Knowledge-Based Systems 255 (2022) 109687

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Stochastic cubic-regularized policy gradient method

Pengfei Wang [a,d], Hongyu Wang [a,d], Nenggan Zheng [a,b,c,d,*]

[a] Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China
[b] CCAI by MOE and Zhejiang Provincial Government, Hangzhou 310027, China
[c] Zhejiang Lab, Hangzhou 311121, China
[d] College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 18 April 2022
Received in revised form 28 July 2022
Accepted 11 August 2022
Available online 19 August 2022

Keywords:
Reinforcement learning
Policy gradient
Stochastic optimization
Non-convex optimization
Second-order stationary point

ABSTRACT

Policy-based reinforcement learning methods have achieved great achievements in real-world decision-making problems. However, the theoretical understanding of policy-based methods is still limited. Specifically, existing works mainly focus on first-order stationary point policies (FOSPs); in some very special reinforcement learning settings (e.g., tabular case and function approximation with restricted parametric policy classes) some works consider globally optimal policy. It is well-known that FOSPs could be undesirable local optima or saddle points, and obtaining a global optimum is generally NP-hard. In this paper, we propose a policy gradient method that provably converges to second-order stationary point policies (SOSPs) for any differentiable policy classes. The proposed method is computationally efficient, and it judiciously uses cubic-regularized subroutines to escape saddle points while at the same time minimizing the Hessian-based computations. We prove that the method enjoys the sample complexity of $\widetilde{\mathcal{O}}(\epsilon^{-3.5})$, which improves upon the current optimal complexity $\widetilde{\mathcal{O}}\left(\epsilon^{-4.5}\right)$. Finally, experimental results are provided to demonstrate the effectiveness of the method.

1. Introduction

Over the past decade, we have witnessed the successful application of reinforcement learning (RL) in many scientific and engineering fields, such as robotics [1,2], gaming [3,4], healthcare [5,6], autonomous driving [7], traffic signal control [8], just to name a few. Unlike the more widely known supervised learning (SL) [9] and traditional optimization problems [10,11], RL is a goal-oriented sequential decision-making paradigm under uncertainly. Specifically, in RL an agent interacts with the environment and improves the decision policy by maximizing the cumulative reward. Reinforcement learning algorithms [12] can be roughly divided into two classes: (i) policy-based methods, i.e., directly parameterizing the policy, and optimizing it via maximizing the cumulative reward; and (ii) value-based methods characterized by finding the value function by solving the Bellman equation [13]. In this paper, we focus on policy-based methods due to their merits including implementation simplicity [14], scalability to large and even continuous spaces [15,16], and natural connection to stochastic optimization [17].

The study of policy gradient methods traces its roots to some early work in RL [18,19]. However, it was not until 1992 that a classical policy-gradient method, named REINFORCE, was proposed by [20]. In REINFORCE, the parameterized policy is updated along an approximate improvement direction estimated from data sampled by interacting with environment. It can be seen that the idea of REINFORCE is similar to SGD [21] in stochastic optimization. REINFORCE also suffers from the problem of high variance in its estimates like SGD. As such, reducing variance to improve sample efficiency has always been the core issue of policy-based methods. Classical approaches to reduce variance include using baseline [22,23] and designing actor–critic-type methods [24,25]. Moreover, we have seen a line of research in recent years that borrows the variance-reduced techniques from stochastic optimization to reduce variance in RL [26–30].

Despite its growing attention nowadays, developing theoretical analysis for policy-based methods is far from complete. A main handicap is the inherent non-convexity of the objective function, even for simple control problems [31]. In general, obtaining a global optimum for the non-convex function is NP-hard [32], although the global optimum convergence results of policy-based methods have been developed in several very special RL settings (e.g., tabular, LQR) [33–37]. Much of the existing work settles for a more modest goal, i.e., First-Order Stationary Point (FOSP) [14,38–40]. However, it is well known in non-convex optimization that FOSPs could be undesirable local optima or saddle points. From a non-convex optimization perspective, it is beneficial to develop methods that may escape from saddle points and find a Second-Order Stationary Point (SOSP). For RL, [41] first

[*] Corresponding author at: Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China.
E-mail address: zng@cs.zju.edu.cn (N. Zheng).

https://doi.org/10.1016/j.knosys.2022.109687
0950-7051/© 2022 Elsevier B.V. All rights reserved.

proposed a policy-based method named MRPG, which is a variant of REINFORCE with a periodically enlarged learning rate rule and provably converges to SOSPs with $\widetilde{\mathcal{O}}(\epsilon^{-9})$ sample complexity. Adapting the CNC technique [42] from stochastic optimization to RL, [43] proposed a new analysis of REINFORCE under more restrictive assumptions and proved that the algorithm also converges to SOSPs and yields an improved sample complexity of $\widetilde{\mathcal{O}}(\epsilon^{-4.5})$.

Given the close relationship between policy-based methods and stochastic optimization methods, we ask a natural question: what inspirations do recent algorithms that provably converges to SOSPs in non-convex optimization have on policy-based methods in RL? A classic algorithm for finding SOSPs in non-convex optimization is cubic-regularized (CR) Newton method [44], which naturally incorporates Hessian matrix (or second-order information) to escape saddle points. The CR Newton method requires the computation of the exact full gradient and Hessian, which is prohibitively expensive and is not available in the stochastic setting as is the case for RL. [45] proposed a stochastic variant of the CR Newton method, named SCR, which only requires stochastic gradient and Hessian–vector product evaluations. Moreover, SCR utilizes an algorithm from [46] to solve the CR (a second-order subroutine) more efficiently. But it still exhibits lower convergence rate in practice, since solving the CR requires the Hessian-based computations in each iteration. To reduce the Hessian-based computations, [47] developed a framework that carefully alternates between first-order and second-order subroutines using gradient and Hessian information, respectively. Inspired by this line of work, we develop a policy-based approach to find SOSPs, the core idea of which is to further decouple the first-order and the second-order subroutines ensuring that the most part of the optimization process is in the first-order subroutine and the second-order subroutine is invoked only when arriving near a FOSP. In a nutshell, our main contributions are summarized as follows:

1.1. Main contributions

A New Algorithm. We propose a Stochastic Cubic-Regularized Policy Gradient (SCR-PG) method in which the second-order subroutine is invoked only when the iterate arrives near a FOSP. If the iterate is a SOSP, SCR-PG terminates early and outputs the iterate, otherwise potentially escapes saddle points. Moreover, the new method is simple to use, since it only leverages stochastic gradient and Hessian–vector product evaluations which are both implementable in linear time with respect to the problem dimension, and only needs to approximately solve the cubic regularization rather than solve it exactly. The new method achieves the good properties of previous works [44,45,47] while avoiding their drawbacks.

Theoretical Guarantee. We provide a non-asymptotic analysis of SCR-PG's complexity with high probability. We prove that, under mild assumptions, to find an $\{\epsilon, \sqrt{\rho\epsilon}\}$-approximate SOSP where ϵ is a predefined precision accuracy and ρ is the Lipschitz constant of the Hessian matrix of the expected return (see Lemma 1), the sample complexity of SCR-PG is at most $\widetilde{\mathcal{O}}\left(\epsilon^{-3.5}\right)$. Note that this complexity is better than the complexity $\mathcal{O}\left(\epsilon^{-4}\right)$ of REINFORCE for finding FOSPs and the best-known complexity $\widetilde{\mathcal{O}}(\epsilon^{-4.5})$ for finding SOSPs proposed in [43].

1.2. Related work

Variance-Reduced Policy Gradient Methods. In stochastic optimization, variance-reduced techniques (e.g., SVRG, SARAH, SPIDER) have been actively studied. Policy-based methods equipped with various variance-reduced gradient estimators have been developed for RL [26–30]. All the work leverages the semi-stochastic gradient estimators to reduce variance. Under relatively strong assumptions on the RL problem and the parameterized policy, the sample complexity of variance-reduced policy gradient methods improves the vanilla result $\mathcal{O}(\epsilon^{-4})$ of REINFORCE by a factor $\mathcal{O}(\epsilon^{-1})$. However, we emphasize that all the work has been only proven to converge to FOSPs and generally requires strict learning rates to achieve this optimal complexity.

Global Optimum in Several Special RL Settings. The global optimum convergence results of policy-based methods have been developed in several very special RL settings [33–36]. Among them, [33] demonstrated the global convergence of policy-based method with the exact full gradient for the linear–quadratic regulator (LQR) problem. [34] analyzed three exact policy-based algorithms: projected PG (on the simplex), PG (with a softmax policy parameterization) and natural PG for the tabular case, and proved that their iteration complexities are $\mathcal{O}(\epsilon^{-2})$, $\mathcal{O}(\epsilon^{-2})$ and $\mathcal{O}(\epsilon^{-1})$, respectively. Later, [35] improved the results in [34], showing that exact PG for softmax tabular without regularization converges to the global optimum with a $\mathcal{O}(\epsilon^{-1})$ convergence rate, and achieves a linear convergence with entropy regularization. Also assuming access to exact full gradient, [36] demonstrated that entropy-regularized natural PG methods under softmax parameterization converges linearly - even quadratically around the optimal policy. It is worth mentioning that our analysis is significantly different from the analysis of the above work, since all of the above theoretical analyses are based on the exact full gradient while our analysis is built upon sample-based gradient estimates.

Relations with Second-Order Methods in Non-convex Optimization. As mentioned earlier, we cannot directly follow the second-order methods [44,45] since all the work heavily relies on the Hessian-based computations in each iteration. Besides the above work, variants of stochastic CR have been developed in [48,49] which combine the cubic regularization with variance-reduced techniques. In particular, [49] improves the sample complexity from $\widetilde{\mathcal{O}}(\epsilon^{-3.5})$ to $\widetilde{\mathcal{O}}(\epsilon^{-3})$, however the work also requires Hessian-based computations in each iteration, so it is slow in practice and is not suitable for RL especially with large or continuous spaces. To reduce the Hessian-based computations, [47] developed algorithms that alternate between first-order and second-order subroutines. In [47], the authors showed that a simple mix of the first-order and second-order methods (a first-order subroutine followed by a second-order subroutine in each iteration) can escape saddles points faster. This motivates us to further decouple the first-order and the second-order subroutines. Besides CR-based methods, some stochastic gradient algorithms with additive noise are able to escape saddle points and find SOSPs [50–52]. While these methods based on first-order information are appealing, they have strong dependence on the problem dimension and suffer from sample inefficiency.

1.3. Notation

Throughout this paper, we use $\|\cdot\|$ to denote the ℓ_2-norm of a vector or spectral norm of a matrix; we use $\lambda_{\max}(A)$ to denote the maximum eigenvalue of a matrix A; we use $\mathbf{I}_d$ to denote the $d \times d$ identity. We write $a_n = \mathcal{O}(b_n)$ for sequences a_n and b_n if there is a global constant C such that $a_n \leq Cb_n$, and $a_n = \widetilde{\mathcal{O}}(b_n)$ if C further hides a poly-logarithmic factor of the parameters. We write $a_n = \Omega(b_n)$ for sequences a_n and b_n if there is a global constant C such that $a_n \geq Cb_n$. For any symmetric matrices A and $B \in \mathbb{R}^{d\times d}$, we use $A \preceq B$ to denote that $B - A$ is positive semidefinite. Unless otherwise stated, $\mathbb{E}$ is used to denote the full expectation with respect to all random variables.

2. Preliminaries

Markov Decision Process (MDP). We consider a discounted Markov decision process defined by a tuple $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r, \rho\}$, where $\mathcal{S}$ is a state space; $\mathcal{A}$ is an action space; $\mathcal{P}(s'|s, a)$ is a Markovian transition model that determines the probability density from state s to s' given action $a \in \mathcal{A}$; $\gamma \in (0, 1)$ is the discount factor; $r(s, a)$ is the reward function for state s and action a, and $|r(s, a)| \le R$ for any state–action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$; and $\rho(s_0)$ is the distribution of the starting state s_0. A policy π at state s is a probability density $\pi(\cdot|s)$ over action space $\mathcal{A}$. In episodic tasks, the agent interacts with the environment following a policy π, generating a trajectory τ which is a sequence of states and actions, denoted by $(s_0, a_0, s_1, a_1, \dots, s_{H-1}, a_{H-1})$, where $s_0 \sim \rho(s_0)$ and H is called the trajectory horizon or episode length. Given a trajectory τ, a cumulative discounted reward can be expressed as $R(\tau) = \sum_{j=0}^{H-1} \gamma^j r(s_j, a_j)$.

We consider the function approximation setting and assume that $\pi(\cdot|s)$ is parameterized by an unknown parameter $\theta \in \mathbb{R}^d$, denoted by π_θ. Given an initial state distribution $\rho(s_0)$, the distribution density $p(\tau|\theta)$ of trajectory τ induced by π_θ is presented as $p(\tau|\theta) = \rho(s_0) \prod_{j=0}^{H-1} \pi_\theta(a_j|s_j) \mathcal{P}(s_{j+1}|s_j, a_j)$. The goal of an agent is to find a parameterized policy π_θ so that maximizes the expected return:

$$\max_{\theta \in \mathbb{R}^d} \{J(\theta) = \mathbb{E}_{\tau \sim p(\cdot|\theta)} [R(\tau)]\}. \tag{1}$$

A policy parameter $\theta^* \in \arg\max_{\theta \in \mathbb{R}^d} J(\theta)$ is said to be optimal, and the corresponding optimal expected return is denoted as $J^* = J(\theta^*)$.

REINFORCE. To maximize the expected return $J(\theta)$, the parameter θ can be updated as $\theta_{t+1} = \theta_t + \eta \nabla_\theta J(\theta_t)$, where $\eta > 0$ is the learning rate and the exact full gradient $\nabla_\theta J(\theta)$ is derived as

$$\begin{aligned} \nabla_\theta J(\theta) &= \int_\tau \nabla_\theta p(\tau|\theta) R(\tau) d\tau = \int_\tau \nabla_\theta \log p(\tau|\theta) p(\tau|\theta) R(\tau) d\tau \\ &= \mathbb{E}_{\tau \sim p(\cdot|\theta)} [\nabla_\theta \log p(\tau|\theta) R(\tau)]. \end{aligned} \tag{2}$$

Since the distribution $p(\tau|\theta)$ is unknown, we cannot calculate $\nabla_\theta J(\theta)$ in practice. Similar to SGD, REINFORCE construct a stochastic gradient estimator using a trajectory τ as

$$g(\tau|\theta) = \sum_{j=0}^{H-1} \nabla_\theta \log \pi_\theta(a_j|s_j) R(\tau), \tag{3}$$

where $g(\tau|\theta)$ is an unbiased estimator of the exact full gradient $\nabla_\theta J(\theta)$ i.e., $\mathbb{E}_{\tau \sim p(\cdot|\theta)} [g(\tau|\theta)] = \nabla_\theta J(\theta)$, and the unbiased gradient estimator $g(\tau|\theta)$ is often called the REINFORCE gradient estimator.

Stochastic Hessian Estimator. The Hessian matrix of the expected return $J(\theta)$ is derived as

$$\begin{aligned} \nabla_\theta^2 J(\theta) &= \int_\tau \left(\nabla_\theta^2 \log p(\tau|\theta) + \nabla_\theta \log p(\tau|\theta) \nabla_\theta \log p(\tau|\theta)^{\mathrm{T}}\right) \\ &\quad \times p(\tau|\theta) R(\tau) d\tau \\ &= \mathbb{E}_{\tau \sim p(\cdot|\theta)} \left[\left(\nabla_\theta^2 \log p(\tau|\theta) + \nabla \log p(\tau|\theta) \right.\right. \\ &\quad \left.\left. \times \ \nabla_\theta \log p(\tau|\theta)^{\mathrm{T}}\right) R(\tau)\right]. \end{aligned} \tag{4}$$

Similarly, based on a trajectory τ an unbiased stochastic Hessian estimator of the exact full Hessian matrix $\nabla_\theta^2 J(\theta)$ can be constructed as

$$H(\tau|\theta) = \left(\nabla_\theta^2 \Phi(\theta, \tau) + \nabla_\theta \Phi(\theta, \tau) \nabla_\theta \Phi(\theta, \tau)^{\mathrm{T}}\right) R(\tau), \tag{5}$$

where $\nabla \Phi(\theta, \tau) = \sum_{j=0}^{H-1} \nabla \log \pi_\theta(a_j|s_j)$, $\nabla^2 \Phi(\theta, \tau) = \sum_{j=0}^{H-1} \nabla^2 \log \pi_\theta(a_j|s_j)$, and $\mathbb{E}[\|H(\tau|\theta)\|] = \nabla_\theta^2 J(\theta)$. Instead of computing the stochastic Hessian estimator $H(\tau|\theta)$ exactly, we access the function $H(\tau|\theta)[\cdot] : \mathbb{R}^d \to \mathbb{R}^d$, where $H(\tau|\theta)[\mathbf{v}] = \left(\nabla_\theta^2 \Phi(\theta, \tau) \cdot \mathbf{v} + \nabla_\theta \Phi(\theta, \tau) \nabla_\theta \Phi(\theta, \tau)^{\mathrm{T}} \cdot \mathbf{v}\right) R(\tau)$ for any $\mathbf{v} \in \mathbb{R}^d$, and the Hessian–vector product $\nabla_\theta^2 \Phi(\theta, \tau) \cdot \mathbf{v}$ can be calculated with automatic differentiation [53].

$(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSPs. In general, the expected return $J(\theta)$ is non-convex. Compared to previous policy-based methods that are satisfied with ϵ-approximate FOSPs, i.e., $\|\nabla_\theta J(\theta)\| \le \epsilon$, the aim of the paper is to design policy-based method that is able to find $\left(\epsilon, \sqrt{\rho\epsilon}\right)$-approximate SOSPs:

$$\|\nabla_\theta J(\theta)\| \le \epsilon \quad \text{and} \quad \lambda_{\max}(\nabla_\theta^2 J(\theta)) \le \sqrt{\rho\epsilon}, \tag{6}$$

in which ρ is the Lipschitz constant of the Hessian matrix $\nabla_\theta^2 J(\theta)$ (see Lemma 1).

Cubic-Regularized (Newton) Method. The most classic approach to find the SOSPs is cubic-regularized (Newton) method, which was originally proposed by Nesterov and Polyak [44]. Specifically, at the tth iteration, cubic-regularized method maximizes a cubic-regularized second-order Tylor expansion at the current iterate θ_t. The update rule can be written as follows:

$$\Delta_t = \arg\max_{\Delta \in \mathbb{R}^d} \Delta^{\mathrm{T}} \nabla J(\theta_t) + \frac{1}{2} \Delta^{\mathrm{T}} \nabla^2 J(\theta_t) \Delta - \frac{\rho}{6} \|\Delta\|^3, \tag{7}$$

$$\theta_{t+1} = \theta_t + \Delta_t. \tag{8}$$

It has been shown that to find an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSPs, cubic-regularized method requires $\mathcal{O}(\epsilon^{-1.5})$ iterations [44].

3. Methodology

In this section, we propose SCR-PG to maximize the expected return $J(\theta)$. The pseudocode is given in Algorithm 1. In the initialization, we set the initial policy parameter to be θ^0. SCR-PG consists of S epochs, each epoch is composed of two subroutines: one is a first-order subroutine (Line 2–10) including an inner loop consisting of stochastic gradient ascent updates; the other is a second-order subroutine[1] (Line 11–20) which is invoked only when the "if statement" in Line 12 holds.

Taking the sth epoch as an example, the policy parameter θ_0^{s+1} is set to the same value as the policy parameter $\widetilde{\theta}^s$. The method samples B_N trajectories $\{\tau_i\}_{i=1}^{B_N}$ induced by the policy parameter θ_0^{s+1}, and approximates the exact policy gradient over these trajectories, i.e., $v_0^{s+1} = \frac{1}{B_N} \sum_{i=1}^{B_N} g(\tau_i|\theta_0^{s+1})$. The policy parameter θ_1^{s+1} is updated based on v_0^{s+1}, i.e., $\theta_1^{s+1} = \theta_0^{s+1} + \eta v_0^{s+1}$.

Then, SCR-PG enters the inner loop consisting of stochastic gradient ascent steps, which aims to progressively reduce the high variance in its estimates. In Algorithm 1, we employ a SARAH-type variance-reduced technique [54], which can be replaced by SVRG-type gradient descent technique [55]. More specifically, within the tth inner loop at the sth epoch, the method samples B_M trajectories $\{\tau_i\}_{i=1}^{B_M}$ induced by the current policy θ_t^{s+1}. The method constructs a semi-stochastic recursive gradient estimator, i.e.,

$$v_t^{s+1} = v_{t-1}^{s+1} + \frac{1}{B_M} \sum_{i=1}^{B_M} \left(g(\tau_i|\theta_t^{s+1}) - g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right), \tag{9}$$

where $g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) = \omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) g(\tau_i|\theta_{t-1}^{s+1})$, and $\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) = \prod_{j=0}^{H-1} \pi_{\theta_{t-1}^{s+1}}(a_j|s_j) / \pi_{\theta_t^{s+1}}(a_j|s_j)$. The method updates the policy parameter θ_{t+1}^{s+1} along the direction of v_t^{s+1}, i.e., $\theta_{t+1}^{s+1} = \theta_t^{s+1} + \eta v_t^{s+1}$. After m iterations, SCR-PG obtains two reference policy parameters: $\widetilde{\theta}^{s+1} = \theta_m^{s+1}$ and $\widehat{\theta}^{s+1} = \theta_t^{s+1}$ with t chosen uniformly at random from $\{0, 1, \dots, m-1\}$.

[1] Although we call it a second-order subroutine, in fact, we do not calculate and store the exact full Hessian matrix in this subroutine, and only stochastic gradient and stochastic Hessian–vector product evaluations are required.

Algorithm 1 Stochastic Cubic-Regularized Policy Gradient Method

```
Input: An initial $\widetilde{\theta}^0 = \widehat{\theta}^0 = \theta^0$ and $B_N, B_M, B_H, m, \eta, \epsilon$.
for $s = 0$ to $S - 1$ do
    $\theta_0^{s+1} = \widetilde{\theta}^s$
    $v_0^{s+1} = \frac{1}{B_N}\sum_{i=1}^{B_N} g(\tau_i|\theta_0^{s+1})$
    $\theta_1^{s+1} = \theta_0^{s+1} + \eta v_0^{s+1}$
    for $t = 1$ to $m - 1$ do
        $v_t^{s+1} = v_{t-1}^{s+1} + \frac{1}{B_M}\sum_{i=1}^{B_M}(g(\tau_i|\theta_t^{s+1}) - g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}))$
        $\theta_{t+1}^{s+1} = \theta_t^{s+1} + \eta v_t^{s+1}$
    end for
    $\widetilde{\theta}^{s+1} = \theta_m^{s+1}$
    $\widehat{\theta}^{s+1} = \theta_t^{s+1}$ with $t$ chosen uniformly at random from $\{0, 1, \cdots, m-1\}$
    $g^{s+1} = \frac{1}{B_N}\sum_{i=1}^{B_N} g(\tau_i|\widehat{\theta}^{s+1})$
    if $\left\|g^{s+1}\right\| \leq \epsilon$ then
        $H^{s+1}[\cdot] = \frac{1}{B_H}\sum_{j=1}^{B_H} H(\tau_j|\widehat{\theta}^{s+1})[\cdot]$
        $\Delta^{s+1}, \Delta_m^{s+1} \leftarrow$ Cubic-Subsolver $\left(g^{s+1}, H^{s+1}[\cdot], \epsilon\right)$
        if $\Delta_m^{s+1} < \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$ then
            return $\theta^* \leftarrow \widehat{\theta}^{s+1}$
        else
            $\widetilde{\theta}^{s+1} \leftarrow \widehat{\theta}^{s+1} + \Delta^{s+1}$
        end if
    end if
end for
Output: $\theta^*$ if the early termination condition was reached, otherwise $\widetilde{\theta}^S$.
```

Algorithm 2 Cubic-Subsolver $(g, H[\cdot], \epsilon)$

```
Input: $g$, $H[\cdot]$, $\epsilon$
1:  if $\|g\| \geq \frac{L^2}{\rho}$ then
2:      $R_c \leftarrow -\frac{g^{\mathrm{T}}H[g]}{\rho\|g\|^2} + \sqrt{\left(\frac{g^{\mathrm{T}}H[g]}{\rho\|g\|^2}\right)^2 + \frac{2\|g\|}{\rho}}$
3:      $\Delta \leftarrow \frac{g}{\|g\|}R_c$
4:  else
5:      $\Delta \leftarrow 0, \sigma \leftarrow c'\frac{\sqrt{\rho\epsilon}}{L}, \eta \leftarrow \frac{1}{20L}$
6:      $\widetilde{g} \leftarrow g + \sigma\zeta$ for $\zeta \sim \mathrm{Unif}(\mathbb{S}^{d-1})$
7:      for $t = 1, \cdots, \mathcal{T}(\epsilon)$ do
8:          $\Delta \leftarrow \Delta + \eta\left(\widetilde{g} + H[\Delta] - \frac{\rho}{2}\|\Delta\|\,\Delta\right)$
9:      end for
10: end if
11: $\Delta_m \leftarrow g^{\mathrm{T}}\Delta + \frac{1}{2}\Delta^{\mathrm{T}}H[\Delta] - \frac{\rho}{6}\|\Delta\|^3$
Output: $\Delta$, $\Delta_m$
```

Afterwards, SCR-PG computes the policy gradient estimator over B_N trajectories $\{\tau_i\}_{i=1}^{B_N}$ induced by the reference policy parameter $\widehat{\theta}^{s+1}$, i.e., $g^{s+1} = \frac{1}{B_N}\sum_{i=1}^{B_N} g(\tau_i|\widehat{\theta}^{s+1})$. When $\left\|g^{s+1}\right\| > \epsilon$, SCR-PG continues to the first-order subroutine to search for FOSPs. When $\left\|g^{s+1}\right\| \leq \epsilon$, SCR-PG enters the second-order subroutine. In this case, SCR-PG computes the stochastic Hessian estimator $H^{s+1}[\cdot] = \frac{1}{B_H}\sum_{j=1}^{B_H} H(\tau_j|\widehat{\theta}^{s+1})[\cdot]$ over B_H trajectories $\{\tau_j\}_{j=1}^{B_H}$. With g^{s+1} and $H^{s+1}[\cdot]$, SCR-PG approximately performs a cubic-regularized update. A subroutine Cubic-Subsolver (see Algorithm 2) is used to approximately solve the cubic regularization subproblem (7). Cubic-Subsolver receives g^{s+1}, $H^{s+1}[\cdot]$, ϵ as input and outputs Δ^{s+1}, Δ_m^{s+1}, i.e.,

$$\Delta^{s+1}, \Delta_m^{s+1} \leftarrow \text{Cubic-Subsolver}(g^{s+1}, H^{s+1}[\cdot], \epsilon). \tag{10}$$

Denote the cubic regularization subproblem at the sth epoch as follows:

$$m^{s+1}(\widehat{\theta}^{s+1} + \Delta) = J(\widehat{\theta}^{s+1}) + \Delta^{\mathrm{T}}g^{s+1} + \frac{1}{2}\Delta^{\mathrm{T}}H^{s+1}[\Delta] - \frac{\rho}{6}\|\Delta\|^3. \tag{11}$$

In fact, Cubic-Subsolver returns the approximate maximum Δ^{s+1} of the cubic regularization as well as the cubic regularization subproblem change $\Delta_m^{s+1} = m^{s+1}(\widehat{\theta}^{s+1} + \Delta^{s+1}) - m^{s+1}(\widehat{\theta}^{s+1})$. When $\Delta_m^{s+1} < \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$, SCR-PG terminates early and returns $\widehat{\theta}^{s+1}$ as its output; otherwise the method updates the policy parameter $\widetilde{\theta}^{s+1} = \widehat{\theta}^{s+1} + \Delta^{s+1}$ and continues the loop.

Remark 1. We now make some remarks about the SCR-PG algorithm.

(i) $\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) = \prod_{j=0}^{H-1}\pi_{\theta_{t-1}^{s+1}}(a_j|s_j)/\pi_{\theta_t^{s+1}}(a_j|s_j)$ is an importance sampling weight used to compensate for the distribution shift from policy $\pi_{\theta_t^{s+1}}$ to $\pi_{\theta_{t-1}^{s+1}}$. It is easy to verify that $\mathbb{E}_{\tau_i \sim p(\cdot|\theta_t^{s+1})}\left[g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right] = \nabla J(\theta_{t-1}^{s+1})$.

(ii) Algorithm 2 is a simple variant of standard gradient ascent. The main differences between them are two-folds: (1) lines 1-4: when g is large, the iterate only moves one step along the direction of g, which already guarantees sufficient ascent; (2) line 6: a small perturbation is added to g to avoid a "hard" case for the cubic regularization subproblem.

(iii) Δ^{s+1} may not be an exact solver of $m^{s+1}(\Delta)$. We tolerate a certain amount of suboptimality. In Lemma 3, we show that after $\mathcal{T}(\epsilon) = \widetilde{\mathcal{O}}\left(L/\sqrt{\rho\epsilon}\right)$ iterations, *Cubic-Subsolver* yields a good enough approximate maximum.

(iv) If $\Delta_m^{s+1} < \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$, the reference policy parameter $\widehat{\theta}^{s+1}$ is already an ϵ-approximate SOSP. If not, the reference policy parameter $\widehat{\theta}^{s+1}$ may be a saddle point; in this case, the updated reference policy parameter $\widetilde{\theta}^{s+1}$ potentially escape saddle points, i.e., $J(\widetilde{\theta}^{s+1}) - J(\widetilde{\theta}^s) \geq \Omega(\epsilon^{1.5})$ (see Section 5).

(v) In fact, SCR-PG first searches for FOSPs, being in the first-order subroutine. When SCR-PG arrives near a FOSP, it enters the second-order subroutine. If $\Delta_m^{s+1} < \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$, then $\widehat{\theta}^{s+1}$ is a SOSP and the algorithm ends. Otherwise, the objective function value increases by $\Omega(\epsilon^{1.5})$, and SCR-PG continues to search for FOSPs in a new region, repeating the above steps. We conclude that the most part of the optimization process is in the first-order subroutine. The cubic regularization subproblem is relatively expensive compared with the stochastic gradient ascent updates. Therefore, in contrast to the stochastic CR methods in non-convex optimization, e.g., [45,48,49] that the cubic regularization subproblem appears in each iteration, SCR-PG decouples the first-order and second-order subroutines, and reduces the number of oracles to the second-order subroutine, while as the same time ensures converging to SOSPs.

4. Theoretical analysis

In this section, we first introduce the fundamental assumptions used in our analysis, then present our main theoretical results.

Assumption 1. The parameterized policy π_θ is differentiable with respect to θ. Moreover, there exist constants $G > 0$, $M > 0$ such that, for any state–action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, we have

$$\|\nabla_\theta \log \pi_\theta(a|s)\| \leq G, \quad \left\|\nabla_\theta^2 \log \pi_\theta(a|s)\right\| \leq M. \tag{12}$$

Assumption 2. There exists a constant $F > 0$ such that, for any state–action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ and parameterized policies $\theta^1, \theta^2 \in \mathbb{R}^d$, we have

$$\left\|\nabla_\theta^2 \log \pi_{\theta^1}(a|s) - \nabla_\theta^2 \log \pi_{\theta^2}(a|s)\right\| \leq F\left\|\theta^1 - \theta^2\right\|. \tag{13}$$

Note that Assumptions 1–2 are widely used to analyze the convergence rate of policy gradient methods [38,56–58]. These assumptions can be satisfied easily by commonly used policies such as the Gaussian policy [59] and the Gibbs policy [14]. Taking the Gaussian policy in continuous spaces as an example, the parameterized policy $\pi_\theta(a|s) = \mathcal{N}(\phi(s)^{\mathrm{T}}\theta, \sigma^2)$. Then $\nabla_\theta \log \pi_\theta(a|s) = [a - \phi(s)^{\mathrm{T}}\theta]\phi(s)/\sigma^2$ and $\nabla^2_\theta \log \pi_\theta(a|s) = \phi(s)\phi(s)^{\mathrm{T}}/\sigma^2$. We see that conditions (12)–(13) holds, as long as the feature map $\phi(s)$ is bounded, and the parameter θ and the actions $a \in \mathcal{A}$ are in some bounded sets.

Assumption 3. There exists a constant $W > 0$ such that, for any parameterized policies $\theta^1, \theta^2 \in \mathbb{R}^d$, and trajectory $\tau \sim p(\cdot|\theta^2)$, we have

$$\mathrm{Var}\left(\omega(\tau|\theta^1, \theta^2)\right) \le W. \tag{14}$$

Assumption 3 ensures that the variance of the importance weight is bounded, which has been widely used in the study of variance-reduced policy gradient methods [26,28,30]. It is noted in [60] that for two Gaussian distribution $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$, the variance of the importance weights from the latter to the former is bounded if $\sigma_2 > \sqrt{2}/2\sigma_1$. Thus, it trivially holds for the fixed-variance Gaussian policies with $\sigma > \sqrt{2}/2\sigma$. In [27, Lemma 6.1], the authors showed that under Assumptions 1 and 3, it holds that $\mathrm{Var}(\omega(\tau|\theta^1, \theta^2)) \le C_W \left\|\theta^1 - \theta^2\right\|^2$, where $C_W = H(2HG^2 + M)(W + 1)$.

Now, we state the main convergence result of Algorithm 1 as follows:

Theorem 1. *Suppose that Assumptions 1, 2 and 3 hold. There exists a constant $c > 0$, such that under $B_N = \frac{8}{3}\max\left(\frac{2\kappa_f}{c\epsilon}, \frac{\kappa_f^2}{c^2\epsilon^2}\right)\log\frac{8\sqrt{\rho}d(J^*-J(\theta^0))}{\delta c\sqrt{\epsilon^3}}$, $B_H = \frac{8}{3}\max\left(\frac{2\kappa_g}{c\sqrt{\epsilon\rho}}, \frac{\kappa_g^2}{c^2\epsilon\rho}\right)\log\frac{8\sqrt{\rho}d(J^*-J(\theta^0))}{\delta c\sqrt{\epsilon^3}}$, $B_M = \frac{\kappa_f^2 C_W+\kappa_g^2}{\epsilon^{1/2}}$, $m = \frac{L^2}{\epsilon^{1/2}}$, and $\eta = \frac{1}{2L}$, SCR-PG will output an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSP with probability at least $1 - \delta$ within $S = \left\lceil\frac{\sqrt{\rho}(J^*-J(\theta^0))}{c\sqrt{\epsilon^3}}\right\rceil$.*

Corollary 1. *Under the conditions in Theorem 1, SCR-PG will return an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSP with probability at least $1-\delta$ with at most $\widetilde{\mathcal{O}}\left(\epsilon^{-3.5}\right)$ sample complexity, which improves upon the best-known complexity $\widetilde{\mathcal{O}}(\epsilon^{-4.5})$ for finding SOSPs proposed in [43].*

Remark 2. SCR-PG minimizes the Hessian-based computations. It can be seen that the total number of oracles to the cubic regularization is $\mathcal{O}(\epsilon^{-1.5})$ only when the second-order subroutine is invoked in each iteration. However, in fact, it is only invoked if the iterate is close to a FOSP. By contrast, in [47], their proposed framework alternates between first-order and second-order subroutines, in which the number of oracles to the cubic regularization is at least $\mathcal{O}(\epsilon^{-1.5})$. Our SCR-PG algorithm is the fastest to escape saddle points by using cheap gradient information most of the time and reducing the use of relatively expensive second-order subroutine.

5. Sketch of proofs

In this section, we outline the proof of Theorem 1. Based on the Assumptions 1 and 2, we have the following lemma.

Lemma 1. *Suppose that Assumptions 1 and 2 hold, we have: (1) The gradient $\nabla_\theta J(\theta)$ of the expected return $J(\theta)$ is Lipschitz continuous with Lipschitz constant L; (2) The Hessian matrix $\nabla^2_\theta J(\theta)$ of the expected return $J(\theta)$ is Lipschitz continuous with Lipschitz constant ρ; (3) The stochastic gradient estimator $g(\tau|\theta)$ is upper bounded by constant κ_f and is Lipschitz continuous with Lipschitz constant κ_g.*

Then, we use the matrix Bernstein inequality [61, Theorem 6.1.1] to analyze the concentration bounds of stochastic gradients and Hessian matrices, and give conditions (cf. Lemma 2) to control the concentration error.

Lemma 2. *Suppose that Assumptions 1 and 2 hold. For any fixed small constants $c_1, c_2 > 0$, we can pick mini-batch sizes $B_N = \frac{8}{3}\max\left(\frac{2\kappa_f}{c_1\epsilon}, \frac{\kappa_f^2}{c_1^2\epsilon^2}\right)\log\frac{2d}{\delta'}$, $B_H = \frac{8}{3}\max\left(\frac{2\kappa_g}{c_2\sqrt{\epsilon\rho}}, \frac{\kappa_g^2}{c_2^2\epsilon\rho}\right)\log\frac{2d}{\delta'}$, s.t., with probability $1 - \delta'$*

$$\left\|\frac{1}{B_N}\sum_{i=1}^{B_N} g(\tau_i|\theta) - \nabla_\theta J(\theta)\right\| \le c_1\epsilon, \quad \forall \tau_i \sim p(\cdot|\theta); \tag{15}$$

$$\left\|\frac{1}{B_H}\sum_{i=1}^{B_H} H(\tau_i|\theta) - \nabla^2_\theta J(\theta)\right\| \le c_2\sqrt{\rho\epsilon}, \quad \forall \tau_i \sim p(\cdot|\theta). \tag{16}$$

Next, we give a Condition 1 and we show in Lemma 3 that Algorithm 2 can satisfy the condition with high probability within $\widetilde{\mathcal{O}}(L/\sqrt{\rho\epsilon})$ iterations.

Condition 1. *For any fixed small constants $c_3 > 0$, $c_4 > 0$, Cubic-Subsolver(g, $H[\cdot]$, ϵ) terminates within $\mathcal{T}(\epsilon)$ gradient iterations (which may depend on c_3, c_4), and returns a Δ satisfying at least one of the following:*

1. $\min\{\widetilde{m}(\Delta), J(\theta + \Delta) - J(\theta)\} \ge \Omega(\sqrt{\epsilon^3/\rho})$; (17)
2. $\|\Delta\| \le \|\Delta_*\| + c_3\sqrt{\frac{\epsilon}{\rho}}$,

and if $\|\Delta_*\| \ge \frac{1}{2}\sqrt{\frac{\epsilon}{\rho}}$, *then* $\widetilde{m}(\Delta) \ge \widetilde{m}(\Delta_*) - \frac{c_4}{12}\rho\left\|\Delta_*\right\|^3$, (18)

where Δ_* is an exact solution, i.e., $\Delta_* = \arg\max_{\Delta\in\mathbb{R}^d}\widetilde{m}(\Delta)$, and $\widetilde{m}(\Delta) = \Delta^{\mathrm{T}}g + \frac{1}{2}\Delta^{\mathrm{T}}H[\Delta] - \frac{\rho}{6}\left\|\Delta\right\|^3$.

The key idea in the proof of Theorem 1 is that at each iteration the improvement of the objective function is guaranteed to be greater than $\Omega(\epsilon^{1.5})$, otherwise Algorithm 1 outputs an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSP. Note that case 1 (i.e., (17)) in Condition 1 always satisfies this improvement, so we only consider the case 2 (i.e., (18)) in the following analysis.

Lemma 3 ([45, Lemma 2])**.** *There exists a constant $c' > 0$, such that under Assumptions 1 and 2 and the same choice of parameters B_N, B_H as in Lemma 2, Algorithm 2 satisfies Condition 1 with probability at least $1 - \delta'$ within $\mathcal{T}(\epsilon) \le \widetilde{\mathcal{O}}(L/\sqrt{\epsilon\cdot\rho})$.*

Lemma 4. *Suppose that Assumptions 1, 2 and 3 hold. Under the setting of Lemma 2 with a fixed constant c_1. Set $B_M = \frac{\kappa_f^2 C_W+\kappa_g^2}{\epsilon^{1/2}}$, $m = \frac{L^2}{\epsilon^{1/2}}$, and $\eta = \frac{1}{2L}$, we have*

$$\mathbb{E}\left[J(\widehat{\theta}^{s+1})\right] \ge \mathbb{E}\left[J(\widetilde{\theta}^{s})\right]; \tag{19}$$

$$\mathbb{E}\left[\left\|\nabla_\theta J(\widehat{\theta}^{s+1})\right\|^2\right] \le 4\epsilon^{1/2}L^{-1}\mathbb{E}\left[J(\widehat{\theta}^{s+1}) - J(\widetilde{\theta}^{s})\right] + (c_1\epsilon)^2. \tag{20}$$

Lemma 5. *Suppose that Assumptions 1 and 2 hold. Under the setting of Lemma 2 with any fixed small constants c_1, c_2, we have*

$$J(\widetilde{\theta}^{s+1}) \ge J(\widehat{\theta}^{s+1}); \tag{21}$$

if $\lambda_{\max}\left(\nabla^2_\theta J(\widehat{\theta}^{s+1})\right) \ge \sqrt{\rho\epsilon}$, *then* $J(\widetilde{\theta}^{s+1})$

$$\ge J(\widehat{\theta}^{s+1}) + \frac{2}{3}(1 - c_5)\sqrt{\frac{\epsilon^3}{\rho}}, \tag{22}$$

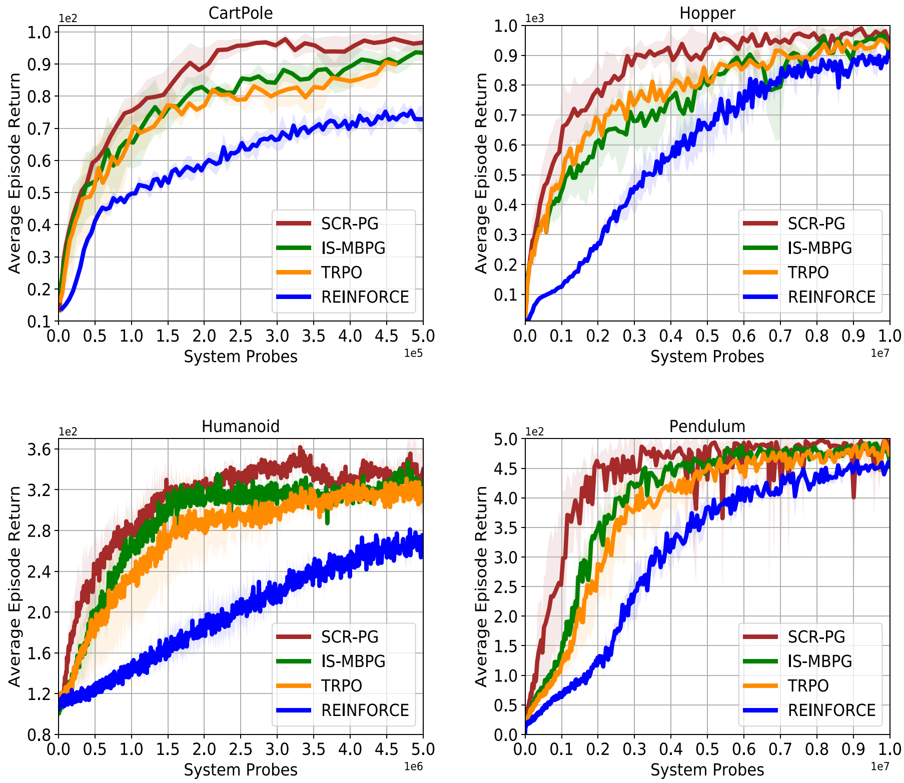


Fig. 1. Performance of SCR-PG and compared algorithms at four environments.

where c_5 is a small enough constant (which depends only on c_1, c_2, c_3, c_4).

The key idea of the proof of Theorem 1 is to show that the objective function value has a monotonic improvement in each epoch, otherwise SCR-PG terminates early and outputs an approximate SOSP. Based on the magnitude of the gradient and the curvature of the Hessian, we split our analysis into three cases: $\left\|g^{s+1}\right\| > \epsilon$; $\left\|g^{s+1}\right\| \leq \epsilon$ and $\nabla_\theta^2 J(\widetilde{\theta}^{s+1}) \leq \sqrt{\rho\epsilon}\mathbf{I}$; and $\left\|g^{s+1}\right\| \leq \epsilon$ and $\lambda_{\max}\left(\nabla_\theta^2 J(\widetilde{\theta}^{s+1})\right) \geq \sqrt{\rho\epsilon}$. In **Case I**, it can be seen that $\widetilde{\theta}^{s+1}$ is in a non-stationary region. We present a lemma (i.e., Lemma 4) that quantifies the behavior of the objective function in the first-order subroutine. From the lemma, we see that the first-order subroutine ensures that the objective function value increases by $\Omega(\epsilon^{1.5})$. In **Case II**, we see that $\widetilde{\theta}^{s+1}$ is near a saddle point. We present a lemma (i.e., Lemma 5). It is shown that in this case $\Delta_m^{s+1} \geq \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$ and the objective function value also increases by at least $\Omega(\epsilon^{1.5})$. In **Case III**, $\widetilde{\theta}^{s+1}$ is an approximate SOSP. In this case, if $\Delta_m^{s+1} < \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$, SCR-PG outputs the approximate SOSP. If not, Lemma 5 indicates that the objective function value increases by $\Omega(\epsilon^{1.5})$. Overall, SCR-PG outputs an approximate SOSP, otherwise $\mathbb{E}\left[J(\widetilde{\theta}^{s+1}) - J(\widetilde{\theta}^{s})\right] \geq \Omega(\epsilon^{1.5})$.

6. Experiments

To validate the effectiveness of SCR-PG, we test it on several benchmark RL tasks including CartPole, Hopper, Humanoid and Pendulum. The first one is a classic discrete control task [62], and the later three tasks are widely used continuous RL tasks, which are from Mujoco environments [63]. For baselines, we compare SCR-PG with three popular and successful algorithms: classical policy-gradient method REINFORCE [20]; TRPO [16] which also leverages the second-order information with strong empirical performance; and sample-efficient variance-reduced method IS-MBPG [30].

In this experiment, we use *categorical* policy for CartPole and *Gaussian* policy for the other environments where the policy is approximated by neural networks. For a fair comparison, all algorithms are initialized with the same random policy and are run 10 times to ease the impact of randomness. We provide average episode return and variance interval for each of them. We use system probes (i.e., the number of state transitions) to measure the sample complexity. For SCR-PG, we choose parameters as in Theorem 1. For the other algorithms, we use the parameter settings from their original papers. All parameters and neural network architectures are presented in Appendix F.

First, we compare the convergence rate of SCR-PG with the other algorithms. The learning curves for SCR-PG and the other algorithms are provided in Fig. 1. It can be seen that our SCR-PG method exhibits better performance than the other methods in all tasks. The average episode return of SCR-PG grows rapidly particularly in Hopper, Humanoid and Pendulum. We note that in Theorem 1 a constant learning rate can guarantee the convergence of SCR-PG, speeding up the convergence rate, whereas the other methods generally require strict learning rates to guarantee convergence. For instance, the learning rate of IS-MBPG is required to be $\mathcal{O}(1/t^{1/3})$ (decay to zero).

Next, we examine the effect of the second-order subroutine of SCR-PG on its performance. Experimental results are shown in Fig. 2. Here, SCR-PG* represents Algorithm 1 removing the second-order subroutine (i.e., Line 11–20). Note that SCR-PG provably converges to SOSPs, while SCR-PG* can only converge to FOSPs. We see that, SCR-PG and SCR-PG* have similar performance at the beginning of training process, but SCR-PG outperforms SCR-PG* at the middle and late stage of training. This is probably because SOSPs may be global or near-global solutions for a large class of learning problems but FOSPs may be undesirable saddle points.

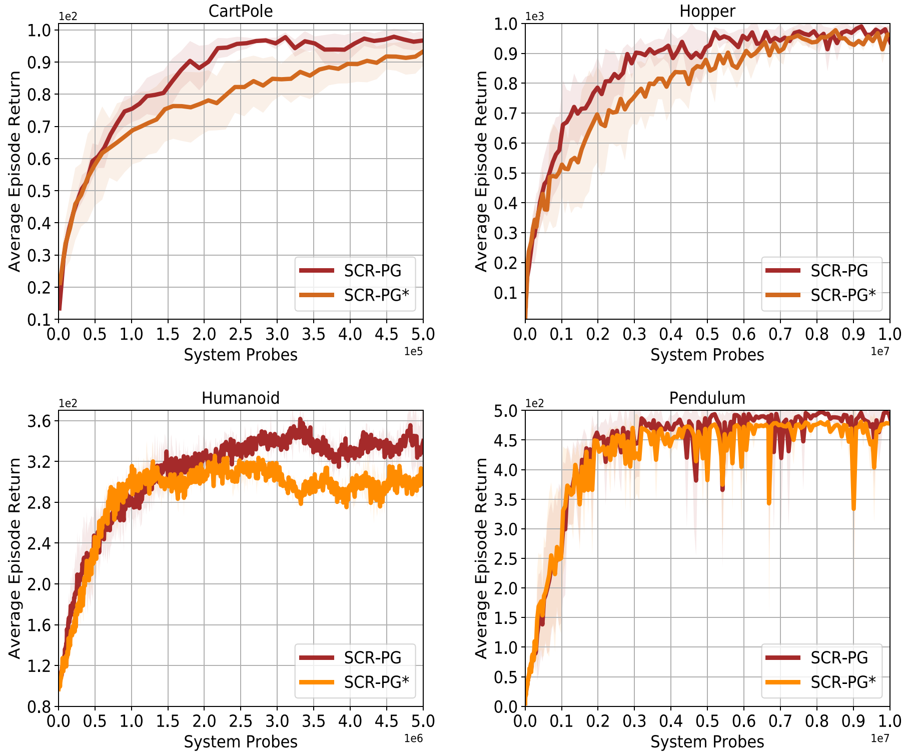
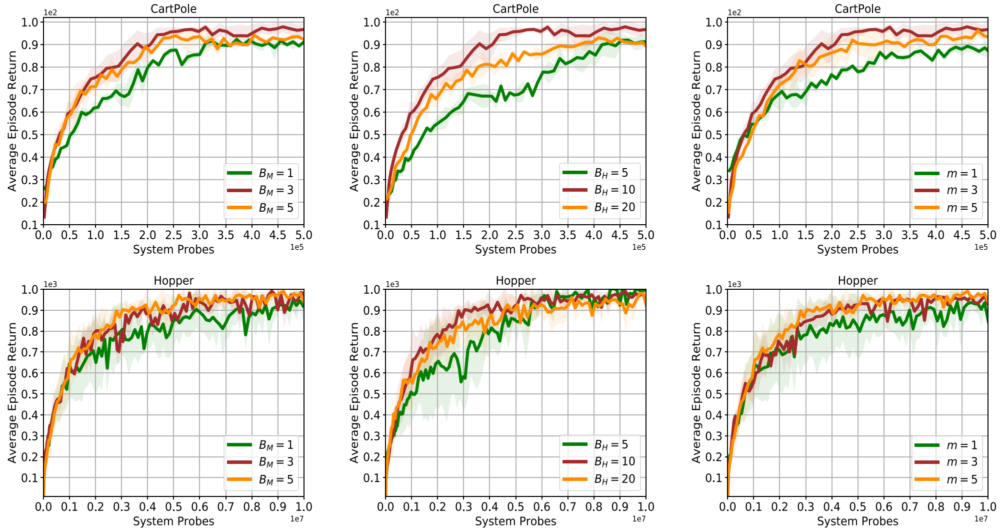


Fig. 2. Comparison of SCR-PG and SCR-PG* at four environments.



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

Fig. 3. a–c. Comparison of different parameters B_M, B_H, and m on the average episode return of SCR-PG at CartPole and Hopper environments.

In addition, we examine the effect of the batch sizes B_M, B_H, and the epoch size m within each epoch of SCR-PG on its performance. We choose batch size B_N of Algorithm 1 to be $N = 100$, choose mini-batch sizes B_M and B_H as $\{1, 3, 5\}$ and $\{5, 10, 20\}$ respectively, and choose the epoch size m as $\{1, 3, 5\}$. The results are shown in Fig. 3a–c. We find that SCR-PG with these different parameters can achieve quite good performance. This demonstrate that SCR-PG is not very sensitive to the selection of these hyper-parameters.

7. Conclusion

In this paper, we propose a policy gradient algorithm SCR-PG, which consists of two separate subroutines. SCR-PG minimizes the Hessian-based computations by invoking the second-order subroutines judiciously. The method only leverages stochastic gradients and Hessian–vector product evaluations and avoids solving the cubic submodel exactly. We prove that SCR-PG converges to $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSPs with high sample-efficiency.

Experimental results on four benchmark RL tasks confirm the superior performance of SCR-PG.

CRediT authorship contribution statement

Pengfei Wang: Conceptualization, Methodology, Software, Writing – original draft. **Hongyu Wang:** Methodology, Software. **Nenggan Zheng:** Supervision, Project administration, Resources, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Key R&D Program of China (2020YFB1313501), Zhejiang Provincial Natural Science Foundation, China (LR19F020005), National Natural Science Foundation of China (61972347), the Key R&D program of Zhejiang Province (2021C03003, 2022C01022, 2022C01119), the Fundamental Research Funds for the Central Universities, China (No. 226-2022-00051).

Appendix A. Proof of Lemma 1

(1): To prove the L-Lipschitz continuity of $\nabla_\theta J(\theta)$, we show that

$$\begin{aligned}
&\left\|\nabla_\theta^2 J(\theta)\right\| \\
&= \left\| \int_\tau \left(\nabla_\theta^2 \log p(\tau|\theta) + \nabla_\theta \log p(\tau|\theta)\nabla_\theta \log p(\tau|\theta)^{\mathrm{T}}\right) \right. \\
&\quad \left. \times\ p(\tau|\theta)R(\tau)d\tau \right\| \\
&\le \int_\tau \left(\left\|\nabla_\theta^2 \log p(\tau|\theta)\right\| + \|\nabla_\theta \log p(\tau|\theta)\|^2\right) p(\tau|\theta)\,\|R(\tau)\|\,d\tau.
\end{aligned} \tag{A.1}$$

Recall that $p(\tau|\theta) = \rho(s_0)\prod_{j=0}^{H-1}\pi_\theta(a_j|s_j)\mathcal{P}(s_{j+1}|s_j, a_j)$, by Assumption 1, we have $\|\nabla_\theta \log p(\tau|\theta)\| \le \sum_{j=0}^{H-1}\left\|\nabla_\theta \log \pi_\theta(a_j|s_j)\right\| \le HG$. Similarly, we have $\left\|\nabla_\theta^2 \log p(\tau|\theta)\right\| \le \sum_{j=0}^{H-1}\left\|\nabla_\theta^2 \log \pi_\theta(a_j|s_j)\right\| \le HM$. Then, we have

$$\left\|\nabla_\theta^2 J(\theta)\right\| \le \int_\tau (HM + H^2G^2)\,\|R(\tau)\|\,d\tau \le \frac{R}{1-\gamma}(HM + H^2G^2). \tag{A.2}$$

Let $L = \frac{R}{1-\gamma}(HM + H^2G^2)$, then $\nabla J(\theta)$ is Lipschitz continuous with Lipschitz constant L.

(2): To prove the ρ-Lipschitz continuity of $\nabla_\theta^2 J(\theta)$, for $\forall\theta^1$, $\theta^2 \in \mathbb{R}^d$, we have

$$\begin{aligned}
&\left\|\nabla_\theta^2 J(\theta^1) - \nabla_\theta^2 J(\theta^2)\right\| \\
&\le \frac{R}{1-\gamma}\left(\int_\tau \left\|\nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^1) - \nabla_\theta^2 \log p(\tau|\theta^2)p(\tau|\theta^2)\right\| \right. \\
&+\ \left\|\nabla_\theta \log p(\tau|\theta^1)\nabla_\theta \log p(\tau|\theta^1)^{\mathrm{T}}p(\tau|\theta^1) - \nabla_\theta \log p(\tau|\theta^2)\nabla_\theta \right. \\
&\quad \left.\left. \times\ \log p(\tau|\theta^2)^{\mathrm{T}}p(\tau|\theta^2)\right\| d\tau \right) \\
&\le \frac{R}{1-\gamma}\left(\int_\tau \left\|\nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^1) - \nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^2)\right\| \right. \\
&+\ \left\|\nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^2) - \nabla_\theta^2 \log p(\tau|\theta^2)p(\tau|\theta^2)\right\| \\
&+\ \left\|\nabla_\theta \log p(\tau|\theta^1)\nabla_\theta \log p(\tau|\theta^1)^{\mathrm{T}}(p(\tau|\theta^1) - p(\tau|\theta^2))\right\| \\
&+\ \left\|\nabla_\theta \log p(\tau|\theta^1)\nabla_\theta \log p(\tau|\theta^1)^{\mathrm{T}} \right. \\
&\quad \left.\left. -\ \nabla_\theta \log p(\tau|\theta^2)\nabla_\theta \log p(\tau|\theta^2)^{\mathrm{T}}\right\| d\tau \right).
\end{aligned} \tag{A.3}$$

In the following, we consider the four terms in RHS of (A.3), respectively.

First, we consider the first term as follows

$$\begin{aligned}
&\int_\tau \left\|\nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^1) - \nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^2)\right\| d\tau \\
&\le HM \int_\tau \left\|p(\tau|\theta^1) - p(\tau|\theta^2)\right\| d\tau \\
&\le HM \int_\tau \left\|\theta^1 - \theta^2\right\| HGd\tau \\
&= H^2GM \left\|\theta^1 - \theta^2\right\|,
\end{aligned} \tag{A.4}$$

where (A.4) holds by $\left\|\nabla_\theta^2 \log p(\tau|\theta^1)\right\| \le HM$ and $\|\nabla_\theta \log \pi_\theta(a|s)\| \le HG$.

Second, we consider the second term as follows

$$\begin{aligned}
&\int_\tau \left\|\nabla_\theta^2 \log p(\tau|\theta^1)p(\tau|\theta^2) - \nabla_\theta^2 \log p(\tau|\theta^2)p(\tau|\theta^2)\right\| d\tau \\
&\le \int_\tau \left\|\nabla_\theta^2 \log p(\tau|\theta^1) - \nabla_\theta^2 \log p(\tau|\theta^2)\right\| |p(\tau|\theta^2)|d\tau \\
&\le \int_\tau \left\|\nabla_\theta^2 \log p(\tau|\theta^1) - \nabla_\theta^2 \log p(\tau|\theta^2)\right\| d\tau \\
&\le \int_\tau \sum_{j=0}^{H-1} \left\|\nabla_\theta^2 \log \pi_{\theta^1}(a_j|s_j) - \nabla_\theta^2 \log \pi_{\theta^2}(a_j|s_j)\right\| d\tau \\
&\le HF \left\|\theta^1 - \theta^2\right\|,
\end{aligned} \tag{A.5}$$

where the last inequality comes from Assumption 2.

Third, we then consider the third term as follows

$$\begin{aligned}
&\int_\tau \left\|\nabla_\theta \log p(\tau|\theta^1)\nabla_\theta \log p(\tau|\theta^1)^{\mathrm{T}}(p(\tau|\theta^1) - p(\tau|\theta^2))\right\| d\tau \\
&\le \int_\tau \left\|\nabla_\theta \log p(\tau|\theta^1)\nabla_\theta \log p(\tau|\theta^1)^{\mathrm{T}}\right\| \left\|p(\tau|\theta^1) - p(\tau|\theta^2)\right\| d\tau \\
&\le H^2G^2 \int_\tau \left\|p(\tau|\theta^1) - p(\tau|\theta^2)\right\| d\tau \\
&\le H^3G^3 \left\|\theta^1 - \theta^2\right\|,
\end{aligned} \tag{A.6}$$

where the last two inequalities in (A.6) are derived as in (A.4).

Finally, we consider the fourth term as follows

$$\begin{aligned}
&\int_\tau \left\|\nabla_\theta \log p(\tau|\theta^1)\nabla_\theta \log p(\tau|\theta^1)^{\mathrm{T}} \right. \\
&\quad \left. -\ \nabla_\theta \log p(\tau|\theta^2)\nabla_\theta \log p(\tau|\theta^2)^{\mathrm{T}}\right\| d\tau \\
&\le \int_\tau \left\|\nabla_\theta \log p(\tau|\theta^1) + \nabla_\theta \log p(\tau|\theta^2)\right\| \\
&\quad \times \left\|\nabla_\theta \log p(\tau|\theta^1) - \nabla_\theta \log p(\tau|\theta^2)\right\| d\tau \\
&\le 2HG \int_\tau \left\|\nabla_\theta \log p(\tau|\theta^1) - \nabla_\theta \log p(\tau|\theta^2)\right\| d\tau \\
&\le 2HG \int_\tau HM \left\|\theta^1 - \theta^2\right\| d\tau = 2H^2GM \left\|\theta^1 - \theta^2\right\|,
\end{aligned} \tag{A.7}$$

where (A.7) holds by $\|\nabla_\theta \log p(\tau|\theta)\| \le HG$ and $\left\|\nabla_\theta^2 \log p(\tau|\theta)\right\| \le HM$. Substituting (A.4), (A.5), (A.6), and (A.7) into (A.3), we have

$$\left\|\nabla_\theta^2 J(\theta^1) - \nabla_\theta^2 J(\theta^2)\right\| \le \frac{(3H^2GM + H^3G^3 + HF)R}{1-\gamma}\left\|\theta^1 - \theta^2\right\|. \tag{A.8}$$

Let $\rho = \frac{(3H^2GM+H^3G^3+HF)R}{1-\gamma}$, then $\nabla^2_\theta J(\theta)$ is Lipschitz continuous with Lipschitz constant ρ.

(3): Recall that $g(\tau|\theta) = \sum_{j=0}^{H-1} \nabla_\theta \log \pi_\theta(a_j|s_j)R(\tau)$, by Assumption 1, we have $\|g(\tau|\theta)\| \le \sum_{j=0}^{H-1} \left\|\nabla_\theta \log \pi_\theta(a_j|s_j)\right\| \cdot \|R(\tau)\| \le \frac{HGR}{1-\gamma}$. Let $\kappa_f = \frac{HGR}{1-\gamma}$, then the stochastic gradient $g(\tau|\theta)$ is upper bounded by κ_f. Next, $\forall \theta^1, \theta^2 \in \mathbb{R}^d$, we have

$$\begin{aligned}
&\left\|g(\tau|\theta^1) - g(\tau|\theta^2)\right\| \\
&= \left\| \sum_{j=0}^{H-1} \left(\nabla_\theta \log \pi_{\theta^1}(a_j|s_j) - \nabla_\theta \log \pi_{\theta^2}(a_j|s_j)\right) R(\tau) \right\| \\
&\le \frac{R}{1-\gamma} \sum_{j=0}^{H-1} \left\|\nabla_\theta \log \pi_{\theta^1}(a_j|s_j) - \nabla_\theta \log \pi_{\theta^2}(a_j|s_j)\right\| \\
&\le \frac{R}{1-\gamma} \sum_{j=0}^{H-1} M \left\|\theta^1 - \theta^2\right\| \le \frac{HRM}{1-\gamma} \left\|\theta^1 - \theta^2\right\|. \qquad \text{(A.9)}
\end{aligned}$$

Let $\kappa_g = \frac{HRM}{1-\gamma}$, then $g(\tau|\theta)$ is Lipschitz continuous with Lipschitz constant κ_g.

Appendix B. Proof of Lemma 2

For notational convenience, define $y_i = g(\tau_i|\theta) - \nabla_\theta J(\theta)$ and $y = \frac{1}{B_N}\sum_{i=1}^{B_N} y_i$. By Lemma 1, the triangle inequality, and Jensens inequality, the matrix variance of $y = \frac{1}{B_N}\sum_{i=1}^{B_N} g(\tau_i|\theta) - \nabla_\theta J(\theta)$ is upper bounded by

$$\mathbf{v}[y] = \frac{1}{B_N^2} \max\left\{ \left\| \mathbb{E}\left[\sum_{i=1}^{B_N} y_i y_i^{\mathrm{T}}\right] \right\|, \left\| \mathbb{E}\left[\sum_{i=1}^{B_N} y_i^{\mathrm{T}} y_i\right] \right\| \right\} \le \frac{\kappa_f^2}{B_N}, \qquad \text{(B.1)}$$

where the inequality holds also by $\mathbb{E}\left[\|g(\tau_i|\theta) - \nabla_\theta J(\theta)\|^2\right] \le \mathbb{E}\left[\|g(\tau_i|\theta)\|^2\right] \le \kappa_f^2$. Again by Lemma 1, $\|g(\tau_i|\theta) - \nabla_\theta J(\theta)\| \le \|g(\tau_i|\theta)\| + \|\mathbb{E}\left[g(\tau|\theta)\right]\| \le 2\kappa_f$. Thus, a direct application of the matrix Bernstein inequality gives, $\forall t \ge 0$,

$$\begin{aligned}
\mathbb{P}[\|y\| \ge t] &\le 2d \exp\left(-\frac{t^2/2}{\mathbf{v}[y] + 2\kappa_f/(3B_N)}\right) \\
&\le 2d \exp\left(-\frac{3B_N}{8} \min\left\{\frac{t}{2\kappa_f}, \frac{t^2}{\kappa_f^2}\right\}\right). \qquad \text{(B.2)}
\end{aligned}$$

Substituting $t = c_1\epsilon$ into the above inequality (B.2), we have $\|\frac{1}{B_N}\sum_{i=1}^{B_N} g(\tau_i|\theta) - \nabla_\theta J(\theta)\| \le c_1\epsilon$ with probability $1-\delta'$ for $B_N \ge \frac{8}{3}\max\left(\frac{2\kappa_f}{c_1\epsilon}, \frac{\kappa_f^2}{c_1^2\epsilon^2}\right)\log\frac{2d}{\delta'}$.

Define $z_i = H(\tau_i|\theta) - \nabla^2_\theta J(\theta)$ and $z = \frac{1}{B_H}\sum_{i=1}^{B_H} z_i$. Since $g(\tau|\theta)$ is Lipschitz continuous with constant κ_g, then $\|H(\tau|\theta)\| \le \kappa_g$. Similarly, a direct application of the matrix Bernstein inequality gives, $\forall t \ge 0$

$$\begin{aligned}
\mathbb{P}[\|z\| \ge t] &\le 2d \exp\left(-\frac{t^2/2}{\mathbf{v}[z] + 2\kappa_g/(3B_H)}\right) \\
&\le 2d \exp\left(-\frac{3B_H}{8} \min\left\{\frac{t}{2\kappa_g}, \frac{t^2}{\kappa_g^2}\right\}\right). \qquad \text{(B.3)}
\end{aligned}$$

Substituting $t = c_2\sqrt{\epsilon\rho}$ into the above inequality (B.3), we have $\|\frac{1}{B_H}\sum_{i=1}^{B_N} H(\tau_i|\theta) - \nabla^2_\theta J(\theta)\| \le c_2\sqrt{\rho\epsilon}$ with probability $1-\delta'$ for $B_H \ge \frac{8}{3}\max\left(\frac{2\kappa_g}{c_2\sqrt{\epsilon\rho}}, \frac{\kappa_g^2}{c_2^2\epsilon\rho}\right)\log\frac{2d}{\delta'}$.

Appendix C. Proof of Lemma 4

Since $v_t^{s+1} = v_{t-1}^{s+1} + \frac{1}{B_M}\sum_{i=1}^{B_M}\left(g(\tau_i|\theta_t^{s+1}) - g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right)$, and $\mathbb{E}\left[\nabla_\theta J(\theta_t^{s+1}) - \nabla_\theta J(\theta_{t-1}^{s+1}) - \frac{1}{B_M}\sum_{i=1}^{B_M}\left(g(\tau_i|\theta_t^{s+1}) - g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right)\right] = 0$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - v_t^{s+1}\right\|^2\right] = \mathbb{E}\left[\left\|\nabla_\theta J(\theta_{t-1}^{s+1}) - v_{t-1}^{s+1}\right\|^2\right] \\
&+ \mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - \nabla_\theta J(\theta_{t-1}^{s+1}) - \frac{1}{B_M}\sum_{i=1}^{B_M}(g(\tau_i|\theta_t^{s+1}) \right.\right. \\
&\quad \left.\left. - g_\omega(\tau_i|\theta_{t-1}^{s+1}), \theta_t^{s+1})\right\|^2\right] \\
&\le \mathbb{E}\left[\left\|\nabla_\theta J(\theta_{t-1}^{s+1}) - v_{t-1}^{s+1}\right\|^2\right] \\
&\quad + \frac{1}{B_M^2}\sum_{i=1}^{B_M} \underbrace{\mathbb{E}\left[\left\|g(\tau_i|\theta_t^{s+1}) - g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right\|^2\right]}_{T_1}, \qquad \text{(C.1)}
\end{aligned}$$

where the last inequality holds by $\mathbb{E}\left[\|x - \mathbb{E}[x]\|^2\right] \le \mathbb{E}\left[\|x\|^2\right]$ for any $x \in \mathbb{R}^d$. Then, we give an upper bound of the term T_1 as follows:

$$\begin{aligned}
T_1 &= \mathbb{E}\left[\left\|g(\tau_i|\theta_t^{s+1}) - g_\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right\|^2\right] \\
&\le \mathbb{E}\left[\left\|g(\tau_i|\theta_t^{s+1}) - \omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})g(\tau_i|\theta_{t-1}^{s+1})\right\|^2\right] \\
&\le 2\mathbb{E}\left[\left\|\left(\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) - 1\right) g(\tau_i|\theta_{t-1}^{s+1})\right\|^2\right] \\
&\quad + 2\mathbb{E}\left[\left\|g(\tau_i|\theta_t^{s+1}) - g(\tau_i|\theta_{t-1}^{s+1})\right\|^2\right] \\
&\le 2\kappa_f^2\mathbb{E}\left[\left\|\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) - 1\right\|^2\right] + 2\kappa_g^2\mathbb{E}\left[\left\|\theta_t^{s+1} - \theta_{t-1}^{s+1}\right\|^2\right] \\
&\le 2(\kappa_f^2 C_W + \kappa_g^2)\mathbb{E}\left[\left\|\theta_t^{s+1} - \theta_{t-1}^{s+1}\right\|^2\right], \qquad \text{(C.2)}
\end{aligned}$$

where the third inequality holds by Lemma 1, and the last inequality holds by that $\mathrm{var}\left(\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1})\right) = \mathbb{E}\left[\left\|\omega(\tau_i|\theta_{t-1}^{s+1}, \theta_t^{s+1}) - 1\right\|^2\right] \le C_W\left\|\theta_t^{s+1} - \theta_{t-1}^{s+1}\right\|^2$ and $C_W = H(2HG^2 + M)(W+1)$. Combining (C.1) and (C.2), we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - v_t^{s+1}\right\|^2\right] \\
&\le \mathbb{E}\left[\left\|\nabla_\theta J(\theta_{t-1}^{s+1}) - v_{t-1}^{s+1}\right\|^2\right] + 2\frac{\kappa_f^2 C_W + \kappa_g^2}{B_M}\mathbb{E}\left[\left\|\theta_t^{s+1} - \theta_{t-1}^{s+1}\right\|^2\right] \\
&\le \mathbb{E}\left[\left\|\nabla_\theta J(\theta_0^{s+1}) - v_0^{s+1}\right\|^2\right] + 2\frac{\kappa_f^2 C_W + \kappa_g^2}{B_M} \\
&\quad \times \sum_{i=1}^{t}\mathbb{E}\left[\left\|\theta_i^{s+1} - \theta_{i-1}^{s+1}\right\|^2\right]. \qquad \text{(C.3)}
\end{aligned}$$

Summing the above inequality (C.3) over t from 0 to $m-1$, we have

$$\begin{aligned}
&\sum_{t=0}^{m-1}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - v_t^{s+1}\right\|^2\right] \\
&\le m\mathbb{E}\left[\left\|\nabla_\theta J(\theta_0^{s+1}) - v_0^{s+1}\right\|^2\right] + 2\frac{\kappa_f^2 C_W + \kappa_g^2}{B_M} \\
&\quad \times \sum_{t=1}^{m-1}\sum_{i=1}^{t}\mathbb{E}\left[\left\|\theta_i^{s+1} - \theta_{i-1}^{s+1}\right\|^2\right] \\
&\le m\mathbb{E}\left[\left\|\nabla_\theta J(\theta_0^{s+1}) - v_0^{s+1}\right\|^2\right] + 2m\frac{\kappa_f^2 C_W + \kappa_g^2}{B_M} \\
&\quad \times \sum_{t=1}^{m-1}\mathbb{E}\left[\left\|\theta_t^{s+1} - \theta_{t-1}^{s+1}\right\|^2\right]. \qquad \text{(C.4)}
\end{aligned}$$

Next, according to the updating rule and Lemma 1, it holds that

$$\begin{aligned}\mathbb{E}\left[J(\theta_{t+1}^{s+1})\right] \ge{}& \mathbb{E}\left[J(\theta_t^{s+1})\right] + \mathbb{E}\left[\langle \nabla_\theta J(\theta_t^{s+1}), \theta_{t+1}^{s+1} - \theta_t^{s+1}\rangle\right] \\ &- \frac{L}{2}\mathbb{E}\left[\left\|\theta_{t+1}^{s+1} - \theta_t^{s+1}\right\|^2\right] \\ \ge{}& \mathbb{E}\left[J(\theta_t^{s+1})\right] + \eta\mathbb{E}\left[\langle \nabla_\theta J(\theta_t^{s+1}), v_t^{s+1}\rangle\right] \\ &- \frac{\eta^2 L}{2}\mathbb{E}\left[\left\|v_t^{s+1}\right\|^2\right] \\ ={}& \mathbb{E}\left[J(\theta_t^{s+1})\right] + \frac{\eta}{2}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1})\right\|^2\right] \\ &+ \left(\frac{\eta}{2} - \frac{\eta^2 L}{2}\right)\mathbb{E}\left[\left\|v_t^{s+1}\right\|^2\right] \\ &- \frac{\eta}{2}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - v_t^{s+1}\right\|^2\right]. \end{aligned} \tag{C.5}$$

Then, we can rewrite (C.5) as follows:

$$\begin{aligned}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1})\right\|^2\right] \le{}& \frac{2}{\eta}\mathbb{E}\left[J(\theta_{t+1}^{s+1}) - J(\theta_t^{s+1})\right] \\ &+ \mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - v_t^{s+1}\right\|^2\right] \\ &- (1-\eta L)\mathbb{E}\left[\left\|v_t^{s+1}\right\|^2\right]. \end{aligned} \tag{C.6}$$

Summing the above inequality (C.6) over $t = 0$ to $m-1$, we have

$$\begin{aligned}\sum_{t=0}^{m-1}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1})\right\|^2\right] \le{}& \frac{2}{\eta}\mathbb{E}[J(\theta_m^{s+1}) - J(\theta_0^{s+1})] \\ &+ \sum_{t=0}^{m-1}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1}) - v_t^{s+1}\right\|^2\right] \\ &- (1-\eta L)\sum_{t=0}^{m-1}\mathbb{E}\left[\left\|v_t^{s+1}\right\|^2\right]. \end{aligned} \tag{C.7}$$

Substituting (C.4) into the above (C.7), we have

$$\begin{aligned}\sum_{t=0}^{m-1}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1})\right\|^2\right] \le{}& \frac{2}{\eta}\mathbb{E}[J(\theta_m^{s+1}) - J(\theta_0^{s+1})] \\ &+ m\mathbb{E}\left[\left\|\nabla_\theta J(\theta_0^{s+1}) - v_0^{s+1}\right\|^2\right] \\ &- \left(1 - \eta L - 2m\eta^2\frac{\kappa_f^2 C_W + \kappa_g^2}{B_M}\right) \\ &\times \sum_{t=0}^{m-1}\mathbb{E}\left[\left\|v_t^{s+1}\right\|^2\right]. \end{aligned} \tag{C.8}$$

Given $B_M = \frac{\kappa_f^2 C_W + \kappa_g^2}{\epsilon^{1/2}}$, $m = \frac{L^2}{\epsilon^{1/2}}$, and $\eta = \frac{1}{2L}$, we see that $1 - \eta L - 2m\eta^2\frac{\kappa_f^2 C_W + \kappa_g^2}{B_M} \ge 0$. Under the setting of Lemma 2, we have

$$\mathbb{E}\left[\left\|\nabla_\theta J(\theta_0^{s+1}) - v_0^{s+1}\right\|^2\right] \le (c_1\epsilon)^2. \tag{C.9}$$

The above analysis implies that

$$\frac{1}{m}\sum_{t=0}^{m-1}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1})\right\|^2\right] \le \frac{2}{\eta m}\mathbb{E}[J(\theta_m^{s+1}) - J(\theta_0^{s+1})] + (c_1\epsilon)^2. \tag{C.10}$$

Since $0 \le \frac{1}{m}\sum_{t=0}^{m-1}\mathbb{E}\left[\left\|\nabla_\theta J(\theta_t^{s+1})\right\|^2\right]$, we have $0 \le \frac{2}{\eta m}\mathbb{E}[J(\theta_m^{s+1}) - J(\theta_0^{s+1})] + (c_1\epsilon)^2$ holds for any $\epsilon > 0$. Then $\mathbb{E}\left[J(\tilde{\theta}^{s+1})\right] \ge \mathbb{E}\left[J(\tilde{\theta}^{s})\right]$. On the other hand, substituting $B_M = \frac{\kappa_f^2 C_W + \kappa_g^2}{\epsilon^{1/2}}$, $m = \frac{L^2}{\epsilon^{1/2}}$, and $\eta = \frac{1}{2L}$ into (C.10), we then have

$$\mathbb{E}\left[\left\|\nabla_\theta J(\widehat{\theta}^{s+1})\right\|^2\right] \le 4\epsilon^{1/2}L^{-1}\mathbb{E}[J(\widetilde{\theta}^{s+1}) - J(\widetilde{\theta}^{s})] + (c_1\epsilon)^2. \tag{C.11}$$

Appendix D. Proof of Lemma 5

Recall that $\Delta_*^{s+1} = \arg\max_{\Delta\in\mathbb{R}^d}\widetilde{m}^{s+1}(\Delta)$, as shown in [44], a global optimum of $\widetilde{m}^{s+1}(\Delta)$ satisfies:

$$g^{s+1} + H^{s+1}[\Delta_*^{s+1}] - \frac{\rho}{2}\left\|\Delta_*^{s+1}\right\|\Delta_*^{s+1} = 0, \tag{D.1}$$

$$H^{s+1} - \frac{\rho}{2}\left\|\Delta_*^{s+1}\right\|\mathbf{I} \preceq 0, \tag{D.2}$$

where $\mathbf{I}$ is the identity matrix with the same size as H^{s+1}. Then, we have

$$\begin{aligned}&m^{s+1}(\widehat{\theta}^{s+1} + \Delta_*^{s+1}) - m^{s+1}(\widehat{\theta}^{s+1}) \\ &= (\Delta_*^{s+1})^{\mathrm{T}}g^{s+1} + \frac{1}{2}(\Delta_*^{s+1})^{\mathrm{T}}H^{s+1}[\Delta_*^{s+1}] - \frac{\rho}{6}\left\|\Delta_*^{s+1}\right\|^3 \\ &= -\frac{1}{2}(\Delta_*^{s+1})^{\mathrm{T}}H^{s+1}[\Delta_*^{s+1}] + \frac{\rho}{3}\left\|\Delta_*^{s+1}\right\|^3 \\ &\ge -\frac{\rho}{4}\left\|\Delta_*^{s+1}\right\|^3 + \frac{\rho}{3}\left\|\Delta_*^{s+1}\right\|^3 = \frac{\rho}{12}\left\|\Delta_*^{s+1}\right\|^3. \end{aligned} \tag{D.3}$$

From (D.3), to guarantee sufficient ascent it suffices to lower bound $\left\|\Delta_*^{s+1}\right\|$. Based on (D.2), we have $\left\|\Delta_*^{s+1}\right\| \ge \frac{2}{\rho}\left\|H^{s+1}\right\|$. We derive the lower bound of $\left\|H^{s+1}\right\|$ as:

$$\begin{aligned}\left\|H^{s+1}\right\| &= \left\|H^{s+1} - \nabla_\theta^2 J(\widehat{\theta}^{s+1}) + \nabla_\theta^2 J(\widehat{\theta}^{s+1})\right\| \\ &\ge \left\|\nabla_\theta^2 J(\widehat{\theta}^{s+1})\right\| - \underbrace{\left\|H^{s+1} - \nabla_\theta^2 J(\widehat{\theta}^{s+1})\right\|}_{T_2}, \end{aligned} \tag{D.4}$$

According to Lemma 2, we given an upper bound of the term T_2:

$$T_2 = \left\|H^{s+1} - \nabla_\theta^2 J(\widehat{\theta}^{s+1})\right\| \le c_2\sqrt{\rho\epsilon}. \tag{D.5}$$

When $\lambda_{\max}\left(\nabla_\theta^2 J(\widehat{\theta}^{s+1})\right) \ge \sqrt{\rho\epsilon}$, we have $\left\|\Delta_*^{s+1}\right\| \ge \frac{2}{\rho}\left\|H^{s+1}\right\| \ge 2(1-c_2)\sqrt{\frac{\epsilon}{\rho}}$. By Condition 1 and Lemma 3, we have

$$m^{s+1}(\widehat{\theta}^{s+1} + \Delta^{s+1}) \ge m^{s+1}(\widehat{\theta}^{s+1} + \Delta_*^{s+1}) - \frac{c_4}{12}\rho\left\|\Delta_*^{s+1}\right\|^3. \tag{D.6}$$

Combining (D.3) and (D.6), it follows that

$$\Delta_m^{s+1} = m^{s+1}(\widehat{\theta}^{s+1} + \Delta^{s+1}) - m^{s+1}(\widehat{\theta}^{s+1}) \ge (1-c_4)\frac{\rho}{12}\left\|\Delta_*^{s+1}\right\|^3. \tag{D.7}$$

According to Lemmas 2 and 3 and (D.7), we consider $J(\widetilde{\theta}^{s+1}) - J(\widehat{\theta}^{s+1})$ as follows

$$\begin{aligned}&J(\widetilde{\theta}^{s+1}) - J(\widehat{\theta}^{s+1}) \\ &\ge (\Delta^{s+1})^{\mathrm{T}}\nabla_\theta J(\widehat{\theta}^{s+1}) + \frac{1}{2}(\Delta^{s+1})^{\mathrm{T}}\nabla_\theta^2 J(\widehat{\theta}^{s+1})\Delta^{s+1} - \frac{\rho}{6}\left\|\Delta^{s+1}\right\|^3 \\ &\ge m^{s+1}(\widehat{\theta}^{s+1} + \Delta^{s+1}) - m^{s+1}(\widehat{\theta}^{s+1}) - c_1\epsilon\left\|\Delta^{s+1}\right\| \\ &\quad - \frac{1}{2}c_2\sqrt{\rho\epsilon}\left\|\Delta^{s+1}\right\|^2 \\ &\ge m^{s+1}(\widehat{\theta}^{s+1} + \Delta^{s+1}) - m^{s+1}(\widehat{\theta}^{s+1}) \\ &\quad - c_1\epsilon\left(\left\|\Delta_*^{s+1}\right\| + c_3\sqrt{\frac{\epsilon}{\rho}}\right) - \frac{1}{2}c_2\sqrt{\rho\epsilon}\left(\left\|\Delta_*^{s+1}\right\| + c_3\sqrt{\frac{\epsilon}{\rho}}\right)^2 \\ &\ge (1-c_4)\frac{\rho}{12}\left\|\Delta_*^{s+1}\right\|^3 - \frac{1}{2}c_2\sqrt{\rho\epsilon}\left\|\Delta_*^{s+1}\right\|^2 \\ &\quad - (c_1 + c_2c_3)\epsilon\left\|\Delta_*^{s+1}\right\| - \left(c_1c_2 + \frac{c_2c_3^2}{2}\right)\sqrt{\frac{\epsilon^3}{\rho}}. \end{aligned} \tag{D.8}$$

Given that $\left\|\Delta_*^{s+1}\right\| \geq 2(1-c_2)\sqrt{\frac{\epsilon}{\rho}}$, we have

$$\begin{aligned} & J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \\ & \geq \frac{2}{3}\left(1-3c_1-6c_2-c_4-4c_2^3-3c_2^2c_4-3c_2c_3\right. \\ & \quad \left. -\frac{3}{2}c_1c_2-\frac{3c_2c_3^2}{4}\right)\sqrt{\frac{\epsilon^3}{\rho}}, \end{aligned} \tag{D.9}$$

where the numerical constants c_1, c_2, c_3, c_4 can be made arbitrarily small. Now let $c_5 = 3c_1 + 6c_2 + c_4 + 4c_2^3 + 3c_2^2c_4 + 3c_2c_3 + \frac{3}{2}c_1c_2 + \frac{3c_2c_3^2}{4}$, we have

$$J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq \frac{2}{3}(1-c_5)\sqrt{\frac{\epsilon^3}{\rho}}. \tag{D.10}$$

Next, we prove $J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq 0$ always holds. To this end, we only prove that $J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq 0$, when $\nabla_\theta^2 J(\widehat{\theta}^{s+1}) \preceq \sqrt{\rho\epsilon}\mathbf{I}$. Our analysis is divided into two cases: $\left\|\Delta_*^{s+1}\right\| \geq 2(1-c_2)\sqrt{\frac{\epsilon}{\rho}}$ and $\left\|\Delta_*^{s+1}\right\| \leq 2(1-c_2)\sqrt{\frac{\epsilon}{\rho}}$. When $\left\|\Delta_*^{s+1}\right\| \geq 2(1-c_2)\sqrt{\frac{\epsilon}{\rho}}$, the above analysis implies that $J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq \frac{2}{3}(1-c_5)\sqrt{\frac{\epsilon^3}{\rho}}$. When $\left\|\Delta_*^{s+1}\right\| \leq 2(1-c_3)\sqrt{\frac{\epsilon}{\rho}}$, based on (D.8), we have

$$\begin{aligned} & J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq m^{s+1}(\widehat{\theta}^{s+1}+\Delta^{s+1})-m^{s+1}(\widehat{\theta}^{s+1}) \\ & -c_1\epsilon\left(\left\|\Delta_*^{s+1}\right\|+c_3\sqrt{\frac{\epsilon}{\rho}}\right)-\frac{1}{2}c_2\sqrt{\rho\epsilon}\left(\left\|\Delta_*^{s+1}\right\|+c_3\sqrt{\frac{\epsilon}{\rho}}\right)^2. \end{aligned} \tag{D.11}$$

Assume $\Delta_m^{s+1} \geq \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$ (if not, SCR-PG terminates early), we derive (D.11) as follows

$$J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq \left(\frac{1}{2}-c_1(2-c_3)-\frac{1}{2}c_2(2-c_3)^2\right)\sqrt{\frac{\epsilon^3}{\rho}}. \tag{D.12}$$

Define $c_6 = c_1(2-c_3)+\frac{1}{2}c_2(2-c_3)^2$ be a small enough constant (depends on c_1, c_2 and c_3), we have $J(\widetilde{\theta}^{s+1})-J(\widehat{\theta}^{s+1}) \geq \left(\frac{1}{2}-c_6\right)\sqrt{\frac{\epsilon^3}{\rho}}$.

Appendix E. Proof of Theorem 1

We prove that in each iteration the function value will achieve an ascent greater than $\Omega(\epsilon^{1.5})$, otherwise the algorithm terminates early and output an ϵ-approximate SOSP. First, when $\left\|g^{s+1}\right\| > \epsilon$, by Lemma 2, we have

$$\begin{aligned} \left\|\nabla_\theta J(\widehat{\theta}^{s+1})\right\| &= \left\|g^{s+1}+\nabla_\theta J(\widehat{\theta}^{s+1})-g^{s+1}\right\| \\ &\geq \epsilon-\left\|g^{s+1}-\nabla_\theta J(\widehat{\theta}^{s+1})\right\| \geq (1-c_1)\epsilon, \end{aligned} \tag{E.1}$$

From Jensen's inequality and (E.1), we have

$$\mathbb{E}\left[\left\|\nabla_\theta J(\widehat{\theta}^{s+1})\right\|^2\right] \geq \left(\mathbb{E}\left[\left\|\nabla_\theta J(\widehat{\theta}^{s+1})\right\|\right]\right)^2 \geq (1-c_1)^2\epsilon^2. \tag{E.2}$$

By Lemma 4, we then have

$$(1-c_1)^2\epsilon^2 \leq 4\epsilon^{1/2}L^{-1}\mathbb{E}\left[J(\widetilde{\theta}^{s+1})-J(\widetilde{\theta}^{s})\right]+(c_1\epsilon)^2. \tag{E.3}$$

Rewriting (E.3), we have

$$\mathbb{E}\left[J(\widetilde{\theta}^{s+1})\right] \geq \mathbb{E}\left[J(\widetilde{\theta}^{s})\right]+\frac{1}{4}(1-2c_1)\epsilon^{3/2}L. \tag{E.4}$$

Table F.1
Hyper-parameter settings.

Environments	CartPole	Hopper	Humanoid	Pendulum
Horizon	100	1000	500	500
Neural network sizes	8 × 8	64 × 64	64 × 64	64 × 64
NN activation function	Tanh	Tanh	Tanh	Tanh
Number of timesteps	5×10^5	1×10^7	5×10^6	1×10^7
Is-MBPG k	0.75	0.75	0.75	0.75
Is-MBPG c	2	1	2	1
Is-MBPG m	2	1	1	1
REINFORCE η	0.01	0.01	0.01	0.01
B_N	100	100	100	100
B_M	3	5	5	5
B_H	10	10	10	10
m	3	5	5	5
$\mathcal{T}(\epsilon)$	3	3	3	3
SCR-PG η	0.1	0.1	0.01	0.1

Second, when $\left\|g^{s+1}\right\| \leq \epsilon$ and $\lambda_{\max}\left(\nabla_\theta^2 J(\widehat{\theta}^{s+1})\right) \geq \sqrt{\rho\epsilon}$, $\widehat{\theta}^{s+1}$ is not a SOSP, and SCR-PG enters the second-order subroutine. By Lemmas 4 and 5,

$$\begin{aligned} \mathbb{E}\left[J(\widetilde{\theta}^{s+1})\right] &\geq \mathbb{E}\left[J(\widehat{\theta}^{s+1})\right]+\frac{2}{3}(1-c_5)\sqrt{\frac{\epsilon^3}{\rho}} \\ &\geq \mathbb{E}\left[J(\widetilde{\theta}^{s})\right]+\frac{2}{3}(1-c_5)\sqrt{\frac{\epsilon^3}{\rho}}. \end{aligned} \tag{E.5}$$

Third, when $\left\|g^{s+1}\right\| \leq \epsilon$ and $\nabla_\theta^2 J(\widehat{\theta}^{s+1}) \preceq \sqrt{\rho\epsilon}\mathbf{I}$, by Lemma 2, we have

$$\begin{aligned} \left\|\nabla_\theta J(\widehat{\theta}^{s+1})\right\| &\leq \left\|g^{s+1}+\nabla_\theta J(\widehat{\theta}^{s+1})-g^{s+1}\right\| \\ &\leq \epsilon+\left\|g^{s+1}-\nabla_\theta J(\widehat{\theta}^{s+1})\right\| \leq (1+c_1)\epsilon. \end{aligned} \tag{E.6}$$

(E.6) indicates that by simply rescaling ϵ, $\widehat{\theta}^{s+1}$ is an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSP. By Lemma 5, if $\lambda_{\max}(\nabla_\theta^2 J(\widehat{\theta}^{s+1})) \geq \sqrt{\rho\epsilon}$, we have $\Delta_m^s \geq \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$ for sufficiently small c_2, c_4. Conversely, when $\Delta_m^{s+1} < \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$, SCR-PG falls in the third case. In this case, even if $\Delta_m^{s+1} \geq \frac{1}{2}\sqrt{\frac{\epsilon^3}{\rho}}$, (D.12) ensures that $J(\widetilde{\theta}^{s+1})-J(\widetilde{\theta}^{s}) \geq \left(\frac{1}{2}-c_6\right)\sqrt{\frac{\epsilon^3}{\rho}}$.

Define $\bar{c} = \min\{\frac{1}{4}(1-2c_1)\sqrt{\rho}L, \frac{2}{3}(1-c_5), \frac{1}{2}-c_6\}$, Algorithm 1 happens at most $S = \left\lceil \frac{\sqrt{\rho}(J^*-J(\theta^0))}{\bar{c}\sqrt{\epsilon^3}} \right\rceil$, otherwise Algorithm 1 will output an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSP. At each iteration, the event of the concentration conditions and Cubic-Subsolver outputs a satisfying Condition 1 hold has probability greater than $1-4\delta'$. Assume that $1-4\delta'S \geq 1-\delta$, set $\delta' = \frac{\delta\bar{c}\sqrt{\epsilon^3}}{4\sqrt{\rho}(J^*-J(\theta^0))}$, we conclude that Algorithm 1 will output an $(\epsilon, \sqrt{\rho\epsilon})$-approximate SOSP with probability at least $1-\delta$ within $S = \left\lceil \frac{\sqrt{\rho}(J^*-J(\theta^0))}{\bar{c}\sqrt{\epsilon^3}} \right\rceil$. Let $c = \min\{\bar{c}, c_1, c_2\}$, we complete the proof.

Proof of Corollary 1. The stochastic gradient sample complexity is $S \times (2B_N + m \times B_M)$, i.e., $\mathcal{O}\left(\epsilon^{-3.5}\right)$. The stochastic Hessian–vector product sample complexity is $S \times B_H \times \mathcal{T}(\epsilon)$, i.e., $\widetilde{\mathcal{O}}\left(\epsilon^{-3}\right)$. □

Appendix F. Detail hyper-parameter settings

We present the hyper-parameter settings in Table F.1.

References

[1] J. Kober, J.A. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, Int. J. Robot. Res. 32 (2013) 1238–1274.

[2] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al., Scalable deep reinforcement learning for vision-based robotic manipulation, in: Conference on Robot Learning, 2018, pp. 651–673.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, Nature 550 (2017) 354–359.

[4] O. Vinyals, I. Babuschkin, W.M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D.H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al., Grandmaster level in StarCraft II using multi-agent reinforcement learning, Nature 575 (2019) 350–354.

[5] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al., Highly accurate protein structure prediction with AlphaFold, Nature 596 (2021) 583–589.

[6] A. Tiwari, S. Saha, P. Bhattacharyya, A knowledge infused context driven dialogue agent for disease diagnosis using hierarchical reinforcement learning, Knowl.-Based Syst. 242 (2022) 108292.

[7] A.E. Sallab, M. Abdou, E. Perot, S. Yogamani, Deep reinforcement learning framework for autonomous driving, Electron. Imaging 2017 (2017) 70–76.

[8] Q. Wu, J. Wu, J. Shen, B. Du, A. Telikani, M. Fahmideh, C. Liang, Distributed agent-based deep reinforcement learning for large scale traffic signal control, Knowl.-Based Syst. 241 (2022) 108304.

[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[10] W. Deng, X. Zhang, Y. Zhou, Y. Liu, X. Zhou, H. Chen, H. Zhao, An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems, Inform. Sci. 585 (2022) 441–453.

[11] W. Deng, Z. Li, X. Li, H. Chen, H. Zhao, Compound fault diagnosis using optimized MCKD and sparse representation for rolling bearings, IEEE Trans. Instrum. Meas. 71 (2022) 1–9.

[12] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Massachusetts, 2018.

[13] R. Bellman, On the theory of dynamic programming, Proc. Natl. Acad. Sci. USA 38 (1952) 716.

[14] R.S. Sutton, D. McAllester, S. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, Adv. Neural Inf. Process. Syst. 12 (1999) 1057–1063.

[15] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, M. Riedmiller, Deterministic policy gradient algorithms, in: International Conference on Machine Learning, 2014, pp. 387–395.

[16] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust region policy optimization, in: International Conference on Machine Learning, 2015, pp. 1889–1897.

[17] D. Lee, N. He, P. Kamalaruban, V. Cevher, Optimization for reinforcement learning: From a single agent to cooperative agents, IEEE Signal Process. Mag. 37 (2020) 123–135.

[18] I.H. Witten, An adaptive optimal controller for discrete-time Markov environments, Inf. Control 34 (1977) 286–295.

[19] A.G. Barto, R.S. Sutton, C.W. Anderson, Neuronlike adaptive elements that can solve difficult learning control problems, IEEE Trans. Syst. Man Cybern. (1983) 834–846.

[20] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (1992) 229–256.

[21] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Vol. 87, Springer Science & Business Media, Berlin, 2003.

[22] L. Weaver, N. Tao, The optimal reward baseline for gradient-based reinforcement learning, 2013, arXiv preprint arXiv:1301.2315.

[23] P.S. Thomas, E. Brunskill, Policy gradient methods for reinforcement learning with function approximation and action-dependent baselines, 2017, arXiv preprint arXiv:1706.06643.

[24] V. Konda, J. Tsitsiklis, Actor-critic algorithms, Adv. Neural Inf. Process. Syst. 12 (1999).

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

[26] M. Papini, D. Binaghi, G. Canonaco, M. Pirotta, M. Restelli, Stochastic variance-reduced policy gradient, in: International Conference on Machine Learning, 2018, pp. 4026–4035.

[27] P. Xu, F. Gao, Q. Gu, An improved convergence analysis of stochastic variance-reduced policy gradient, in: Uncertainty in Artificial Intelligence, 2020, pp. 541–551.

[28] P. Xu, F. Gao, Q. Gu, Sample efficient policy gradient methods with recursive variance reduction, 2019, arXiv preprint arXiv:1909.08610.

[29] Z. Shen, A. Ribeiro, H. Hassani, H. Qian, C. Mi, Hessian aided policy gradient, in: International Conference on Machine Learning, 2019, pp. 5729–5738.

[30] F. Huang, S. Gao, J. Pei, H. Huang, Momentum-Based Policy Gradient Methods, in: Proceedings of Machine Learning and Systems 2020, 2020, pp. 3996–4007.

[31] J. Bhandari, D. Russo, Global optimality guarantees for policy gradient methods, 2019, arXiv: Learning.

[32] P. Jain, P. Kar, Non-convex optimization for machine learning, 2017, arXiv preprint arXiv:1712.07897.

[33] M. Fazel, R. Ge, S. Kakade, M. Mesbahi, Global convergence of policy gradient methods for the linear quadratic regulator, in: International Conference on Machine Learning, 2018, pp. 1467–1476.

[34] A. Agarwal, S.M. Kakade, J.D. Lee, G. Mahajan, Optimality and approximation with policy gradient methods in Markov decision processes, in: Conference on Learning Theory, 2020, pp. 64–66.

[35] J. Mei, C. Xiao, C. Szepesvari, D. Schuurmans, On the global convergence rates of softmax policy gradient methods, 2020, arXiv preprint arXiv:2005.06392.

[36] S. Cen, C. Cheng, Y. Chen, Y. Wei, Y. Chi, Fast global convergence of natural policy gradient methods with entropy regularization, Oper. Res. (2021).

[37] V. Fathi, J. Arabneydi, A.G. Aghdam, Reinforcement learning in linear quadratic deep structured teams: Global convergence of policy gradient methods, in: 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp. 4927–4932.

[38] D.D. Castro, R. Meir, A convergent online single time scale actor critic algorithm, J. Mach. Learn. Res. 11 (2010) 367–410.

[39] T. Xu, Z. Wang, Y. Liang, Non-asymptotic convergence analysis of two time-scale (natural) actor-critic algorithms, 2020, arXiv preprint arXiv:2005.03557.

[40] N. Pham, L. Nguyen, D. Phan, P.H. Nguyen, M. Dijk, Q. Tran-Dinh, A hybrid stochastic policy gradient algorithm for reinforcement learning, in: International Conference on Artificial Intelligence and Statistics, 2020, pp. 374–385.

[41] K. Zhang, A. Koppel, H. Zhu, T. Basar, Global convergence of policy gradient methods to (almost) locally optimal policies., 2019, arXiv: Optimization and Control.

[42] H. Daneshmand, J. Kohler, A. Lucchi, T. Hofmann, Escaping saddles with stochastic gradients, in: International Conference on Machine Learning, 2018, pp. 1155–1164.

[43] L. Yang, Q. Zheng, G. Pan, Sample Complexity of Policy Gradient Finding Second-Order Stationary Points, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 10630–10638.

[44] Y. Nesterov, B.T. Polyak, Cubic regularization of Newton method and its global performance, Math. Program. 108 (2006) 177–205.

[45] N. Tripuraneni, M. Stern, C. Jin, J. Regier, M.I. Jordan, Stochastic cubic regularization for fast nonconvex optimization, in: Advances in Neural Information Processing Systems, 2018, pp. 2899–2908.

[46] N. Agarwal, Z. Allen-Zhu, B. Bullins, E. Hazan, T. Ma, Finding approximate local minima for nonconvex optimization in linear time, 2016, arXiv preprint arXiv:1611.01146.

[47] S. Reddi, M. Zaheer, S. Sra, B. Poczos, F. Bach, R. Salakhutdinov, A. Smola, A generic approach for escaping saddle points, in: International Conference on Artificial Intelligence and Statistics, 2018, pp. 1233–1242.

[48] D. Zhou, P. Xu, Q. Gu, Stochastic variance-reduced cubic regularized Newton method, in: International Conference on Machine Learning, 2018, pp. 5985–5994.

[49] D. Zhou, Q. Gu, Stochastic recursive variance-reduced cubic regularization methods, in: International Conference on Artificial Intelligence and Statistics, 2020, pp. 3980–3990.

[50] R. Ge, F. Huang, C. Jin, Y. Yuan, Escaping from saddle points—online stochastic gradient for tensor decomposition, in: Conference on Learning Theory, 2015, pp. 797–842.

[51] C. Jin, R. Ge, P. Netrapalli, S.M. Kakade, M.I. Jordan, How to escape saddle points efficiently, in: International Conference on Machine Learning, 2017, pp. 1724–1732.

[52] J.D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M.I. Jordan, B. Recht, First-order methods almost always avoid strict saddle points, Math. Program. 176 (2019) 311–337.

[53] C.C. Margossian, A review of automatic differentiation and its efficient implementation, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 9 (2019) e1305.

[54] L.M. Nguyen, J. Liu, K. Scheinberg, M. Takáč, SARAH: A novel method for machine learning problems using stochastic recursive gradient, in: International Conference on Machine Learning, 2017, pp. 2613–2621.

[55] L. Xiao, T. Zhang, A proximal stochastic gradient method with progressive variance reduction, SIAM J. Optim. 24 (2014) 2057–2075.

[56] M. Pirotta, M. Restelli, L. Bascetta, Policy gradient in lipschitz markov decision processes, Mach. Learn. 100 (2015) 255–283.

[57] K. Zhang, Z. Yang, H. Liu, T. Zhang, T. Basar, Fully decentralized multi-agent reinforcement learning with networked agents, in: International Conference on Machine Learning, 2018, pp. 5872–5881.

[58] T. Chen, K. Zhang, G.B. Giannakis, T. Basar, Communication-efficient distributed reinforcement learning, 2018, arXiv preprint arXiv:1812.03239.

[59] M. Pirotta, M. Restelli, L. Bascetta, Adaptive step-size for policy gradient methods, Adv. Neural Inf. Process. Syst. 26 (2013).

[60] C. Cortes, Y. Mansour, M. Mohri, Learning bounds for importance weighting, Adv. Neural Inf. Process. Syst. 23 (2010).

[61] J.A. Tropp, An introduction to matrix concentration inequalities, Found. Trends® Mach. Learn. 8 (2015) 1–230.

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

[63] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ IROS, 2012, pp. 5026–5033.