# Efficient Nonconvex Learning for Policy Gradient Algorithm via Stochastic Gradient MCMC

**Jinke Li**
Department of Applied Mathematics and Statistics
Johns Hopkins University
jli300@jh.edu

## Abstract

Policy gradient (PG) algorithms have been widely applied to reinforcement learnig (RL) problems in recent years. However, PG algorithms are not straightforward and need to account for I) sample efficiency in the non-stationary sampling process; II) the non-concave objective function. In this work, we re-think the optimization problem as an instance of sampling a complex probability distribution. Our improvements follows from the powerful gradient-based Monte Carlo methods, yielding a global convergence guarantee for PG methods; moreover, we employ the efficient gradient estimator for reducing variance in sub-sampling with the unknown environment. Empirically, we test the algorithm through experiments on classic control and Atari games. The results shed some light on using nonconvex learning algorithms in RL tasks.

## 1 Introduction

A major work behind the recent success in reinforcement learning is the large family of policy gradient methods (Williams,1992 [20]; Sutton et al., 1999 [18]), for example, the natural policy gradient (NPG) method (Kakade, 2001 [11]), the actor-critic method (Konda & Tsitsiklis, 2000 [12]), the trust region policy optimization (TRPO) (Schulman et al., 2015a [16] ). Policy gradient method [18] parameterizes policy by an unknown parameter $\theta$ and find the optimal policy by optimizing $\theta$. However, like MAP estimate $\theta = \arg\max\{\log(p(\theta) + \sum_{i=1}^{N} p(d_i|\theta)\}$ in deep learning, the objective function $J(\theta)$ is usually non-convex (non-concave). For stochastic optimization problem, it is intractable to compute the gradient of $J(\theta)$ exactly, the common gradient estimators like REINFORCE (Williams et al. 1992 [20]), GPOMDP (Baxter et al. 2001 [2]) require $O(1/\epsilon^2)$ trajectories to find an stationary point such that $E[\|J(\theta)\|_2^2] \leq \epsilon$. Stochastic Recursive Variance Reduced Policy Gradient (SRVR-PG) (Xu et al., 2020 [21]) achieve a state-of-the-art sample complexity in policy gradient algorithms, which only takes $O(1/\epsilon^{3/2})$ trajectories to converge to an $\epsilon$-stationary point. This paper continues on this line of research, aiming to reduce variance in the proposed algorithm.

In contrast to the empirical success of PG methods, their theoretical convergence guarantees, have not been addressed satisfactorily. Various first-order optimization algorithms such as stochastic gradient descent (Ghadimi et al., [9] and more recently variance-reduced stochastic gradient descent (Reddi et al., [15]; Allen-Zhu et al., [1]) are only guaranteed to converge to a stationary point, which might be a local minimum or a saddle point. The core question in solving a non-convex optimization problem is: can we design an efficient algorithm that is guaranteed to converge to a global minimum? Jalaj et al. (2020) established the global optimality guarantees for PG methods, which can explain the effectiveness of PG methods in some special settings such as finite states and actions MDPs. However, most RL tasks have continuous actions and states, leading the suboptimal stationary points in vanilla (stochastic) gradient updates. Dalalyan (2016, 2017) [6] [5] showed that sampling from a distribution which concentrates around the global minimum of objective function $F(\mathbf{x})$ is a similar

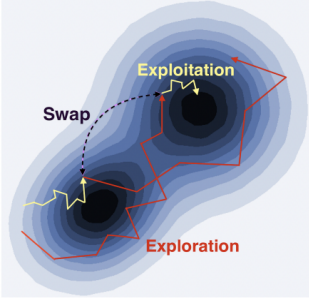



Figure 1: Paths of reSG-MCMC (Wei Deng et al. (2020) [7]

task as minimizing $F(\mathbf{x})$ via certain optimization models. This justifies the use of stochastic gradient Langevin dynamics (SGLD) (Weillin and Teh, 2011 [19]). The first order Langevin dynamics is defined by the following stochastic differential equation

$$d\mathbf{X}_t = -\nabla F(\mathbf{X}_t)dt + \sqrt{2\beta}d\mathbf{W}_t \tag{1}$$

where $\beta > 0$ is the "temperature" parameter, and $\mathbf{W}_t$ is the standard Brownian motion. Under certain assumptions, one can show that the distribution of diffusion $\mathbf{X}_t$ converge to its unique stationary distribution (Chiang, 1987 [4]), the Gibbs distiburion $\pi_\beta(\mathbf{X}) \propto \exp(-F(\mathbf{X})/\beta)$, which concentrates on the global minimum of $F$ (Hwang, 1980 [10]; Gelfand, 1991 [8]). Unlike the first order algorithms for general unconstrained optimization which have been well studied, the theoretical guarantee of SGLD is still under studied. Raginsky et al. (2017)[14] showed that SGLD converges to an almost minimizer, Simsekli et al. (2018) [17] proposed an stochastic L-BFGS algorithm for non-convex optimization based on SGLD. In this case, we can observe that SGLD algorithm is an efficient sampling algorithm in non-convex learning, where the techniques are called Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC).

In most recent work, the variants of SGLD has been proposed to accelerate the vanilla Langevin Dynamics algorithm. Wei Deng et al. (2020) [7] developed Replica Excange Stochastic Gradient MCMC (reSG-MCMC), which uses parallel Langevin diffusion, one for *exploration*, another for *exploitation*. As shown in Fig.1, the exploration process has a high-temperature $\beta$, while the exploitation process is with a low-temperature $\beta$. The high-temperature $\beta$ achieves the exploration effect, facilitating the particle to converge to the Gibbs distribution of the whole domain, and the low-temperature $\beta$ explore the local geometry rapidly, but more likely to get trapped in local optima. The two particles are allowed to swap, which lead to a good approximation to the correct distribution.

**Our Contributions** Motivated by these advances and the questions that remain to be answered, we aim in this paper to improve the convergence of PG methods, and their variance-reduced variants. We have designed a algorithm based on SRVR-PG, and establish its global convergence of 2-Wasserstein distance between the current and target distribution with the reSG-MCMC, which extends the Bhandari's work (). These improvements are verified in numerical reinforcement learning tasks, especially in environments with continuous actions. In addition, our results can deepen our understanding of policy gradient methods, which demonstrate that the sample-efficient gradient estimator is as essential as the global optimality.

## 2 Preliminaries

Consider a discounted Markov decision process $M$ defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces of the agent. $\mathcal{P}$ is a Markov transition model, where $\mathbb{P}(s^{'}|s, a)$ defines the transition density from states $s$ to $s^{'}$ under action $a$. $\mathcal{R}$ is the reward function, where $\mathcal{R}(s, a) \in [-R, R]$ is the expected reward for state-action pair $(s, a)$. The agent's behaviour is controlled by a policy $\pi$, where $\pi(\cdot \mid s)$ denotes the density distribution over $\mathcal{A}$ in state $s$. Interacting with the environments, a trajectory $\tau$ is a sequence of states and actions $(s_0, a_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1})$. We rank the policy based on their expected total reward: $J(\pi) = \mathbb{E}_{\tau \sim p(\cdot|\pi)}[\mathcal{R}(\tau) \mid M]$. The optimization problem is to find the optimal policy $\pi^* \in \arg\max_\pi \{J(\pi)\}$. Searching for a locally optimal policy is performed through gradient ascent, where the policy gradient theorem (Sutton et al,

2000) states:

$$\nabla J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} [\nabla \log p_{\boldsymbol{\theta}}(\tau) \mathcal{R}(\tau)] \tag{2}$$

However, exact computation is intractable in (2) since we don't know the distribution of $p_\theta(\tau)$. The most common policy gradient estimators (e.g., REINFORCE (Williams, 1992)[]) and GPOMDP (Baxter & Bartlett, 2001 []) can be expressed as follows

$$\widehat{\nabla}_N J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} g\left(\tau_i \mid \boldsymbol{\theta}\right), \tag{3}$$

where $\{\tau_i\}_{i=1}^N$ is a batch of trajectories collected from the interaction with the environment, and $g(\tau_i|\theta)$ is an unbiased estimator of $\nabla \log p_{\boldsymbol{\theta}}\left(\tau_i\right) \mathcal{R}\left(\tau_i\right)$. The GPOMDP is usually useful due to its lower variance, we use the GPOMDP as the basic estimator in the whole paper. The GPOMDP estimator is defined as follows:

$$g\left(\tau_i \mid \boldsymbol{\theta}\right) = \sum_{h=0}^{H-1} \left( \sum_{t=0}^{h} \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}\left(a_t^i \mid s_t^i\right) \right) \left(\gamma^h r\left(s_h^i, a_h^i\right)\right) \tag{4}$$

## 3 Variance-Reduced Policy Gradient Methods

Recently, Xu et al. [] proposes an algorithm called Stochastic Recursive Variance Reduced Policy Gradient (SRVR-PG), which applies variance-reduction on PG. It achieves a sample complexity of $O(\epsilon^{-1.5})$ to find a $\epsilon$-stationary point, compared with the $O(1/\epsilon^2)$ sample complexity of GPOMDP. SRVR-PG algorithm contains two steps within a epoch: the first is to sample $N$ trajectories and estimate the gradient $\mathbf{v}_0$ using GPOMDP; the second is to run $t$ iterations, sampling $B$ trajectories based on current policy parameters, then with the recursive semi-stochastic gradient estimator:

$$\mathbf{v}_t = \frac{1}{B} \sum_{j=1}^{B} g\left(\tau_j \mid \boldsymbol{\theta}_t\right) - \frac{1}{B} \sum_{j=1}^{B} g_\omega\left(\tau_j \mid \boldsymbol{\theta}_t\right) + \mathbf{v}_{t-1}, \tag{5}$$

the algorithm can effectively reduce the variance. In (5), the step-wise importance weighted estimator aims to obtain an unbiased GPOMDP estimator of previous parameter $\boldsymbol{\theta}_{t-1}$ using trajectories sampled by current $\boldsymbol{\theta}_t$:

$$g_\omega\left(\tau_j \mid \boldsymbol{\theta}_t\right) = \sum_{h=0}^{H-1} \omega_{0:h}\left(\tau \mid \boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t\right) \left( \sum_{t=0}^{h} \nabla_{\boldsymbol{\theta}_{t-1}} \log \pi_{\boldsymbol{\theta}_{t-1}}\left(a_t^j \mid s_t^j\right) \right) \gamma^h r\left(s_h^j, a_h^j\right), \tag{6}$$

where $\omega_{0:h}\left(\tau \mid \boldsymbol{\theta}_2, \boldsymbol{\theta}_1\right) = \prod_{h'=0}^{h} \pi_{\theta_2}\left(a_h \mid s_h\right) / \pi_{\theta_1}\left(a_h \mid s_h\right)$ is the importance weight. The whole algorithm is expressed in Algorithm 1. The core of this algorithm is the recursive semi-stochastic policy gradient, which is inspired from the stochastic path-integrated differential estimator (Fang et al., 2018 []). The following theorem states that SRVR-PG has the state-of-the-art sample complexity in PG methods, where all the assumptions are summarized in []

**Theorem 1.** *Set step size as $\eta = 1/(4L)$, the batch size parameters as $N = O(1/\epsilon)$ and $B = O\left(1/\epsilon^{1/2}\right)$ respectively, epoch length as $m = O\left(1/\epsilon^{1/2}\right)$ and the number of epochs as $S = O\left(1/\epsilon^{1/2}\right)$. Then Algorithm 1 outputs a point $\theta_{out}$ that satisfies $\mathbb{E}\left[\|\boldsymbol{\theta}_{out}\|_2^2\right] \leq \epsilon$ within $O\left(1/\epsilon^{3/2}\right)$ trajectories in total.*

## 4 The Proposed Algorithm

The SRVR-PG algorithm only guarantees to find an $\epsilon$-stationary point, which could be a suboptimal stationary point. Recently, there has been a surging research interest in investigating the global convergence of PG methods, which is beyond the convergence to first-order stationary policies. In the special case with linear dynamics and quadratic reward, [] shows that PG methods with random search converge to the globally optimal policy with linear rates, and [] show that for finite-MDPs and several control tasks, the nonconvex objective has no suboptimal local mimima. However, with continuous

**Algorithm 1:** Stochastic Recursive Variance Reduced Policy Gradient

**input**: number of epochs $S$, epoch size $m$, step size $\eta$, batch size $N$, mini-batch size $B$, gradient estimator $g$, initial parameter $\widetilde{\boldsymbol{\theta}}^0 = \boldsymbol{\theta}_0$

**for** $s = 0, 1, ..., S-1$ **do**

$\boldsymbol{\theta}_0^{s+1} = \widetilde{\boldsymbol{\theta}}^s$

Sample $N$ trajectories $\{\tau_i\}$ from $p\left(\cdot \mid \widetilde{\boldsymbol{\theta}}^s\right)$

$\mathbf{v}_0^{s+1} = \widehat{\nabla}_\theta J\left(\widetilde{\boldsymbol{\theta}}^s\right) := 1/N \sum_{i=1}^N g\left(\tau_i \mid \tilde{\boldsymbol{\theta}}^s\right)$

$\boldsymbol{\theta}_1^{s+1} = \boldsymbol{\theta}_0^{s+1} + \eta \mathbf{v}_0^{s+1}$

**for** $t = 1, ..., m-1$ **do**

Sample $B$ trajectories $\{\tau_j\}$ from $p\left(\cdot \mid \boldsymbol{\theta}_t^{s+1}\right)$

$\mathbf{v}_t^{s+1} = \mathbf{v}_{t-1}^{s+1} + \frac{1}{B}\sum_{j=1}^B \left(g\left(\tau_j \mid \boldsymbol{\theta}_t^{s+1}\right) - g_\omega\left(\tau_j \mid \boldsymbol{\theta}_{t-1}^{s+1}\right)\right)$

$\boldsymbol{\theta}_{t+1}^{s+1} = \boldsymbol{\theta}_t^{s+1} + \eta \mathbf{v}_t^{s+1}$

**end**

$\tilde{\boldsymbol{\theta}}^{s+1} = \boldsymbol{\theta}_m^{s+1}$

**end**

**return** $\boldsymbol{\theta}_{\text{out}}$, which is uniformly picked form $\{\boldsymbol{\theta}_t^s\}_{t=0,...,m-1;s=0,...,S}$

action space and observation space, things are complicated. The stochastic gradient Monte Carlo methods are powerful in nonconvex learning, which treats the optimization as a posterior sampling problem. The most popular variant of stochastic gradient descent is stochastic gradient Langevin Dynmaics (SGLD), given a possibly non-convex function $\widetilde{L}$, SGLD performs the iterative update:

$$\widetilde{\boldsymbol{\beta}}_{k+1} = \widetilde{\boldsymbol{\beta}}_k - \eta_k \nabla \widetilde{L}\left(\widetilde{\boldsymbol{\beta}}_k\right) + \sqrt{2\eta_k \tau_1}\boldsymbol{\xi}_k,$$

where $\eta_k$ is the learning rate, the stochastic gradient $\nabla \widetilde{L}\left(\widetilde{\boldsymbol{\beta}}_k\right)$ is the unbiased estimator of the exact gradient, $\xi$ is a standard $d$-dimensional Gaussian vector with mean $\mathbf{0}$ and identity covariance matrix. However, despite their scalability with respect to the data size, their mixing rates are often extremely slow for complex networks with rugged energy landscapes. To speed up the convergence, the replica exchange stochastic gradient Langevin dynamics (reSGLD) (Deng et al., [7] uses multiple processes based on SGLD where interactions between different SGLD chains are conducted in a manner that encourages large jumps of parameters. Using the same notations, the replica exchange gradient Langevin dynamics is defined as follows:

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)} = \widetilde{\boldsymbol{\beta}}_k^{(1)} - \eta_k \nabla \widetilde{L}\left(\widetilde{\boldsymbol{\beta}}_k^{(1)}\right) + \sqrt{2\eta_k \tau_1}\boldsymbol{\xi}_k^{(1)} \tag{7}$$

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)} = \widetilde{\boldsymbol{\beta}}_k^{(2)} - \eta_k \nabla \widetilde{L}\left(\widetilde{\boldsymbol{\beta}}_k^{(2)}\right) + \sqrt{2\eta_k \tau_2}\boldsymbol{\xi}_k^{(2)} \tag{8}$$

Moreover, we can swap the chains in (7) and (8) with the stochastic swapping rate

$$\mathbb{S}\left(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}\right) = e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}\left(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}\right) - \widetilde{L}\left(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}\right)\right)}$$

With the swapping rate, the lower temperature particle can jump to the higher temperature particle's position, and then do the "exploitation" work. Surprisingly, this swap can accelerate the sampling process. We present the results from [7] to see that the algorithm has exponential convergence rate respect to 2-Wasserstein distance.

**Lemma 2.** *Suppose there exists a Lipschitz constant $C > 0$, such that for every $x, y \in \mathbb{R}^d$, we have $\|\nabla L(x) - \nabla L(y)\| \leq C\|x - y\|$. Moreover, there exist constant $\alpha > 0$ and $b \leq 0$ such that $\forall x \in \mathbb{R}^d, \langle x, \nabla U(x)\rangle \geq a\|x\|^2 - b$. Then, with a small learning rate $\eta$, we have that*

$$\mathbb{E}\left[\sup_{0\leq t\leq T}\left\|\boldsymbol{\beta}_t - \widetilde{\boldsymbol{\beta}}_t^\eta\right\|^2\right] \leq \tilde{\mathcal{O}}\left(\eta + \max_i \mathbb{E}\left[\|\phi_i\|^2\right] + \max_i \sqrt{\mathbb{E}\left[|\psi_i|^2\right]}\right)$$

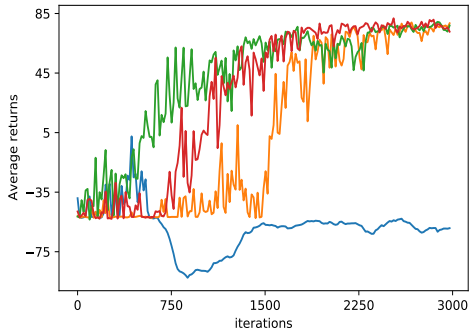



Figure 2: Some extreme cases of SRVR-PG on MountainCarContinous

*, where $\boldsymbol{\beta}_t$ is the SDE diffusion of Langevin dynamics, and $\widetilde{\boldsymbol{\beta}}_t^{\eta}$ is our discretization particle, $\phi := \nabla\widetilde{L} - \nabla L$ is the noise of the estimator, and $\psi := \widetilde{S} - S$ is the noise of swapping rate.*

Using Wasserstein distance to measure the optimal gap, we have

**Theorem 3.** *Let the assumption in Lemma 1 hold, for the distribution $\mu_{kk\leq 0} =$ associated with our discrete particle $\left\{\tilde{\boldsymbol{\beta}}_k\right\}_{k\geq 1}$, we have the following estimates for $k \in \mathbb{N}^+$,*

$$\mathcal{W}_2\left(\mu_k, \pi\right) \leq D_0 e^{-k\eta(1+\delta_S)/c_{LS}} + \tilde{\mathcal{O}}\left(\eta^{\frac{1}{2}} + \max_i\left(\mathbb{E}\left[\|\phi_i\|^2\right]\right)^{\frac{1}{2}} + \max_i\left(\mathbb{E}\left[|\psi_i|^2\right]\right)^{\frac{1}{4}}\right) \tag{9}$$

*where* $D_0 = \sqrt{2c_{LS}D\left(\mu_0\|\pi\right)}, \delta_S := \min_i \dfrac{\mathcal{E}_S\left(\sqrt{\frac{d\mu_i}{d\pi}}\right)}{\mathcal{E}\left(\sqrt{\frac{d\mu_i}{d\pi}}\right)} - 1.$

This analysis sheds light on the accelerated convergences in terms of 2-Wasserstein distance, which is useful in implementing the PG methods in RL tasks. We expect to use Langevin Dynamics and its variants to facilitate the non-convex optimization in policy gradient methods. The new algorithm called SRVR-reSGLD we propose is to incorporate variance reduction into sampling process of Langevin dynamics, which is describe in Algorithm 2. The sample efficient estimator is essential since our optimal convergence gap contains the error led by noisy gradient estimators. From numerical experiments, the SRVR-PG sometimes falls into a bad local trap, which makes the learning process almost stop. The Figure 2 shows that with one optimizer, the algorithm may face some difficulty to behave normal with any random seeds. Thus, the "exploration" chain in the reSGLD is effective for the agent to avoid the local trap.

## 5 Experiments

In this section, we evaluate the performance of SRVRPG-reSGLD on well known RL tasks: Pendulum, Hopper, Mountain Car, HalfCheetah. Due to our computation restriction, we only present one experiment on Mujoco task, while the classic control uses 5 experiments. The Mountain Car and Pendulum are classic control tasks in Gym, while the Hopper and HalfCheetah are more difficult 3D-continuous-control locomotion task.

For all four tasks, we use a Gaussian policy with a neural network. For baselines, we compare the proposed SRVR-reSGLD with SRVR-PG and GPOMDP using reSGLD optimization algorithm. Since the stochastic gradient MCMC for optimization requires temperature converges to zero to find the global optimal, we use a exponential decay: $\tau_k = \tau_{k-1}/1.001$, or $\tau_k = \tau k - 1/1.02$. Figure 3 shows the result on the comparision, and our proposed algorithm outperforms the original SRVR-PG in all tasks, especially in more complex environments (Hopper, HalfCheetah), which is consistent with the global convergence guarantee by SGMCMC. However, the reSGLD induces additional variance, which can be seen in Figure 3(c) and Figure 3(d). Furthermore, the reSGLD with GPOMDP estimator

**Algorithm 2:** Replica Exchange Stochastic Variance Reduced Gradient Langevin Dynamics

**input**: number of epoch $S$, epoch size $m$, batch size $N$,mini-batch size $B$, temperatures $\tau_1$ and $\tau_2$, learning rate $\eta$

**initialization**: $\tilde{\theta^0}^{(1)} = \theta^{0(1)}, \tilde{\theta^0}^{(2)} = \theta^{0(2)}$

**for** $s = 0, 1, ..., S-1$ **do**

$\theta_0^{(s+1)(c)} = \tilde{\theta^s}^{(c)}$

Sample N trajectories $\{\tau_j\}^{(c)}$ from $p(\cdot \mid \tilde{\theta^s}^{(c)})$

$\tilde{v}^{(c)} = \frac{1}{N}\sum_{i=1}^{N} g({\tau_i}^{(c)} \mid \tilde{\theta^s}^{(c)})$

**for** $t = 0, 1, ..., m-1$ **do**

Sample $B$ trajectories $\{\tau_j\}^{(c)}$ from $p(\cdot \mid \theta_t^{s+1(c)})$

$v_t^{s+1(c)} = \frac{1}{B}\sum_{i=1}^{B}\Big(g({\tau_i}^{(c)} \mid \theta_t^{s+1(c)}) - g_\omega({\tau_i}^{(c)} \mid \tilde{\theta^s}^{(c)})\Big)$

$v_t^{s+1(c)} = v_t^{s+1(c)} + \tilde{v}^{(c)}$

$\theta_{t+1}^{s+1(c)} = \theta_t^{s+1(c)} + \eta v_t^{s+1(c)} + \sqrt{2\eta\tau_c}\xi^{(c)}$

Generate a uniform random number $u \in [0, 1]$

$S_1 = \left(1 \wedge e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(v_t^{s+1(1)} - v_t^{s+1(2)}\right)}\right)$

**if** $u < S_1$ **then**

Swap $\theta_{t+1}^{s+1(1)}$ and $\theta_{t+1}^{s+1(2)}$

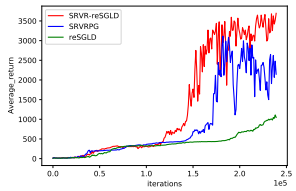
**end**

**end**

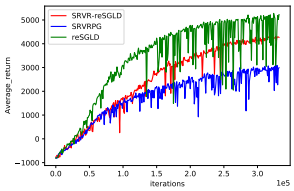
$\tilde{\theta^s}^{(c)} = \theta_m^{s\ (c)}$

**end**

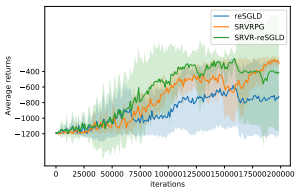
**return** $\theta_m^{s\ (\tau_1 \wedge \tau_2)}$



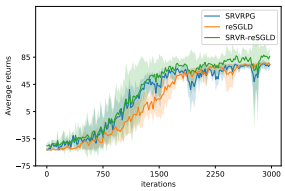

(a) Hopper-v1




(b) HalfCheetah-v1




(c) Pendulum-v0




(d) MountainCarContinuous-v0

Figure 3: Experiments on four RL tasks

| | Mountain Car | Pendulum | Half Cheetah(same with Hopper) |
|---|---|---|---|
| Adam $\alpha$ (SRVRPG) | $5 \cdot 10^{-2}$ | $10^{-3}$ | $10^{-3}$ |
| Adam $\alpha$ (GPOMDP) | $10^{-2}$ | $10^{-3}$ | $10^{-2}$ |
| Adam $\beta_1$ | 0.9 | 0.9 | 0.9 |
| Adam $\beta_2$ | 0.99 | 0.99 | 0.99 |
| Initial $\tau_1$ | 1 | 1 | 1 |
| Initial $\tau 2$ | 0.01 | 0.01 | 0.01 |
| Mini-batch size $B$ (SRVRPG) | 5 | 100 | 10 |
| Batch size (GPOMDP) | 10 | 250 | 100 |
| Max number of sub-iterations | 2 | 1 | 20 |
| Learning rate | 5e-3 | 2.5e-3 | 2.5e-3 |
| Baseline | No | No | No |
| Discount factor $\gamma$ | 0.999 | 0.995 | 0.99 |
| Total number of trajectories | 3000 | 200000 | 300000 |

Table 1: Parameter settings

is hard to achieve good performance. Unlike the training dataset in traditional deep learning task, we can only sample limited number of trajectories to estimate the gradient, sample-efficient estimator is essential in RL task. It is intractable to interact only with our global convergence guarantee.

## 6 Conclusion

We propose a novel global convergent policy gradient method called SRVR-reSGLD, which is built on a sample-efficient stochastic policy gradient estimator and SGMCMC. We shows that with global convergence guarantee in PG algorithm is effective both in theory and practice. Also, the SGMCMC can introduce high variance, the sample complexity is an issue we must address. Experiments on the classic control reinforcement learning benchmarks and Mujoco environment validate the advantage of our proposed algorithms.

## References


[1] Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International conference on machine learning*, pages 699–707. PMLR, 2016.

[2] Jonathan Baxter and Peter L Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[3] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[4] Tzuu-Shuh Chiang, Chii-Ruey Hwang, and Shuenn Jyi Sheu. Diffusion for global optimization in r^n. *SIAM Journal on Control and Optimization*, 25(3):737–753, 1987.

[5] Arnak Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. In *Conference on Learning Theory*, pages 678–689. PMLR, 2017.

[6] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):651–676, 2017.

[7] Wei Deng, Qi Feng, Liyao Gao, Faming Liang, and Guang Lin. Non-convex learning via replica exchange stochastic gradient mcmc. In *International Conference on Machine Learning*, pages 2474–2483. PMLR, 2020.

[8] Saul B Gelfand and Sanjoy K Mitter. Recursive stochastic algorithms for global optimization in $\mathbb{R}^d$. *SIAM Journal on Control and Optimization*, 29(5):999–1018, 1991.

[9] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[10] Chii-Ruey Hwang. Laplace's method revisited: weak convergence of probability measures. *The Annals of Probability*, pages 1177–1182, 1980.

[11] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[12] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[13] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[14] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703. PMLR, 2017.

[15] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323. PMLR, 2016.

[16] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[17] Umut Simsekli, Cagatay Yildiz, Than Huy Nguyen, Taylan Cemgil, and Gael Richard. Asynchronous stochastic quasi-newton mcmc for non-convex optimization. In *International Conference on Machine Learning*, pages 4674–4683. PMLR, 2018.

[18] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[19] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

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

[21] Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. *arXiv preprint arXiv:1909.08610*, 2019.