1}{32\zeta^2}\big)\big)}\frac{\sigma}{\epsilon}$, the outer loop times $K = \frac{8L(\mathbb{E}[\mathcal{J}(\tilde{\theta}_0)]-\mathcal{J}(\theta^\star))(1+\frac{1}{16\zeta^2})}{\sqrt{\big(\frac{1}{8L\zeta^2}+\frac{1}{2(\zeta-\frac{5}{32})}\big(1+\frac{1}{32\zeta^2}\big)\big)}\big(\zeta-\frac{5}{32}\big)}\frac{\sigma}{\epsilon}$, and step size $\alpha = \frac{1}{4L}$. Then, Algorithm 2 outputs $\tilde{\theta}_K$ satisties*

$$\mathbb{E}\big[\|\mathcal{G}^{\psi}_{\alpha,\langle-\nabla J(\tilde{\theta}_K),\theta\rangle}(\tilde{\theta}_K)\|\big] \le \epsilon. \tag{30}$$

For its proof, see Appendix C. Theorem 3 illustrates that `VRMPO` needs $K(N_1+(m-1)N_2) = \frac{8L(\mathbb{E}[\mathcal{J}(\tilde{\theta}_0)]-\mathcal{J}(\theta^*))}{(\zeta-\frac{5}{32})}\big(1+\frac{1}{16\zeta^2}\big)\Big(1+\sqrt{\big(\frac{1}{8L\zeta^2}+\frac{1}{2(\zeta-\frac{5}{32})}\big(1+\frac{1}{32\zeta^2}\big)\big)}\frac{\sigma}{\epsilon}\Big)\frac{1}{\epsilon^2} = \mathcal{O}(\frac{1}{\epsilon^3})$ random trajectories to achieve the ϵ-FOSP. As far as we know, our `VRMPO` matches the best sample complexity as `HAPG` (Shen et al. 2019) and `SRVR-PG` (Xu et al. 2020; Xu 2021). In fact, according to Shen et al. (2019), `REINFORCE` needs $\mathcal{O}(\epsilon^{-4})$ random trajectories to achieve the ϵ-FOSP, and no provable improvement on its complexity has been made so far. The same order of sample complexity of `REINFORCE` is shown by Xu et al. (2019). With the additional assumptions $\mathbb{V}\mathrm{ar}[\prod_{h=0}^{H}\frac{\pi_{\theta_0}(a_h|s_h)}{\pi_{\theta_t}(a_h|s_h)}]$,$\mathbb{V}\mathrm{ar}[g(\tau|\theta)] < +\infty$, Papini et al. (2018) show that the `SVRPG` achieves the sample complexity of $\mathcal{O}(\epsilon^{-4})$. Later, under the same assumption as Papini et al. (2018), Xu et al. (2019) reduce the sample complexity of `SVRPG` to $\mathcal{O}(\epsilon^{-\frac{10}{3}})$. We summarize it in Table 1.

Algorithm 2: VRMPO.

Initialize: Policy $\pi_\theta(a|s)$ with parameter $\tilde{\theta}_0$, mirror map ψ, step-size $\alpha > 0$, epoch size K,m.
for $k = 1$ **to** K **do**
  $\theta_{k,0} = \tilde{\theta}_{k-1}$, generate $\mathcal{T}_k = \{\tau_i\}_{i=1}^{N_1} \sim \pi_{\theta_{k,0}}$
  $\theta_{k,1} = \theta_{k,0} - \alpha G_{k,0}$, where $G_{k,0} = -\hat{\nabla}_{N_1}J(\theta_{k,0}) = -\frac{1}{N_1}\sum_{i=1}^{N_1} g(\tau_i|\theta_{k,0})$.
  for $t = 1$ **to** $m-1$ **do**
    Generate $\{\tau_j\}_{j=1}^{N_2} \sim \pi_{\theta_{k,t}}$

$$G_{k,t} = G_{k,t-1} + \frac{1}{N_2}\sum_{j=1}^{N_2}(-g(\tau_j|\theta_{k,t}) + g(\tau_j|\theta_{k,t-1})), \tag{28}$$

$$\theta_{k,t+1} = \arg\min_{\omega}\{\langle G_{k,t}, \omega\rangle + \frac{1}{\alpha}D_\psi(\omega, \theta_{k,t})\}. \tag{29}$$

  end for
  $\tilde{\theta}_k = \theta_{k,t}$ with t chosen uniformly randomly from $\{0, 1, ..., m\}$.
end for
Output: $\tilde{\theta}_K$.

Remark 2. *It's remarkable that although our* `VRMPO` *shares sample complexity with* `HAPG`, `SRVR-PG`, *and* `VR-BGPO`*(Huang et al. 2021), the difference between our* `VRMPO` *and theirs are at least three aspects: Firstly, Shen et al. (2019) derive their* `HAPG` *from the information of Hessian policy, our* `VRMPO` *provides a simple recursive formulation to conduct the gradient estimator. Secondly, if the mirror map ψ is reduced to the ℓ_2-norm, then* `VRMPO` *is* `SRVR-PG` *exactly, i.e.,* `VRMPO` *unifies* `SRVR-PG`. *From Table 1, we see* `VRMPO` *needs less conditions than Xu et al. (2020) to achieve the same sample complexity. Finally, Shen et al. (2019), Xu et al. (2020) and Huang et al. (2021) only provide an off-line (i.e., Monte Carlo) policy gradient estimator, which is limited in complex domains. We have provided an on-line version of* `VRMPO`, *and discuss some insights of practical tracks to the application to the complex domains, please see the section of experiment on MuJoCo task, Appendix E.1.*

Related Works

Stochastic Variance Reduced Gradient in RL. To our best knowledge, Du et al. (2017) firstly introduce `SVRG` (Johnson and Zhang 2013) to off-policy evaluation (Yang et al. 2018).

Algorithm	Conditions	Complexity
VPG REINFORCE	Assumption 1 $\mathbb{V}\mathrm{ar}[g(\tau\|\theta)] < +\infty$	$\mathcal{O}(\epsilon^{-4})$
TRPO (Shani et al. 2020)	Assumption 1	$\mathcal{O}(\epsilon^{-4})$
TRPO (Liu et al. 2019)	Assumption 1	$\mathcal{O}(\epsilon^{-8})$
SVRPG (Papini et al. 2018)	Assumption 1 $\mathbb{V}\mathrm{ar}[\rho_t] < +\infty$ $\mathbb{V}\mathrm{ar}[g(\tau\|\theta)] < +\infty$	$\mathcal{O}(\epsilon^{-4})$
SVRPG (Xu et al. 2019)	Assumption 1; $\mathbb{V}\mathrm{ar}[\rho_t] < +\infty$ $\mathbb{V}\mathrm{ar}[g(\tau\|\theta)] < +\infty$	$\mathcal{O}(\epsilon^{-10/3})$
HAPG (Shen et al. 2019)	Assumption 1	$\mathcal{O}(\epsilon^{-3})$
SRVR-PG (Xu et al. 2020; Xu 2021)	Assumption 1 $\mathbb{V}\mathrm{ar}[\rho_t] < +\infty$ $\mathbb{V}\mathrm{ar}[g(\tau\|\theta)] < +\infty$	$\mathcal{O}(\epsilon^{-3})$
VR-PGPO (Huang et al. 2021)	Assumption 1 $\mathbb{V}\mathrm{ar}[\rho_t] < +\infty$ $\mathbb{V}\mathrm{ar}[g(\tau\|\theta)] < +\infty$	$\mathcal{O}(\epsilon^{-3})$
VRMPO (Our Work)	Assumption 1	$\mathcal{O}(\epsilon^{-3})$

Table 1: Comparison of complexity achieves $\|\nabla J(\theta)\| \le \epsilon$. If $\psi(\theta) = \frac{1}{2}\|\theta\|_2^2$, then the result (30) of our VRMPO is also measured by gradient. Beside, $\rho_t \overset{\text{def}}{=} \prod_{i=0}^{H} \frac{\pi_{\theta_0}(a_i|s_i)}{\pi_{\theta_t}(a_i|s_i)}$.

Du et al. (2017) transform the empirical policy evaluation problem into a convex-concave saddle-point problem, then they solve the problem via SVRG straightforwardly. Later, to improve sample efficiency for complex RL, Xu et al. (2017) combine SVRG with TRPO (Schulman et al. 2015). Similarly, Yuan et al. (2019) introduce SARAH (Nguyen et al. 2017a) to TRPO to improve sample efficiency. However, the results presented by Xu et al. (2017) and Yuan et al. (2019) are empirical, which lacks a strong theory analysis. Metelli et al. (2018) present a surrogate objective function with Rényi divergence (Rényi et al. 1961) to reduce the variance. Recently, Papini et al. (2018) propose a stochastic variance reduced version of policy gradient (SVRPG), and they define the gradient estimator via importance sampling:

$$\widetilde{G}_{k-1} + \frac{1}{N}\sum_{j=1}^{N}\Big(-g(\tau_j^k|\theta_t) + \prod_{i=0}^{H}\frac{\pi_{\theta_0}(a_i|s_i)}{\pi_{\theta_t}(a_i|s_i)} g(\tau_j^k|\theta_{t-1})\Big),$$

where $\widetilde{G}_{k-1}$ is an unbiased estimator according to the trajectory generated by $\pi_{\theta_{k-1}}$. Although SVRPG is practical empirically, its gradient estimate is dependent heavily on importance sampling. This fact partially reduces the effectiveness of variance reduction. Later, Shen et al. (2019) remove the importance sampling term, and they construct a Hessian aided policy gradient. Our VRMPO is different from Du et al. (2017); Xu, et al. (2017); Papini et al. (2018), which admits a stochastic recursive iteration to estimate the policy gradient. VRMPO exploits fresh information to improve convergence and reduces variance. Besides, VRMPO reduces the storage cost since it doesn't require to store the complete historical information.

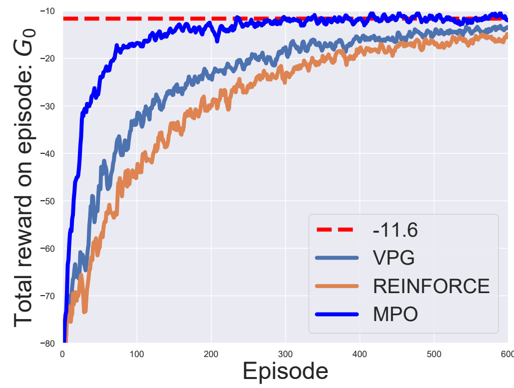


Figure 1: Convergence comparison between our MPO algorithm and REINFORCE/VPG on the SASC domain.

Baseline Methods. *Baseline* (also also known as control variates) is a widely used technique to reduce the variance (Weaver and Tao 2001; Greensmith et al. 2004). For example, A2C (Sutton and Barto 1998; Mnih et al. 2016) introduces the value function as baseline function, Wu et al. (2018) consider action-dependent baseline, and Liu et al. (2018) use the Stein's identity (Stein 1986) as baseline. Q-Prop (Gu et al. 2017) makes use of both the linear dependent baseline and GAE (Schulman et al. 2016) to reduce variance. Cheng et al. (2019) present a predictor-corrector framework transforms a first-order model-free algorithm into a new hybrid method that leverages predictive models to accelerate policy learning. Mao et al. (2019) derive a bias-free, input-dependent baseline to reduce variance, and analytically show its benefits over state-dependent baselines. Recently, Grathwohl et al. (2018); Cheng, et al. (2019) provide a standard explanation for the benefits of such approaches with baseline function. However, the capacity of all the above methods is limited by their choice of baseline function (Liu et al. 2018). In practice, it is troublesome to design a proper baseline function to reduce the variance of policy gradient estimate. Our VRMPO avoids the selection of baseline function, and it uses the current trajectories to construct a novel, efficiently computable gradient to reduce variance and improve sample efficiency.

Experiments

Our experiments cover the following three different aspects:

- We provide a numerical analysis of MPO, and compare the convergence rate of MPO with REINFORCE and VPG on the *Short Corridor with Switched Actions* (SASC) domain (Sutton and Barto 2018).
- We provider a better understand the effect of how the mirror map affects the performance of VRMPO.
- To demonstrate the stability and efficiency of VRMPO on the MuJoCo continuous control tasks, we provide a comprehensive comparison to state-of-the-art policy optimization algorithms.

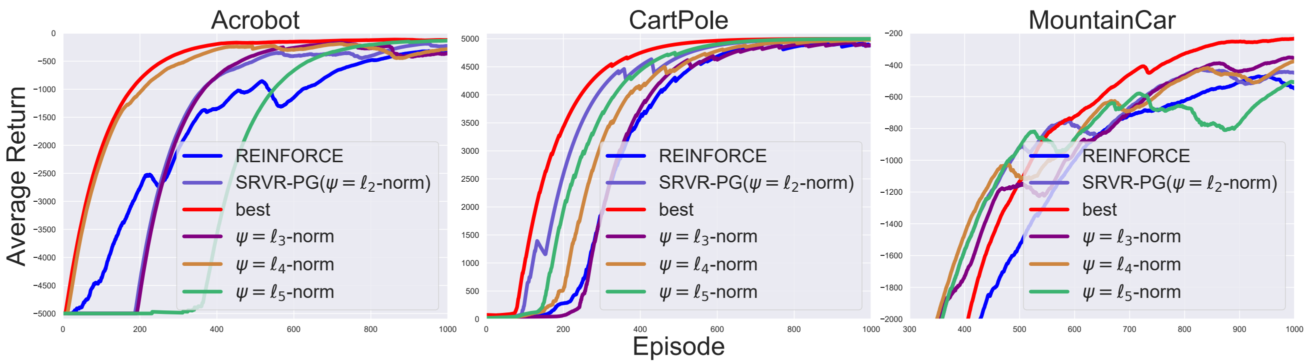


Figure 2: Comparison of the empirical performance of VRMPO between different mirror maps and REINFORCE.

Numerical Analysis of MPO

SASC Domain (see Appendix B): The task is to estimate the optimal value function of state $\mathtt{s}_1$, $V(\mathtt{s}_1) = G_0 \approx -11.6$. Let $\phi(s, \mathtt{right}) = [1, 0]^\top$ and $\phi(s, \mathtt{left}) = [0, 1]^\top$, $s \in \mathcal{S}$. Let $L_\theta(s, a) = \phi^\top(s, a)\theta$, $(s, a) \in \mathcal{S} \times \mathcal{A}$, where $\mathcal{A} = \{\mathtt{right}, \mathtt{left}\}$. $\pi_\theta(a|s)$ is the soft-max distribution defined as $\pi_\theta(a|s) = \frac{\exp\{L_\theta(s,a)\}}{\sum_{a' \in \mathcal{A}} \exp\{L_\theta(s,a')\}}$. The initial parameter $\theta_0 \sim \mathcal{U}[-0.5, 0.5]$, where $\mathcal{U}$ is the uniform distribution.

Before we report the results, it is necessary to explain why we only compare MPO with VPG and REINFORCE. VPG/REINFORCE is one of the most fundamental policy gradient methods in RL, and extensive modern policy-based algorithms are derived from VPG/REINFORCE. Our MPO is a new policy gradient algorithm to learn the parameter. Thus, it is natural to compare with VPG and REINFORCE. The result of Figure 1 shows that MPO converges faster significantly than both REINFORCE and VPG.

Effect of Mirror Map on VRMPO

If $\psi(\cdot)$ is ℓ_p-norm, then $\psi^\star(y) = (\sum_{i=1}^n |y_i|^q)^{\frac{1}{q}}$ is the conjugate map of ψ, where $y = (y_1, y_2, \cdots, y_n)^\top$, $\frac{1}{p} + \frac{1}{q} = 1$, and $p, q > 1$. According to Beck and Teboulle (2003), iteration (27) is equivalent to

$$\theta_{k+1} = \nabla\psi^\star(\nabla\psi(\theta_k) + \alpha G_k),$$

where $\nabla\psi_j(x) = \frac{\mathrm{sign}(x_j)|x_j|^{p-1}}{\|x\|_p^{p-2}}, \nabla\psi_j^\star(y) = \frac{\mathrm{sign}(y_j)|y_j|^{q-1}}{\|y\|_q^{q-2}}$, and j is coordinate index of the vector $\nabla\psi$, $\nabla\psi^\star$.

To compare fairly, we use the same random seed for each domain. The hyper-parameter p runs in the set $[P] = \{1.1, 1.2, \cdots, 1.9, 2, 3, 4, 5\}$. For the non-Euclidean distance case, we only show the results of $p = 3, 4, 5$ in Figure 2, and "best" is a certain hyper-parameter $p \in [P]$ achieves the best performance among the set $[P]$. We use a two-layer feedforward neural network of 200 and 100 hidden nodes, respectively, with rectified linear units (ReLU) activation function between each layer. We run the discounter $\gamma = 0.99$ and the step-size α is chosen by a grid search from the set $\{0.01, 0.02, 0.04, 0.08, 0.1\}$.

The result of Figure 2 shows that the best method is produced by non-Euclidean distance ($p \neq 2$), not the Euclidean distance ($p = 2$). The traditional policy gradient methods such as REINFORCE, VPG, and DPG are all the algorithms update parameters by Euclidean distance. This experiment gives us some light that one can create better algorithms with existing approaches via non-Euclidean distance. Additionally, the result of Figure 2 shows our VRMPO converges faster than REINFORCE, i.e., VRMPO needs less sampled trajectories to reach a convergent state, which supports the complexity analysis in Table 1. Although SRVR-PG achieves the same sample complexity as our VRMPO, result of Figure 2 shows VRMPO converges faster than SRVR-PG.

Evaluate VRMPO on Continuous Control Tasks

It is noteworthy that the policy gradient (26) of VRMPO is an off-line estimator likes REINFOECE. As pointed by Sutton and Barto (2018), REINFOECE converge asymptotically to a local minimum, but like all off-line methods, it is inconvenient for continuous control tasks, and it is limited in the application to some complex domains. This could also happen in VRMPO.

Now, we introduce some practical tricks for on-line implementation of VRMPO. We have provided the complete update rule of on-line VRMPO in Algorithm 3.

Details of Implementation. Firstly, we extend Algorithm 2 to be an actor-critic structure, i.e., we introduce a critic structure to Algorithm 2. Concretely, for each step t, we construct a critic network $Q_\omega(s, a)$ with the parameter ω, sample $\{(s_i, a_i)\}_{i=1}^N$ from a data memory $\mathcal{D}$, and learn the parameter ω via minimizing the critic loss as follows,

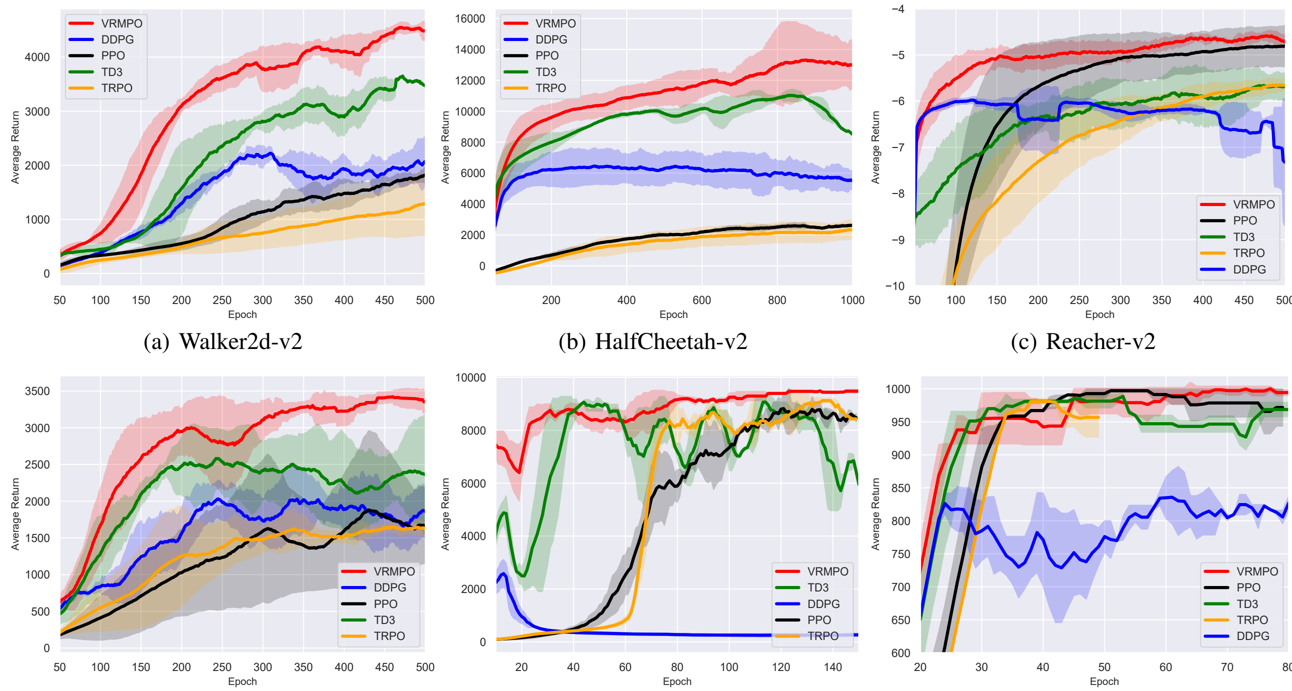
$$L_\omega = \frac{1}{N}\sum_{i=1}^N (r_{i+1} + \gamma Q_{\omega_{k-1}}(s_i, a_i) - Q_\omega(s_i, a_i))^2. \tag{31}$$

For more details, please see Line 17-20 of Algorithm 3. Then, for each pair $(s, a) \sim \mathcal{D}$, we conduct the actor loss

$$L_\theta(s, a) = -\log \pi_\theta(s, a) Q_{\omega_{k-1}}(s, a)$$

to replace $J(\theta)$ to learn parameter θ. For more details, please see Line 9-16 of Algorithm 3 (Appendix E.1).

Score Performance Comparison. From the results of Figure 3 and Table 2, overall, VRMPO outperforms the baseline algorithms in both final performance and learning process. Our VRMPO also learns considerably faster with better



(a) Walker2d-v2 (b) HalfCheetah-v2 (c) Reacher-v2

(d) Hopper-v2 (e) InvDoublePendulum-v2 (f) InvPendulum-v2

Figure 3: Learning curves for continuous control tasks. The shaded region represents the standard deviation of the score over the best three trials. Curves are smoothed uniformly for visual clarity.

performance than the popular TD3 on Walker2d, HalfCheetah, Hopper, InvDoublePendulum (IDP), and Reacher domains. On the InvDoublePendulum task, our VRMPO has only a small advantage over other algorithms. The InvPendulum task is relatively easy, the advantage of our VRMPO becomes more powerful when the task is more difficult. It is worth noticing that on the HalfCheetah domain, our VRMPO achieves a significant max-average score 16000+, which outperforms far more than the second-best score 11781.

Stability. The stability of an algorithm is also an important topic in RL. Although DDPG exploits the off-policy samples, which promotes its efficiency in stable environments. DDPG is unstable on the Reacher task, while our VRMPO learning faster significantly with lower variance. DDPG fails to make any progress on InvDoublePendulum domain, which is corroborated by (Dai et al. 2018). Although TD3 takes the minimum value between a pair of critics to limit overestimation, it learns severely fluctuating in the InvertedDoublePendulum environment. In contrast, our VRMPO is consistently reliable and effective in different tasks.

Variance Comparison. As we can see from the results in Figure 3, our VRMPO converges with a considerably low variance in the Hopper, InvDoublePendulum, and Reacher. Although the asymptotic variance of VRMPO is slightly larger than other algorithms in HalfCheetah, the final performance of VRMPO outperforms all the baselines significantly. The result of Figure 3 also implies conducting a proper gradient estimator not only reduces the variance of the score during the learning but speeds the convergence of training.

Environment	VRMPO	TD3	DDPG	PPO	TRPO
Walker2d	5251.83	4887.85	**5795.13**	3905.99	3636.59
HalfCheetah	**16095.51**	11781.07	8616.29	3542.60	3325.23
Reacher	-0.49	-1.47	-1.55	**-0.44**	-0.66
Hopper	**3751.43**	3482.06	3558.69	3609.65	3578.06
IDP	**9359.82**	9248.27	6958.42	9045.86	9151.56
InvPendulum	**1000.00**	**1000.00**	907.81	**1000.00**	**1000.00**

Table 2: Max-average return over final 50 epochs, where we run 5000 iterations for each epoch. Maximum value for each task is bolded.

Conclusion

In this paper, we analyze the theoretical dilemma of applying SMD to policy optimization. Then, we propose a sample efficient algorithm VRMPO, and prove the sample complexity of VRMPO achieves only $\mathcal{O}(\epsilon^{-3})$. To our best knowledge, VRMPO matches the best sample complexity so far. Finally, we conduct extensive experiments to show our algorithm outperforms state-of-the-art policy gradient methods.

Acknowledgements

This work is supported by China Brain Project (2021ZD0200400), Natural Science Foundation of China (No. 61925603), Zhejiang Lab (No. 2018EB0ZX01), and Ten Thousand Talent Program of Zhejiang Province (No.2018R52039).

References

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2020. Optimality and Approximation with Policy Gradient Methods in Markov Decision Processes. *COLT*.

Bauschke, H. H.; Combettes, P. L.; et al. 2011. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408.

Beck, A.; and Teboulle, M. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3): 167–175.

Bertsekas, D. P. 2009. *Convex optimization theory*. Athena Scientific Belmont.

Cheng, C.-A.; Yan, X.; and Boots, B. 2019. Trajectory-wise Control Variates for Variance Reduction in Policy Gradient Methods. *ICRA*.

Cheng, C.-A.; Yan, X.; Ratliff, N.; and Boots, B. 2019. Predictor-Corrector Policy Optimization. In *ICML*.

Dai, B.; Shaw, A.; Li, L.; Xiao, L.; He, N.; Liu, Z.; Chen, J.; and Song, L. 2018. SBEED: Convergent reinforcement learning with nonlinear function approximation. *ICML*.

Du, S. S.; Chen, J.; Li, L.; Xiao, L.; and Zhou, D. 2017. Stochastic variance reduction methods for policy evaluation. In *ICML*.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *ICML*.

Ghadimi, S.; Lan, G.; and Zhang, H. 2016. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1-2): 267–305.

Grathwohl, W.; Choi, D.; Wu, Y.; Roeder, G.; and Duvenaud, D. 2018. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *ICLR*.

Greensmith, E.; Bartlett, P. L.; Baxter, J.; et al. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *JMLR*, 5(Nov): 1471–1530.

Gu, S.; Lillicrap, T.; Ghahramani, Z.; Turner, R. E.; and Levine, S. 2017. Q-prop: Sample-efficient policy gradient with an off-policy critic. *ICLR*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*.

Huang, F.; Gao, S.; Huang, H.; and et.al. 2021. Bregman gradient policy optimization. *arXiv preprint arXiv:2106.12112*.

Johnson, R.; and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, 315–323.

Kakade, S. M. 2002. A natural policy gradient. In *NeurIPS*, 1531–1538.

Lei, Y.; and Tang, K. 2018. Stochastic composite mirror descent: optimal bounds with high probabilities. In *NeurIPS*, 1519–1529.

Liu, B.; Cai, Q.; Yang, Z.; and Wang, Z. 2019. Neural proximal/trust region policy optimization attains globally optimal policy. *NeurIPS*.

Liu, H.; Feng, Y.; Mao, Y.; Zhou, D.; Peng, J.; and Liu, Q. 2018. Action-dependent control variates for policy optimization via stein identity. *ICLR*.

Mao, H.; Venkatakrishnan, S. B.; Schwarzkopf, M.; and Alizadeh, M. 2019. Variance reduction for reinforcement learning in input-driven environments. *ICLR*.

Metelli, A. M.; Papini, M.; Faccio, F.; and Restelli, M. 2018. Policy optimization via importance sampling. In *NeurIPS*, 5442–5454.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *ICML*, 1928–1937.

Nemirovskij, A. S.; and Yudin, D. B. 1983. *Problem complexity and method efficiency in optimization*. Wiley-Interscience.

Nguyen, L. M.; Liu, J.; Scheinberg, K.; and Takáč, M. 2017a. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *ICML*.

Papini, M.; Binaghi, D.; Canonaco, G.; and Matteo Pirotta, M. R. 2018. Stochastic Variance-Reduced Policy Gradient. In *ICML*.

Peters, J.; Mülling, K.; Altun; and Yasemin. 2010. Relative Entropy Policy Search. In *AAAI*, 1607–1612.

Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. In *ICML*, 314–323.

Rényi, A.; et al. 1961. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *ICML*, 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-dimensional continuous control using generalized advantage estimation. *ICLR*.

Shani, L.; Efroni, Y.; Mannor, S.; and et.al. 2020. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *AAAI*, volume 34, 5668–5675.

Shen, Z.; Ribeiro, A.; Hassani, H.; Qian, H.; and Mi, C. 2019. Hessian Aided Policy Gradient. In *ICML*, 5729–5738.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *ICML*.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of Go without human knowledge. *Nature*, 550(7676): 354.

Stein, C. 1986. Approximate computation of expectations. *Lecture Notes-Monograph Series*, 7: i–164.

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

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *NeurIPS*, 1057–1063.

Weaver, L.; and Tao, N. 2001. The optimal reward baseline for gradient-based reinforcement learning. In *UAI*, 538–545. Morgan Kaufmann Publishers Inc.

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

Wu, C.; Rajeswaran, A.; Duan, Y.; Kumar, V.; Bayen, A. M.; Kakade, S.; Mordatch, I.; and Abbeel, P. 2018. Variance reduction for policy gradient with action-dependent factorized baselines. *ICLR*.

Xing, D.; Liu, Q.; Zheng, Q.; and Pan, G. 2021. Learning with Generated Teammates to Achieve Type-Free Ad-Hoc Teamwork. In *IJCAI*.

Xu, P. 2021. *Sample-Efficient Nonconvex Optimization Algorithms in Machine Learning and Reinforcement Learning*. Ph.D. thesis, UCLA.

Xu, P.; Gao, F.; Gu, Q.; et al. 2019. An Improved Convergence Analysis of Stochastic Variance-Reduced Policy Gradient. *UAI*.

Xu, P.; Gao, F.; Gu, Q.; et al. 2020. Sample efficient policy gradient methods with recursive variance reduction. *ICLR*.

Xu, T.; Liu, Q.; and Peng, J. 2017. Stochastic Variance Reduction for Policy Gradient Estimation. *arXiv preprint arXiv:1710.06034*.

Yang, L.; Ji, J.; Dai, J.; Zhang, Y.; Li, P.; and Pan, G. 2022. CUP: A Conservative Update Policy Algorithm for Safe Reinforcement Learning. *arXiv preprint arXiv:2202.07565*.

Yang, L.; Shi, M.; Zheng, Q.; Meng, W.; and Pan, G. 2018. A unified approach for multi-step temporal-difference learning with eligibility traces in reinforcement learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2984–2990.

Yang, L.; Zheng, Q.; ; and Pan, G. 2021. Sample complexity of policy gradient finding second-order stationary points. In *AAAI*.

Yuan, H.; Li, C. J.; Tang, Y.; and Zhou, Y. 2019. Policy optimization via stochastic recursive gradient algorithm. https://openreview.net/forum?id=rJl3S2A9t7.