Accelerating the Convergence of Policy Gradient Methods

Anant Shyam
Cornell University
ais64@cornell.edu

Daniel Cao
Cornell University
dyc33@cornell.edu

Abstract

Policy gradient methods are critical for finding the optimal parameters for policies in order to be effective in reinforcement learning (RL) settings. However, such methods are often quite expensive and incur large variance estimates of the gradient of the performance function. We provide a detailed analysis of various approaches to accelerating the convergence of policy gradient to an ϵ-stationary point of the optimal policy, as well as giving a high level discussion of the proof techniques for deriving the bounds relating sample complexity and convergence.

1 Introduction

In this paper, we'll analyze the following previous works: Sample Efficient Policy Gradient Methods with Recursive Variance Reduction (Pan Xu, Felicia Gao & Quanquan Gu), Momentum-Based Policy Gradient Methods (Feihu Huang, Shangqian Gao, Jian Pei & Heng Huang), and Policy Gradient Method For Robust Reinforcement Learning (Yue Wang, Shaofeng Zou). We'll refer to these papers as SRVR-PG, MBPG, and RPG respectively.

1.1 RL Background

Through this project, we'll be mainly considering infinite horizon MDPs, i.e $\mathcal{M} = \{S, A, \mathcal{P}, r, \gamma, \rho\}$, where S is the state space, A is the action space, $\mathcal{P}$ is the transition function such that $\mathcal{P}(s'|s,a)$ is the probability of transitioning to a state s' given the previous state s and action a. r represents a reward function, such that $r(s,a) \in [-R, R]$, where R is a constant. γ represents the discount factor, and $\gamma \in (0,1)$, and ρ represents the distribution over the initial state s_0. Define a policy π_θ, parametrized by the parameter θ, such that $\pi_\theta : S \to A$. Moreover, τ represents a trajectory.

We want to find a policy that maximizes the total expected cumulative reward, given by $R(\tau) = \sum_{h=0}^{H-1} \gamma^h r(s_h, a_h)$, where H represents the horizon length, and $\tau = \{s_0, a_0, \dots s_{H-1}, a_{H-1}\}$. Let $J(\theta) = \mathbb{E}_\tau[R(\tau)]$. We define an stationary point of $J(\theta)$ to be $\boldsymbol{\theta}$ such that $||\nabla J(\boldsymbol{\theta})||_2 = 0$ and we will also define an ϵ-stationary point of $J(\theta)$ to be $\boldsymbol{\theta}$ such that $\mathbb{E}[||\nabla J(\boldsymbol{\theta})||_2^2] \leq \epsilon$. Moreover, assuming the Markov property holds, we'll define $p(\tau|\theta)$, the probability of observing a given trajectory τ given the policy parameters θ to be equal to $\rho(s_0)\Pi_{h=0}^{H-1}\pi_\theta(a_h|s_h)\mathcal{P}(s_{h+1}|s_h, a_h)$.

1.2 Policy Gradient Background

Given a performance function $J(\theta)$, it is often not possible to calculate $\nabla J(\theta)$ exactly. Since $J(\theta) = \mathbb{E}_\tau[R(\tau)]$, we can say that $\nabla J(\theta)$

$= \nabla_\theta \mathbb{E}_\tau[R(\tau)] = \nabla_\theta \int_\tau R(\tau) p(\tau|\theta)\, d\tau = \int_\tau R(\tau) \nabla_\theta p(\tau|\theta)\, d\tau$

$= \int_\tau R(\tau)(\nabla_\theta p(\tau|\theta)/p(\tau|\theta)) p(\tau|\theta)\, d\tau = \mathbb{E}_{\tau \sim p}[\nabla_\theta \log p(\tau|\theta) R(\tau)]$

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Note that we don't know the distribution p so we can't directly calculate this expectation. However, we can estimate the gradient through using a batch of trajectories $\tau_1, \ldots, \tau_N$, where our estimated gradient $\hat{\nabla}_\theta J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log p(\tau_i|\theta) R(\tau_i)$. Since $\nabla_\theta \log p(\tau_i|\theta) = \nabla_\theta \log (\rho(s_0)\Pi_{h=0}^{H-1}\pi_\theta(a_h|s_h)\mathcal{P}(s_{h+1}|s_h, a_h)) = \nabla_\theta \sum_{h=0}^{H-1} \log \pi_\theta(a_h^i|s_h^i) = \sum_{h=0}^{H-1} \nabla_\theta \log \pi_\theta(a_h^i|s_h^i)$, and by using the definition of $R(\tau_i)$, we can rewrite our gradient estimate to be $\frac{1}{N}\sum_{i=1}^{N}(\sum_{h=0}^{H-1} \nabla_\theta \log \pi_\theta(a_h^i|s_h^i))(\sum_{h=0}^{H-1} \gamma^h r(s_h^i, a_h^i))$. We will denote $g(\tau_i|\theta) = (\sum_{h=0}^{H-1} \nabla_\theta \log \pi_\theta(a_h^i|s_h^i))(\sum_{h=0}^{H-1} \gamma^h r(s_h^i, a_h^i))$.

2 Robust Policy Gradient Analysis

2.1 Algorithm Overview

In this paper, the first robust policy gradient method was developed, which was proven to convergence to an ϵ optimal policy in $\mathcal{O}(\epsilon^{-3})$ complexity. In this paper, we consider the following MDP: (S, A, P, c, γ), where S, A are defined as usual, P is a transition kernel, where p_s^a is the distribution over the next states given s, a, γ is the discount factor, and c is the cost function, $0 \leq c(s, a) \leq 1$.

Define a sequence of transition kernels $\kappa = (P_0, P_1, \ldots)$. The robust value function is defined as $V^\pi(s) = \max_\kappa \mathbb{E}_\kappa[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t)|s_0 = s, \pi]$. Since we are trying to minimize the cumulative discounted cost, the robust value function is the worst case discounted cost with policy π and starting state s. Now, let the performance function $J_\rho(\pi) = \mathbb{E}_{s\sim\rho}[V^\pi(s)]$. In robust policy gradient, we aim to find θ such that the performance function is minimized. In other words, we are trying to solve the following optimization problem: $\text{argmin}_{\theta\in\Theta} J_\rho(\theta)$.

Assume that the policy class Π_θ is differentiable and k_π Lipschitz continuous. If this assumption holds, then we can say that $\psi_\rho(\theta) = \frac{\gamma R}{(1-\gamma)(1-\gamma+\gamma R)}\sum_{s\in S} d_{s\theta}^{\pi_\theta}(s)\sum_{a\in A}\nabla\pi_\theta(a|s)Q^{\pi_\theta}(s, a) + \frac{1}{1-\gamma+\gamma R}\sum_{s\in S} d_\rho^{\pi_\theta}(s)\sum_{a\in A}\nabla\pi_\theta(a|s)Q^{\pi_\theta}(s, a)$. Shown below is the pseudocode for robust policy gradient:

for $t = 0, \ldots, T-1$ **do**
  $\theta_{t+1} = \Pi_\Theta(\theta_t - \alpha_t\psi_\mu(\theta_t))$

return θ_T

Note that for all most all $\theta \in \Theta$, $J_\rho(\theta)$ is differentiable, which implies that $\psi_\rho(\theta) = \nabla J_\rho(\theta)$. Therefore, for all differentiable θ, we are performing a gradient descent update, and projecting the result on to the parameter space Θ.

Now, let's consider a version of the previous algorithm, known as smoothed robust policy gradient. We optimize over a smoothed, differentiable approximation of J_ρ denoted by $J_{\sigma,\rho}$ for some $\sigma > 0$. Instead of the maximum, we'll use the LogSumExp (LSE) operator instead, where LSE(σ, V) = $\frac{1}{\sigma}$ log $(\sum_{i=1}^{d} e^{\sigma V(i)})$, where $V \in \mathbb{R}^d$. Define $Q_\sigma^\pi(s, a) = c(s, a) + \gamma(1-R)\sum_{s'\in S} p_{s,s'}^a V_\sigma^\pi(s') + \gamma R$ LSE (σ, V_σ^π), and define $V_\sigma^\pi(s)$ analogously. Let $J_\sigma(\theta) = \sum_{s\in S}\rho(s)V_\sigma^\pi(s)$. Shown below is the pseudocode for smoothed robust policy gradient:

for $t = 0, \ldots, T-1$ **do**
  $\theta_{t+1} = \Pi_\Theta(\theta_t - \alpha_t\nabla J_\sigma(\theta_t))$
return θ_T

. Note that we can compute that $\nabla J_\sigma(\theta) = B(\rho, \theta) + \frac{\gamma R\sum_{s\in S} e^{\sigma V_\sigma^{\pi_\theta}(s)} B(s,\theta)}{(1-\gamma)\sum_{s\in S} e^{\sigma V_\sigma^{\pi_\theta}(s)}}$, where $B(s, \theta) = \frac{1}{1-\gamma+\gamma R}\sum_{s'\in S} d_s^\pi(s')\sum_{a\in A}\nabla\pi_\theta(a|s')Q_\sigma^{\pi_\theta}(s', a)$. The authors then claim the following theorem regarding this smoothed objective J_σ: $J_\sigma(\theta) - J_\sigma^* \leq C_{PL}\max_{\hat{\pi}\in(\triangle(|A|))^{|S|}}\langle\pi_\theta - \hat{\pi}, \nabla J_\sigma(\theta)\rangle + (\frac{\gamma R}{1-\gamma})\frac{2\log|S|}{\sigma}$, for some constant C_{PL}. This theorem really illustrates that using the smoothed objective, if the gradient $\nabla J_\sigma(\theta)$ is near zero (or small), then $J_\sigma(\theta)$ is within $\mathcal{O}(\frac{1}{\sigma})$ of the optimal value J^*_σ. Choosing larger values of σ in this case and make this upper bound go to zero, showing optimality.

2.2 Convergence Analysis

Assume that the policy class Π_Θ is l_π smooth, and assume that $||\nabla J_\sigma(\theta_1) - \nabla J_\sigma(\theta_2)|| \leq L_\sigma ||\theta_1 - \theta_2||$. Now, we have the following convergence theorem:

For any $\epsilon > 0$, set $\sigma = \frac{2\gamma R \cdot log|S|}{\epsilon(1-\gamma)}$ and $T = \frac{64|S|C^2_{PL}L_\sigma C_\sigma}{\epsilon^2}$, then $\min_{t \leq T-1} J(\theta_t) - J^* \leq 3\epsilon$. This means that convergence to an ϵ optimum happens in $\mathcal{O}(\epsilon^{-3})$ steps. Now, let's prove this result. We'll summarize the results of this proof, due to this proof being quite involved.

We'll denote the gradient mapping as $G^\alpha(\theta) = \frac{1}{\alpha}(\theta - \Pi_{(\triangle A)^{|S|}}(\theta - \alpha J_\sigma(\theta)))$, and set all $\alpha_t = \frac{1}{L_\sigma}$. Now, we can note that $\min_{0 \leq t \leq T-1} ||G^\alpha(\theta_t)|| \leq \sqrt{\frac{2L_\sigma(J_\sigma(\theta_0) - J^*_\sigma)}{T}}$.

Now, we can bound by $\min_{t \leq T-1} J_\sigma(\theta_t) - J^*_\sigma \leq C_{PL} \max_{\hat{\pi} \in (\triangle(|A|))^{|S|}} \langle \pi_{\theta_W} - \hat{\pi}, \nabla J_\sigma(\theta_W) \rangle + (\frac{\gamma R}{1-\gamma})(\frac{2log|S|}{\sigma})$, where $W = 1 + \text{argmin}_{t \leq T-1} ||G^\alpha(\theta_{W-1})||$.

Through bounding $\max_{\hat{\pi} \in (\triangle(|A|))^{|S|}} \langle \pi_{\theta_W} - \hat{\pi}, \nabla J_\sigma(\theta_W) \rangle \leq \sup_{\hat{\pi} \in (\triangle(|A|))^{|S|}} ||\hat{\pi} - \pi_{\theta_W}|| \frac{\epsilon}{2\sqrt{|S|}C_{PL}}$ and since $||\pi_{\theta_1} - \pi_{\theta_2}|| \leq 2\sqrt{|S|}$, our bound becomes $\frac{\epsilon}{C_{PL}}$. This implies that $\min_{t \leq T-1} J_\sigma(\theta_t) - J^*_\sigma \leq \epsilon + (\frac{\gamma R}{1-\gamma})(\frac{2log|S|}{\sigma})$. Setting $\sigma = \frac{2log|S|(\frac{\gamma R}{1-\gamma})}{\epsilon}$ and $T = \frac{64|S|C^2_{PL}L_\sigma C_\sigma}{\epsilon^2}$ gives the upper bound of 3ϵ.

3 SRVR-PG Analysis

3.1 SRVR-PG Algorithm Discussion

Using a batch of trajectories to approximate the gradient $\nabla J(\theta)$ can still result in a large variance. To address this limitation, the SRVR-PG algorithm was developed. This algorithm consists of S epochs, denoted by $s = 0, \ldots, S-1$. Denote the parameter of a reference policy to be $\tilde{\theta}^0 = \theta_0$ at the initialization of this algorithm. For each iteration s, N trajectories $\tau_1, \ldots, \tau_N$ are sampled from the reference policy $\pi_{\tilde{\theta}^s}$. Then, using these trajectories, we'll compute a gradient estimator $v^s_0 = \frac{1}{N}\sum_{i=1}^N g(\tau_i|\tilde{\theta}^s)$.

We will now define a function $\mathcal{P}_\Theta(\theta)$, which we'll use to update the reference policy parameter θ^s. The update will be of the following form: $\theta^{s+1}_{t+1} = \mathcal{P}_\Theta(\theta^{s+1}_t + \eta v^{s+1}_t)$, where v^{s+1}_t is our gradient direction. We let $\mathcal{P}_\Theta(\theta) = \text{argmin}_{u \in \mathbb{R}^d}[\mathbb{1}_\Theta(u) + \frac{1}{2\eta}||\theta - u||^2_2]$, where $\mathbb{1}_\Theta(u)$ is an indicator function representing whether u is in the parameter set Θ. We do the update $\theta^{s+1}_{t+1} = \mathcal{P}_\Theta(\theta^{s+1}_t + \eta v^{s+1}_t)$ for $m-1$ epochs. Let t represent these epochs, $t = 1, \ldots, m-1$. In each epoch we sample B trajectories $\tau_1, \ldots, \tau_B$, and do the following gradient update: $v^{s+1}_t = v^{s+1}_{t-1} + \frac{1}{B}\sum_{j=1}^B(g(\tau_j|\theta^{s+1}_t) - g_\omega(\tau_j|\theta^{s+1}_{t-1}))$, and then do the update with $\mathcal{P}$ to get the next parameter θ^{s+1}_{t+1}. After we run the above procedure for all $m-1$ epochs, to get an iterate θ^{s+1}_m. We then set our reference policy parameter $\tilde{\theta}^{s+1} = \theta^{s+1}_m$. Shown below is the pseudocode for the algorithm:

$\tilde{\theta}^0 = \theta_0 \in \Theta$
for $s = 0, \ldots, S-1$ **do**
  $\theta^{s+1}_0 = \tilde{\theta}^s$
  Sample trajectories $\tau_1, \ldots, \tau_N$ from $p(\cdot|\tilde{\theta}^s)$
  $v^{s+1}_0 = \hat{\nabla} J(\hat{\theta}^s)$
  $\theta^{s+1}_1 = \mathcal{P}_\Theta(\theta^{s+1}_0 + \eta v^{s+1}_0)$

for $t = 1, \ldots, m-1$ **do**

Sample B trajectories $\tau_1, \ldots, \tau_B$ from $p(\cdot|\theta_t^{s+1})$

$v_t^{s+1} = v_{t-1}^{s+1} + \frac{1}{B}\sum_{j=1}^{B}(g(\tau_j|\theta_t^{s+1}) - g_\omega(\tau_j|\theta_{t-1}^{s+1}))$

$\theta_{t+1}^{s+1} = \mathcal{P}_\Theta(\theta_t^{s+1} + \eta v_t^{s+1})$

$\tilde{\theta}^{s+1} = \theta_m^{s+1}$

Return randomly chosen $\theta \in \{\theta_t^s\}$ for all t, s

3.2 SRVR-PG Convergence Analysis

Assume that $||\nabla_\theta \log \pi_\theta(a|s)|| \leq G$, and $||\nabla_\theta^2 \log \pi_\theta(a|s)|| \leq M$ for all s, a. Assume that $\text{Var}(g(\tau|\theta)) \leq \xi^2$ for some constant $\xi > 0$. Moreover, assume that $\text{Var}(\omega(\tau|\theta_1, \theta_2)) \leq W$, where W is finite, and $\omega(\cdot|\theta_1, \theta_2) = p(\cdot|\theta_1)/p(\cdot|\theta_2)$. Denote the gradient mapping $\mathcal{G}_n(\theta) = \frac{1}{\eta}(\mathcal{P}_\Theta(\theta + \eta\nabla J(\theta)) - \theta)$, which conceptually represents the generalized projected gradient with respect to θ.

Convergence Theorem: If the assumptions above hold, and if $\eta \leq \frac{1}{4L}$, and epoch size m plus mini-batch size B such that $B \geq \frac{72mnG^2(2G^2/M+1)(W+1)\gamma}{(1-\gamma)^2}$. Then, we get the following result:

$\mathbb{E}[||\mathcal{G}_n(\theta)||_2^2] \leq \frac{8[J(\theta^*) - J(\theta_0) - \mathbb{1}_\Theta(\theta^*) + \mathbb{1}_\Theta(\theta_0)]}{\eta S m} + \frac{6\xi^2}{N}$, where θ^* =argmax$_{\theta\in\Theta} J(\theta)$. As proving this claim is extremely involved, we will give a high level overview of the all the proofs in the report.

First, by the definition of $\mathcal{P}_\Theta$, we have that $\theta_{t+1}^{s+1} = \text{argmin}_{u\in\mathbb{R}^d} \mathbb{1}_\Theta(u) + \frac{1}{2\eta}||u - \theta_t^{s+1}||_2^2 + \langle v_t^{s+1}, u\rangle$. We can also define the respective gradient mapping to be $\tilde{\mathcal{G}}_t^{s+1} = \frac{1}{\eta}(\theta_{t+1}^{s+1} - \theta_t^{s+1}) = \frac{1}{\eta}(\mathcal{P}_\Theta(\theta_t^{s+1} - \eta v_t^{s+1}) - \theta_t^{s+1})$. Using the convexity of $\mathbb{1}_\Theta(\cdot)$, $J(\theta)$ being L-smooth, and $\Phi(\theta) = J(\theta) - \mathbb{1}_\Theta(\theta)$, we can conclude that $\Phi(\theta_{t+1}^{s+1}) \geq \Phi(\theta_t^{s+1}) - \frac{\eta}{2}||\nabla J(\theta_t^{s+1}) - v_t^{s+1}||_2^2 + \frac{\eta}{8}||\mathcal{G}_n(\theta_t^{s+1})||_2^2 - \frac{\eta}{4}||\mathcal{G}_n(\theta_t^{s+1}) - \tilde{\mathcal{G}}_t^{s+1}||_2^2 + (\frac{1}{4\eta} - \frac{L}{2})||\theta_{t+1}^{s+1} - \theta_t^{s+1}||_2^2$.

Doing some analysis on $\bar{\theta}_{t+1}^{s+1} = \text{prox}_{\eta\mathbb{1}_\Theta}(\theta_t^{s+1} + \eta\nabla J(\theta_t^{s+1}))$ yields that $\Phi(\theta_{t+1}^{s+1}) \geq \Phi(\theta_t^{s+1}) - \frac{3\eta}{4}||\nabla J(\theta_t^{s+1}) - v_t^{s+1}||_2^2 + \frac{\eta}{8}||\mathcal{G}_n(\theta_t^{s+1})||_2^2 + (\frac{1}{4\eta} - \frac{L}{2})||\theta_{t+1}^{s+1} - \theta_t^{s+1}||_2^2$. Now, when further expanding out $||\nabla J(\theta_t^{s+1}) - v_t^{s+1}||_2^2$ and take expectations, we get that $\mathbb{E}[||\nabla J(\theta_t^{s+1}) - v_t^{s+1}||_2^2] \leq \frac{1}{B^2}\sum_{j\in\mathcal{B}_t}\mathbb{E}[||g_\omega(\tau_j|\theta_{t-1}^{s+1}) - g(\tau_j|\theta_t^{s+1})||_2^2] + ||\nabla J(\theta_{t-1}^{s+1}) - v_{t-1}^{s+1}||_2^2$, where $\mathcal{B}_t$ denotes the minibatch, and B represents its' size.

Through some more analysis, we can show that $\mathbb{E}[||\nabla J(\theta_t^{s+1}) - v_t^{s+1}||_2^2] \leq \frac{C_\gamma}{B}\sum_{l=1}^{t}||\theta_l^{s+1} - \theta_{l-1}^{s+1}||_2^2 + ||\nabla J(\theta_0^{s+1}) - v_0^{s+1}||_2^2$, for a constant C_γ. Now, using the variance assumption and some of the previous results, we can say that $\mathbb{E}_{N,B}[\Phi(\theta_m^{s+1})] \geq \mathbb{E}_{N,B}[\Phi(\theta_0^{s+1})] + \frac{\eta}{8}\sum_{t=0}^{m-1}\mathbb{E}_N[||\mathcal{G}_n(\theta_t^{s+1})||_2^2] - \frac{3m\eta\xi^2}{4N} + (\frac{1}{4\eta} - \frac{L}{2} - \frac{3m\eta C_\gamma}{2B})\sum_{t=0}^{m-1}||\theta_{t+1}^{s+1} - \theta_t^{s+1}||_2^2$. Setting η and B appropriately, we get the following result:

$\mathbb{E}[||\mathcal{G}_n(\theta)||_2^2] \leq \frac{6\xi^2}{N} + \frac{8(\Phi(\theta^*) - \Phi(\theta_0))}{\eta S m}$, which concludes the proof.

3.3 Extension to Parameter Based Exploration

This paper also discusses how to extend SRVR-PG to a policy gradient algorithm using parameter-based exploration (PGPE) as well. It assumes that the parameter θ follows a set prior distribution ρ.

Given this parameter ρ, we define the performance function to be $J(\rho) = \int_\theta\int_\tau p(\theta|\rho)p(\tau|\theta)R(\tau)d\tau d\theta$. The goal of PGPE is to find the prior distribution ρ^* such that $\rho^* = \text{argmax}_\rho J(\rho)$. A linear deterministic policy π is chosen of the following form: $\pi_\theta(a|s) = \delta(a - \theta^T s)$, and the distribution update rule at every step is given by $\rho_{t+1} = \rho_t + \eta\nabla_{\rho_t} J(\rho_t)$. The gradient estimates are similar to those of SRVR-PG. Shown below is the pseudocode of this algorithm.

$\tilde{\theta}^0 = \theta_0 \in \Theta$

for $s = 0, \ldots, S-1$ **do**
    $\rho_0^{s+1} = \rho^s$
    Sample N policy parameters $\theta_1, \ldots, \theta_N$ from $p(\cdot|\rho^s)$
    Sample one trajectory τ_i from each policy π_{θ_i}
    $v_0^{s+1} = \hat{\nabla} J_\rho(\rho^s)$
    $\rho_1^{s+1} = \rho_0^{s+1} + \eta v_0^{s+1}$
    for $t = 1, \ldots, m-1$ **do**
        Sample B trajectories $\tau_1, \ldots, \tau_B$ from $p(\cdot|\theta_t^{s+1})$, sample τ_j from each π_{θ_j}
        $v_t^{s+1} = v_{t-1}^{s+1} + \frac{1}{B}\sum_{j=1}^{B}(g(\tau_j|\rho_t^{s+1}) - g_\omega(\tau_j|\rho_{t-1}^{s+1}))$
        $\rho_{t+1}^{s+1} = \rho_t^{s+1} + \eta v_t^{s+1}$
Return randomly chosen $\rho \in \{\rho_t^s\}$ for all t, s

3.4 Results

The SRVR-PG algorithm was evaluated on many classic reinforcement learning environments, and the results of these experiments are displayed in Figure 1. From these figures, we can see that SRVR-PG overall has higher returns than the other estimators, including SVRPG and GPOMDP.

4 Momentum-Based Policy Gradient Methods

4.1 Brief Background and Motivation

In general, RL optimizes a policy $\pi_\theta(a|s)$, parameterized by $\theta \in \mathbb{R}^d$, to maximize the expected cumulative reward $J(\theta)$: $J(\theta) = \mathbb{E}_{\tau\sim p(\tau|\theta)}[R(\tau)] = \int R(\tau)p(\tau|\theta)d\tau$, where $\tau = (s_0, a_0, s_1, \ldots)$ represents a trajectory, and $p(\tau|\theta)$ is its distribution. While policy gradient methods like REINFORCE provide an effective framework by estimating $\nabla J(\theta)$ using stochastic approximations: $\nabla J(\theta) = \mathbb{E}_{\tau\sim p(\tau|\theta)}[\nabla \log p(\tau|\theta)R(\tau)]$, they suffer from high variance in gradient estimates, often resulting in slow convergence and suboptimal performance.

Prior in variance reduction techniques, such as SVRG and SPIDER, have addressed this challenge in supervised learning but require adaptation for non-stationary RL problems, where $p(\tau|\theta)$ changes with θ. Methods like HAPG and SRVR-PG use recursive gradient updates to achieve better sample complexity, reducing the number of trajectories required to find an ϵ-stationary point of $J(\theta)$. However, these methods often rely on large batch sizes and intricate learning rate schedules, limiting their practical efficiency.

To overcome these limitations, this paper introduces momentum-based policy gradient methods—IS-MBPG and HA-MBPG—which incorporate momentum-driven updates to stabilize gradient estimation and adapt learning rates dynamically. Momentum is a technique used in optimization to stabilize and accelerate convergence. In the context of policy gradient methods, it involves combining the current gradient estimate with a fraction of the previous update to reduce variance and smooth the optimization path. Formally, the momentum update can be represented as $u_t = \beta_t g_t + (1-\beta_t)u_{t-1}$, where g_t is the current gradient, u_t is the momentum-adjusted gradient, and β_t controls the influence of the current and past gradients. This approach helps mitigate noise in stochastic updates and enhances sample efficiency. These methods achieve the optimal sample complexity of $\mathcal{O}(\epsilon^{-3})$ without requiring large batches. Importantly, they maintain efficiency by updating parameters using only a single trajectory at each iteration, offering a significant advancement in reinforcement learning optimization.

4.2 IS-MBPG Algorithm Discussion and Theoretical Setup

The Important-Sampling Momentum-Based Policy Gradient algorithm combines momentum-based updates with importance sampling. At its core, the algorithm seeks to reduce the variance of stochastic gradients while maintaining computational efficiency. This is achieved by leveraging a momentum-driven variance reduction mechanism incorporating importance sampling to correct for non-stationarity in the underlying trajectory distribution $p(\tau|\theta)$.

The main optimization goal is to maximize the expected cumulative reward $J(\theta)$: $J(\theta) = \mathbb{E}_{\tau\sim p(\tau|\theta)}[R(\tau)]$ where the gradient of $J(\theta)$ is approximated using the stochastic gradient:

$\hat{\nabla} J(\theta) = \frac{1}{|B|}\sum_{\tau\in B} g(\tau,\theta)$, and $g(\tau,\theta)$ is defined as: $g(\tau,\theta) = \sum_{h=0}^{H-1}\nabla_\theta \log \pi_\theta(a_h|s_h)\cdot \sum_{h=0}^{H-1}\gamma^h R(s_h,a_h)$, where π_θ is the policy, γ is the discount factor, and H is the horizon length.

The IS-MBPG algorithm refines the above gradient approximation by introducing a momentum-based update: $u_t = \beta_t g(\tau_t|\theta_t) + (1-\beta_t)\left[u_{t-1} + g(\tau_t|\theta_t) - w(\tau_t|\theta_{t-1},\theta_t) g(\tau_t|\theta_{t-1})\right]$, where $w(\tau_t|\theta_{t-1},\theta_t)$ is the importance sampling weight: $w(\tau_t|\theta_{t-1},\theta_t) = \frac{p(\tau_t|\theta_{t-1})}{p(\tau_t|\theta_t)} = \prod_{h=0}^{H-1}\frac{\pi_{\theta_{t-1}}(a_h|s_h)}{\pi_{\theta_t}(a_h|s_h)}$ The weight $w(\tau_t|\theta_{t-1},\theta_t)$ adjusts for the change in policy parameters between successive iterations, ensuring unbiased gradient estimates.

One of the key advantages of IS-MBPG is its ability to dynamically adapt learning rates based on the accumulated gradients. The learning rate η_t is computed as: $\eta_t = \frac{k}{\left(m+\sum_{i=1}^{t} G_i^2\right)^{1/3}}$, where $G_t = \|g(\tau_t,\theta_t)\|$ captures the magnitude of the gradient at each iteration. This adaptive learning rate ensures that the algorithm remains stable and converges efficiently, particularly in high-variance settings.

The momentum-based update in IS-MBPG reduces the variance of the gradient estimator through a combination of the several SGD variant techniques. When $\beta_t = 0$, the algorithm focuses on variance reduction. Conversely, when $\beta_t = 1$, the algorithm performs like SGD, prioritizing simplicity and speed. This flexibility allows IS-MBPG to balance exploration and exploitation dynamically. The IS-MBPG algorithm achieves the optimal sample complexity of $\mathcal{O}(\epsilon^{-3})$ for finding an ϵ-stationary point of the performance function $J(\theta)$. This improvement is achieved without relying on large batch sizes or double-loop structures, making IS-MBPG computationally efficient and scalable. Additionally, the algorithm only requires a single trajectory per iteration, significantly reducing the overhead compared to other variance-reduced policy gradient methods.

We present the convergence properties of the IS-MBPG algorithm. The theoretical guarantees of IS-MBPG rely on several key assumptions and propositions that establish its sample complexity and performance bounds. First make the following assumptions: The gradient and Hessian matrix of the function $\log \pi_\theta(a|s)$ are assumed to be bounded. That is, there exist constants M_g and $M_h > 0$ such that: $\|\nabla_\theta \log \pi_\theta(a|s)\| \le M_g, \quad \|\nabla_\theta^2 \log \pi_\theta(a|s)\| \le M_h$.. Secondly the variance of the stochastic gradient $g(\tau|\theta)$ is bounded. Specifically, there exists a constant $\sigma > 0$ such that for all π_θ: $\mathbb{V}(g(\tau|\theta)) = \mathbb{E}\|g(\tau|\theta) - \nabla J(\theta)\|^2 \le \sigma^2$. Finally, the variance of the importance sampling weight $w(\tau|\theta_1,\theta_2) = p(\tau|\theta_1)/p(\tau|\theta_2)$ is bounded. That is there exists a constant $W > 0$ such that: $\mathbb{V}(w(\tau|\theta_1,\theta_2)) \le W, \quad \forall \theta_1,\theta_2 \in \mathbb{R}^d$, $\tau \sim p(\tau|\theta_2)$ Note these assumptions are standard in the convergence analysis of policy gradient methods as seen in the previous paper with similar assumptions and ensure that the stochastic gradient and importance sampling weights remain stable.

We now present a few propositions. The first is Lipschitz Properties and Smoothness. Suppose $g(\tau|\theta)$ is the policy gradient estimator. Under the boundedness assumptions, the following properties hold: The stochastic gradient $g(\tau|\theta)$ is L-Lipschitz differentiable, i.e., $\|g(\tau|\theta) - g(\tau|\theta')\| \le L\|\theta - \theta'\|$, where $L = \frac{M_h R}{(1-\gamma)^2}$. Next, the performance function $J(\theta)$ is L-smooth, i.e., $\|\nabla^2 J(\theta)\| \le L$. And lastly the stochastic gradient $g(\tau|\theta)$ is bounded, i.e., $\|g(\tau|\theta)\| \le G$ for all $\theta \in \mathbb{R}^d$, where $G = \frac{M_g R}{(1-\gamma)^2}$.

The next proposition is a bounded Hessian: Under the same assumptions, for all θ, the Hessian estimator satisfies: $\|\nabla^2(\theta,\tau)\|^2 \le \frac{H^2 M_g^4 R^2 + M_h^2 R^2}{(1-\gamma)^4} = \tilde{L}^2$. This implies that $J(\theta)$ is $\tilde{L}$-smooth.

4.3 Convergence Analysis of IS-MBPG

Theorem 4: Convergence of IS-MBPG. Let $\{\theta_t\}_{t=1}^T$ be the sequence generated by the IS-MBPG algorithm. Set $k = \mathcal{O}(G^{2/3}/L)$, $c = G^2/(3k^3 L) + 104B^2$, $m = \max\{2G^2, (2Lk)^3, (ck^2/L)^3\}$, and $\eta_0 = k/m^{1/3}$. Then, we have:

$$\mathbb{E}\|\nabla J(\theta_\zeta)\| \le \frac{\sqrt{2\Omega m^{1/6}}}{\sqrt{T}} + \frac{2\Omega^{3/4}}{\sqrt{T}} + \frac{2\sqrt{\Omega}\sigma^{1/3}}{T^{1/3}},$$

where $\Omega = \frac{1}{k}\left(16(J^* - J(\theta_1)) + \frac{m^{1/3}\sigma^2}{8B^2 k} + \frac{c^2 k^3}{4B^2}\ln(T+2)\right)$ and $J^* = \sup_\theta J(\theta) < \infty$.

We now present the proof sketch for this. First define two lemmas which will be used in the proof:

Lemma 3. Under Assumption 1, let $e_t = \nabla J(\theta_t) - u_t$ Given: $0 < \eta_t \le \frac{1}{2L}$ for all $t \ge 1$ we have

$$E[J(\theta_{t+1})] \ge E\left[J(\theta_t) - \frac{3\eta_t}{4}\|e_t\|^2 + \frac{\eta_t}{8}\|\nabla J(\theta_t)\|^2\right].$$

Lemma 4. Variance Dynamics of the Stochastic Gradient. Assume that the stochastic policy gradient u_t is generated by Algorithm 1, and let $e_t = u_t - \nabla J(\theta_t)$. Then the following inequality holds:

$$\mathbb{E}\left[\eta_{t-1}^{-1}\|e_t\|^2\right] \le 2\beta_t^2\eta_{t-1}^{-1}G_t^2 + \eta_{t-1}^{-1}(1-\beta_t)^2\left(1 + 8\eta_{t-1}^2B^2\right)\mathbb{E}\|e_{t-1}\|^2 + 8(1-\beta_t)^2B^2\eta_{t-1}\|\nabla J(\theta_{t-1})\|^2,$$

where $B^2 = L^2 + 2G^2C_w^2$ and $C_w = \sqrt{H(2HM_g^2 + M_h)(W+1)}$. The proof for this expands $\mathbb{E}[\eta_{t-1}^{-1}$ The proof of this lemma comes down to bounding $E\eta_{t-1}^{-1}\|k_{etk}\|^2 \le 2\eta_{t-1}^{-1}(1-\beta_t)^2E\|k_{et-1}\|^2 + 2\beta_t^2\eta_{t-1}^{-1}G_t^2 + 2(1-\beta_t)^2\eta_{t-1}^{-1}E\|g(\tau_t|\theta_t) - w(\tau_t|\theta_{t-1},\theta_t)g(\tau_t|\theta_{t-1})\|$. Denote the last expectation as T_1. Further bounding on T_1 yields $T_1 \le 2(L^2 + 2G^2C_w^2)\|\theta_t - \theta_{t-1}\|^2$.

We now cover the main convergence proof of Theorem 4 in the paper. Assume the given information in the start of the theorem. We first do parameter initialization and provide bounds on η_t and β_t.

We begin by noting that $m \ge (2Lk)^3$ ensures $\eta_t \le k/m^{1/3} \le 1/2L$. Similarly, $m \ge (ck/2L)^3$ ensures $\beta_{t+1} = c\eta_t^2 \le c\eta_t/2L \le \frac{ck}{2Lm^{1/3}} \le 1$. Thus, the learning rate η_t and momentum parameter β_t remain valid for all t.

From Lemma 4, the error term $e_t = u_t - \nabla J(\theta_t)$ satisfies: $\mathbb{E}\left[\eta_{t-1}^{-1}\|e_t\|^2 - \eta_{t-2}^{-1}\|e_{t-1}\|^2\right] \le \mathbb{E}\left[2\beta_t^2\eta_{t-1}^{-1}G_t^2 + \eta_{t-1}^{-1}(1-\beta_t)^2(1+8\eta_{t-1}^2B^2)\|e_{t-1}\|^2 + 8(1-\beta_t)^2B^2\eta_{t-1}\|\nabla J(\theta_{t-1})\|^2\right]$.

After we proceed with refining terms and bounding $\eta_{t-1}^{-1} - \eta_{t-2}^{-1}$. The term $\eta_{t-1}^{-1} - \eta_{t-2}^{-1}$ is bounded by exploiting the concavity of $x^{1/3}$. Specifically: $\eta_{t-1}^{-1} - \eta_{t-2}^{-1} \le \frac{G_t^2}{3k(m+\sum_{i=1}^{t-1}G_i^2)^{2/3}} \le \frac{G_t^2}{3k(m+\sum_{i=1}^{t}G_i^2)^{2/3}}$.. Using $m \ge 2G^2$, this implies: $\eta_{t-1}^{-1} - \eta_{t-2}^{-1} \le \frac{G_t^2}{3k^3L}\eta_t$.

Next we bound on the auxiliary terms. Substituting the previous result into the upper bound of the error dynamics: $T_2 = \left(\eta_{t-1}^{-1} - \eta_{t-2}^{-1} + 8B^2\eta_t - \beta_t\eta_{t-1}^{-1} - 8\eta_t\beta_tB^2\right)\|e_t\|^2$. Using $c = \frac{G^2}{3k^3L} + 104B^2$, we find: $T_2 \le -96B^2\eta_t\|e_t\|^2$.

Now define a Lyapunov function, a scalar function that is used in control theory to prove the stability of an equilibrium of a first order ode. $\Phi_t = J(\theta_t) - \frac{1}{128B^2\eta_{t-1}}\|e_t\|^2$. Using Lemma 3, the analysis yields: $\mathbb{E}[\Phi_{t+1} - \Phi_t] \ge \mathbb{E}\left[-\frac{c\eta_t^3G_{t+1}^2}{64B^2} + \frac{\eta_t}{16}\|\nabla J(\theta_t)\|^2\right]$. Summing over $t = 1$ to T, this implies: $\sum_{t=1}^{T}\mathbb{E}[\eta_t\|\nabla J(\theta_t)\|^2] \le \mathbb{E}\left[16(J^* - J(\theta_1)) + \frac{1}{8B^2\eta_0}\|e_1\|^2 + \frac{c^2k^3}{4B^2}\ln(T+2)\right]$.

Applying the Cauchy-Schwarz inequality, we obtain: $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla J(\theta_t)\| \le \sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla J(\theta_t)\|^2}$. Combining with the above, and using Ω as defined, we conclude: $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla J(\theta_t)\| \le \sqrt{2\Omega m^{1/6}} + \frac{2\Omega^{3/4}}{\sqrt{T}} + \frac{2\sqrt{\Omega}\sigma^{1/3}}{T^{1/3}}$. This establishes the theorem.

Remark 1: Sample Complexity. Since $\Omega = \mathcal{O}(\ln T)$, Theorem 1 shows that the IS-MBPG algorithm achieves an $\mathcal{O}(\sqrt{\ln T}/T^{1/3})$ convergence rate. The algorithm requires a single trajectory per iteration, and the total sample complexity is $\mathcal{O}(\epsilon^{-3})$ to achieve an ϵ-stationary point, where $T = \epsilon^{-3}$.

4.4 HA-MBPG Algorithm Discussion and Analysis

The Hessian-Aided Momentum-Based Policy Gradient (HA-MBPG) algorithm builds upon the momentum-based updates of IS-MBPG while introducing a Hessian-aided technique to further refine the gradient estimates with the same theoretical assumptions as before. The core idea is to leverage second-order information, approximated through the Hessian of the policy's performance function $J(\theta)$, to improve convergence speed and reduce variance in the stochastic gradient estimates.

The optimization objective remains the maximization of the expected cumulative reward: $J(\theta) = \mathbb{E}_{\tau\sim p(\tau|\theta)}[R(\tau)]$. The HA-MBPG algorithm incorporates the Taylor expansion to approximate the difference in gradients $\nabla J(\theta_t)$ and $\nabla J(\theta_{t-1})$ as: $\nabla J(\theta_t) - \nabla J(\theta_{t-1}) \approx \left[\int_0^1 \nabla^2 J(\theta_{t-1} + \alpha(\theta_t - \theta_{t-1}))d\alpha\right](\theta_t - \theta_{t-1})$, where $\nabla^2 J(\theta)$ represents the Hessian matrix of $J(\theta)$. This term captures the curvature of the performance function and provides additional information to guide the update process.

To operationalize this, the algorithm constructs an unbiased estimate of the gradient difference using the sampled trajectory τ and an auxiliary random variable $\alpha \sim U[0,1]$: $\Delta_t = \nabla^2 J(\theta_{t-1} + \alpha(\theta_t - \theta_{t-1}))(\theta_t - \theta_{t-1})$. In practice, the exact computation of $\nabla^2 J(\theta)$ is computationally expensive and thus intractable. Instead, HA-MBPG approximates the Hessian-vector product $\nabla^2 J(\theta)v$ using a finite-difference method from numerical PDE solving methods: $\nabla^2 J(\theta)v \approx \frac{\nabla J(\theta+\delta v) - \nabla J(\theta - \delta v)}{2\delta}$, where $\delta > 0$ is a small scalar and $v = \theta_t - \theta_{t-1}$. This approximation balances computational efficiency with accuracy, making the Hessian-aided updates feasible for practical implementation.

The gradient update rule in HA-MBPG incorporates both momentum and the Hessian-aided term: $u_t = \beta_t w(\tau|\theta_t, \theta_{t-1}) g(\tau|\theta_t) + (1-\beta_t)\left[u_{t-1} + \Delta_t\right]$, where $w(\tau|\theta_t, \theta_{t-1})$ is the importance sampling weight that adjusts for changes in policy parameters: $w(\tau|\theta_t, \theta_{t-1}) = \frac{p(\tau|\theta_t)}{p(\tau|\theta_{t-1})}$.

The learning rate η_t is adaptively calculated as: $\eta_t = \frac{k}{\left(m+\sum_{i=1}^t G_i^2\right)^{1/3}}$, where $G_t = \|g(\tau_t, \theta_t)\|$ reflects the magnitude of the gradient at iteration t. This adaptive rate ensures stability and efficient convergence by scaling learning rates based on the accumulated gradients. We now provide a analysis of the convergence proof.

4.5 Convergence Analysis of HA-MBPG

We provide the following convergence theorem and the lemma used in the proof. Note the proof similarity to the first order momentum based approach first bounding recurrence of the error term using the 2nd order analog to lemma 4, lemma 5, and then defining a Lyapunov potential function, to use smoothness properties to telescope between iterative potential functions and then using Cauchy Schwartz to bound the expected gradient norm.

Lemma 5. Assume that the stochastic policy gradient u_t is generated by Algorithm 2. Let $e_t = u_t - \nabla J(\theta_t)$, then the following holds:

$$\mathbb{E}\left[\eta_{t-1}^{-1}\|e_t\|^2\right] \le 4(W+1)\beta_t^2\eta_{t-1}^{-1}G_t^2 + \eta_{t-1}^{-1}(1-\beta_t)^2\left(1+4\eta_{t-1}^2L^2\right)\mathbb{E}\|e_{t-1}\|^2 + 4(1-\beta_t)^2L^2\eta_{t-1}\|\nabla J(\theta_{t-1})\|^2.$$

Theorem 2: Convergence of HA-MBPG Algorithm. Let $\{\theta_t\}_{t=1}^T$ be the sequence generated by Algorithm 2, and set $k = \mathcal{O}(G^{2/3}/L)$, $c = \frac{G^2}{3k^3L} + 52L^2$, $m = \max\{2G^2, (2Lk)^3, (ck/2L)^3\}$, and $\eta_0 = k/m^{1/3}$. Then, the following bound holds:

$$\mathbb{E}\|\nabla J(\theta_\zeta)\| \le \sqrt{2\Lambda m^{1/6}} + \frac{2\Lambda^{3/4}}{\sqrt{T}} + \frac{2\sqrt{\Lambda}\sigma^{1/3}}{T^{1/3}},$$

where

$$\Lambda = \frac{1}{k}\left(16(J^* - J(\theta_1)) + \frac{m^{1/3}\sigma^2}{4L^2k} + \frac{(W+1)c^2k^3}{2L^2}\ln(T+2)\right),$$

and $J^* = \sup_\theta J(\theta) < \infty$.

Starting off, given $m \ge (2Lk)^3$, we have $\eta_t \le k/m^{1/3} \le 1/2L$. Similarly, $m \ge (ck/2L)^3$ ensures $\beta_{t+1} = c\eta_t^2 \le c\eta_t/2L \le \frac{ck}{2Lm^{1/3}} \le 1$. Thus, both the learning rate η_t and momentum coefficient β_t remain valid for all t. Using Lemma 5, the error term $e_t = u_t - \nabla J(\theta_t)$ satisfies the following recurrence: $\mathbb{E}\left[\eta_{t-1}^{-1}\|e_t\|^2 - \eta_{t-2}^{-1}\|e_{t-1}\|^2\right] \le \mathbb{E}\Big[4(W+1)\beta_t^2\eta_{t-1}^{-1}G_t^2 + \eta_{t-1}^{-1}(1-\beta_t)^2\left(1+4\eta_{t-1}^2L^2\right)\|e_{t-1}\|^2 + 4(1-\beta_t)^2L^2\eta_{t-1}\|\nabla J(\theta_{t-1})\|^2\Big]$. Using the concavity of $x^{1/3}$, we bound the difference $\eta_{t-1}^{-1} - \eta_{t-2}^{-1}$: $\eta_{t-1}^{-1} - \eta_{t-2}^{-1} = \frac{1}{k}\left((m+\sum_{i=1}^t G_i^2)^{1/3} - (m+\sum_{i=1}^{t-1}G_i^2)^{1/3}\right) \le \frac{G_t^2}{3k(m+\sum_{i=1}^t G_i^2)^{2/3}}$. Using $m \ge 2G^2$, we find: $\eta_{t-1}^{-1} - \eta_{t-2}^{-1} \le \frac{G_t^2}{3k^3L}\eta_t$. Substituting the above result into the error dynamics: $T_4 = \left(\eta_{t-1}^{-1} - \eta_{t-2}^{-1} + 4L^2\eta_t - \beta_t\eta_{t-1}^{-1} - 4\eta_t\beta_t L^2\right)\|e_t\|^2$. Using $c = \frac{G^2}{3k^3L} + 52L^2$, we find: $T_4 \leq -48L^2\eta_t\|e_t\|^2$.

Define the Lyapunov function $\Psi_t = J(\theta_t) - \frac{1}{64L^2\eta_{t-1}}\|e_t\|^2$. Using the smoothness of $J(\theta)$ and the refined recurrence, we derive: $\mathbb{E}[\Psi_{t+1} - \Psi_t] \geq \mathbb{E}\left[-\frac{(W+1)c^2\eta_t^3G_{t+1}^2}{32L^2} + \frac{\eta_t}{16}\|\nabla J(\theta_t)\|^2\right]$. Summing over t=1 to T, $\sum_{t=1}^{T}\mathbb{E}[\eta_t\|\nabla J(\theta_t)\|^2] \leq \mathbb{E}\left[16(J^* - J(\theta_1)) + \frac{1}{4L^2\eta_0}\|e_1\|^2 + \frac{(W+1)c^2k^3}{2L^2}\ln(T+2)\right]$. Using the Cauchy-Schwarz inequality, the squared gradient norm is bounded by: $\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla J(\theta_t)\| \leq \sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla J(\theta_t)\|^2}$. Substituting the above and simplifying using Λ: $\mathbb{E}\|\nabla J(\theta_\zeta)\| \leq \sqrt{2\Lambda m^{1/6}} + \frac{2\Lambda^{3/4}}{\sqrt{T}} + \frac{2\sqrt{\Lambda}\sigma^{1/3}}{T^{1/3}}$. Since $\Lambda = \mathcal{O}(\ln T)$, the convergence rate is $\mathcal{O}(\sqrt{\ln T}/T^{1/3})$. Setting $T = \epsilon^{-3}$, the sample complexity is $\mathcal{O}(\epsilon^{-3})$, completing the proof.

5 Algorithm Comparison

The main similarity between these papers is that they all discuss methods for finding ϵ-stationary points of a policy π_θ's objective function $J(\theta)$. These papers both describe their algorithm, as well as provide proofs of their algorithm's convergence guarantees.

In the SRVR-PG and RPG papers, the optimization relies on first-order methods, which utilize estimates of the policy gradient $\nabla J(\theta)$ without explicitly incorporating higher-order derivatives such as the Hessian $\nabla^2 J(\theta)$. For instance, the SRVR-PG method introduces a variance reduction technique by maintaining a control variate, which stabilizes gradient estimates and achieves a sample complexity of $O(\epsilon^{-3})$. This is done by iteratively estimating gradients through stochastic recursive updates, a process governed by: $\nabla_{\text{SVR}}J(\theta) = \nabla J(\theta_k) + \frac{1}{B}\sum_{i=1}^{B}\left(\nabla J_i(\theta) - \nabla J_i(\theta_k)\right)$, where B is the mini-batch size, and $\nabla J_i(\cdot)$ denotes the stochastic gradient of the i-th trajectory. This method avoids computing second-order derivatives, making it computationally attractive.

The RPG paper extends the focus to robust reinforcement learning under model mismatch. Here, the objective is to optimize the worst-case performance of a policy over an uncertainty set $\mathcal{P}$. The robust policy gradient is derived using a sub-gradient approach due to the non-differentiability of the worst-case value function: $\psi_\rho(\theta) = \sum_{s\in\mathcal{S}} d_\rho^{\pi_\theta}(s)\sum_{a\in\mathcal{A}}\nabla\pi_\theta(a|s)Q^{\pi_\theta}(s,a)$, where $d_\rho^{\pi_\theta}(s)$ is the discounted visitation distribution and $Q^{\pi_\theta}(s,a)$ incorporates the worst-case transition kernel. The RPG algorithm employs smoothing techniques to approximate the max operator, enabling efficient computation of gradients while maintaining theoretical robustness guarantees.

On the other hand, the MBPG paper incorporates second-order information by explicitly using the Hessian $\nabla^2 J(\theta)$ in its momentum-based updates. This additional information allows the algorithm to adaptively adjust the update direction, leading to accelerated convergence. The method constructs a momentum term that blends the gradient and Hessian information: $M_t = \beta M_{t-1} + \nabla^2 J(\theta_t)\nabla J(\theta_t)$, where β is the momentum parameter. This approach achieves superior theoretical convergence rates, particularly in non-convex settings, by leveraging curvature information to navigate the landscape of $J(\theta)$ more effectively. The paper proves that MBPG achieves a sample complexity of $O(\epsilon^{-3})$ in certain settings, outperforming purely first-order methods.

While all three papers share a common goal of optimizing $J(\theta)$ efficiently, their methodological choices reflect a trade-off between computational simplicity and theoretical guarantees. SRVR-PG and RPG prioritize sample efficiency and robustness using first-order gradients, while MBPG achieves faster convergence by incorporating second-order derivatives at the cost of increased computational complexity.

6 Conclusion

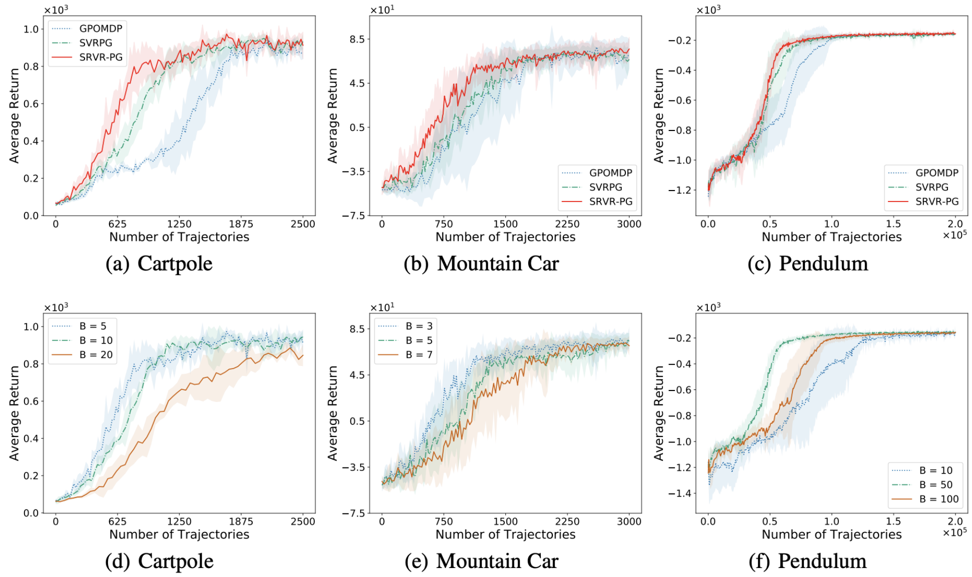
In this report, we describe three fundamental papers for efficient policy gradient algorithms for convergence to ϵ-stationary policies. We also present the convergence analyses for all of these methods, and compare and contrast these algorithms in terms of their efficiency in converging to an ϵ-stationary policy. We hope that our work provides inspiration to future analyses of policy gradient algorithms.

References

[1] Xu, Pan & Gao, Felicia & Gu, Quanquan (2020) Sample Efficient Policy Gradient Methods With Recursive Variance Reduction, *International Conference on Learning Representations*

[2] Wang, Yue & Zou, Shaofeng (2022) Policy Gradient Method For Robust Reinforcement Learning, *International Conference on Machine Learning*

[3] Huang, Feihu & Gao, Shangqian & Pei, Hian & Huang, Heng (2020) Momentum-Based Policy Gradient Methods, *International Conference on Machine Learning*



(a) Cartpole (b) Mountain Car (c) Pendulum

(d) Cartpole (e) Mountain Car (f) Pendulum

Figure 1: (a)-(c): Comparison of different algorithms. Experimental results are averaged over 10 repetitions. (d)-(f): Comparison of different batch size B on the performance of SRVR-PG.

Figure 1: SRVR-PG Experiments Results.

A Appendix / supplemental material

Shown above are the experimental results for the SRVR-PG paper.