Overcoming Sample Complexity in Policy Gradient: Variance Reduction via Defensive Importance Sampling

Chris Junchi Li[◇]

Department of Electrical Engineering and Computer Sciences[◇]
University of California, Berkeley

October 4, 2024

Abstract

Policy gradient methods are widely used in reinforcement learning due to their applicability in large-scale decision-making tasks. However, a major bottleneck for these methods is their high sample complexity, particularly when applied to complex environments with continuous action spaces. In this paper, we propose a novel variance-reduced policy gradient algorithm leveraging defensive importance sampling to improve sample efficiency. Unlike previous approaches, our method achieves an optimal sample complexity of $O(\epsilon^{-3})$ without relying on unrealistic assumptions about the variance of importance weights. We prove matching lower bounds for our algorithm, showing that it achieves state-of-the-art performance under standard assumptions. Through numerical simulations on continuous control tasks, we demonstrate the stability and efficiency of our algorithm in practical settings. Our findings offer new insights into variance reduction techniques and their impact on policy optimization.

Keywords: Variance-Reduced Policy Gradient, Reinforcement Learning, Sample Complexity, Defensive Importance Sampling, Non-Convex Optimization

1 Introduction

Policy gradient (PG) methods have gained significant traction in reinforcement learning (RL) due to their effectiveness in handling large and continuous action spaces. Their applicability extends across diverse fields, including robotics, gaming, and fine-tuning large language models. However, despite their widespread adoption, policy gradient algorithms often require a substantial number of samples to achieve acceptable performance. This is particularly problematic in high-dimensional or continuous control tasks, where sample efficiency is crucial.

To address the sample complexity challenges, recent works have focused on variance reduction techniques. Variance-reduced policy gradient (VR-PG) methods, in particular, have demonstrated improvements in reducing the number of required interactions while maintaining performance. Nonetheless, these methods often impose strong assumptions on the variance of importance weights, which limits their practical utility, especially in continuous action spaces.

In this paper, we present a novel approach to variance reduction for policy gradient methods based on defensive importance sampling. Our method circumvents the need for strong assumptions on the variance of importance weights and achieves an optimal sample complexity of $O(\epsilon^{-3})$ for non-convex policy optimization problems. Additionally, we provide a matching lower bound, demonstrating the theoretical optimality of our approach.

Background Policy gradient methods (PG) are a family of reinforcement learning algorithms [SB18] that are particularly suited for large-scale decision and control problems. Recent applications of PG algorithms range from videogames [WBK+22], to robotics [RHRH22], to large language models [Ope23, ACG+24], where fine-tuning of the LLM with these methods shows a concrete application within the current conversational artificial intelligence boom, and the potential impact of accelerated versions of these methods. In fact, the widespread adoption of these methods is still limited by their need for a huge amount of training data, meaning very long training times in the case of cyber-physical systems or massive computational resources in the case of virtual systems. This motivates the study of the sample complexity of PG from a theoretical perspective [AKLM21] and the development of accelerated algorithms from a practical one.

The sample complexity of policy gradient algorithms can be measured in terms of the number of interaction episodes, or samples, required to find an ϵ-stationary-point [PBC+18]. More formally, if $\Theta \subseteq \mathbb{R}^d$ is a space of policy parameters and $J : \Theta \to \mathbb{R}$ is an objective function measuring the performance of a policy, the goal is to find a $\theta \in \Theta$ such that

$$\|\nabla_\theta J(\theta)\| \leq \epsilon \tag{1}$$

using as few samples as possible, or at least with a number of samples that is polynomial in ϵ^{-1} and all relevant problem-dependent quantities. Alternatively, under stronger assumptions, one can study the number of samples necessary to find a globally optimal policy [LZBY20]. However, since J can be non-convex, convergence to a local optimum is the best one can hope for in full generality. In this paper, we will focus on convergence to ϵ-first-order-stationary-points (ϵ-FOSP). Furthermore, we will restrict our attention to policy gradient algorithms that use first-order information only, which are typically more computationally efficient [FBKH23].

The prototypical policy gradient algorithm is REINFORCE [Wil92] and its sample complexity is $O\left(\epsilon^{-4}\right)$. This simply follows from the sample complexity of stochastic gradient descent [BT96, SMSM99], but the details were only worked out recently (see, e.g., [YGL22]). Variants of REINFORCE based on stochastic variance reduction, a technique originally developed for finite-sum optimization [JZ13], can obtain a better sample complexity. The first algorithm to break the ϵ^{-4} barrier was SVRPG [PBC+18], for which [XGG20a] proved $O\left(\epsilon^{-10/3}\right)$ sample complexity. A further improvement was achieved by SRVR-PG [XGG20b], with a sample complexity of $O\left(\epsilon^{-3}\right)$. This is commonly believed to be optimal (e.g., [YLLZ20]) due to a lower bound for the related setting of non-convex stochastic optimization [ACD+23], although the latter is not directly comparable with policy optimization. Several other algorithms followed: STORM-PG [YLLZ20], PAGE-PG [GZM+22], and others [PNP+20, HGPH20], all with $O\left(\epsilon^{-3}\right)$ sample complexity. Modest empirical improvements over REINFORCE were also observed in these works.

However, all these algorithms employ importance weights to keep the variance-reduced gradient unbiased, an issue specific to RL [PBC+18], which is absent in stochastic optimization. Their sample complexity upper bounds rely on an unrealistic assumption on the variance of the importance weights, which can be infinite in practically relevant cases [ZNS+21]. Notable exceptions are algorithms based on second-order information [SRH+19, FBKH23], and TSIVR-PG by [ZNS+21]. The latter could remove the unrealistic assumption (while still using importance weights) by truncating the gradients. However, their $\widetilde{O}\left(\epsilon^{-3}\right)$ upper bound only holds for softmax policy parametrizations, which are impractical for continuous action spaces. More recently [FBKH23] proposed an accelerated policy gradient algorithm that uses neither importance weighting nor second-order information. However, its sample complexity to find an ϵ-FOSP is $O\left(\epsilon^{-3.5}\right)$. It is then only natural to ask the following question: ***Can policy gradient algorithms achieve*** $O\left(\epsilon^{-3}\right)$ ***sample com-***

plexity to find an ϵ-FOSP, for general policy classes, using first-order information only and without any assumption on the variance of importance weights?

In this paper, we answer positively by designing a variance-reduced policy gradient algorithm based on defensive importance sampling [OZ00], a variant of importance weighting with naturally bounded variance. Although we build on existing variancereduced policy gradient algorithms ([GZM$^+$22], in particular), we are the first to introduce to this line of work the defensive importance sampling technique, which overcomes in a very natural way a major limitation of existing algorithms. Fully exploiting this method poses two main challenges. The first is to design and analyze a defensive sampling scheme within the context of policy gradient learning. The second, more technical, is a characterization of the continuity of the variance of the defensive importance weights, as boundedness alone is not enough to prove the sample complexity upper bound (Lemma 4). Moreover, having removed the unrealistic assumption on the variance of importance weights for general policy classes, we can prove a matching $\Omega\left(\epsilon^{-3}\right)$ sample-complexity lower bound by adapting the one for non-convex stochastic optimization [ACD$^+$23]. Besides answering these theoretical questions, an algorithm based on defensive importance sampling has the potential to be more stable in practice, leading to faster learning. We investigate this through numerical simulations on continuous control benchmarks.

2 Preliminaries

Notation. For a measurable space $\mathcal{X}$ denote by $\Delta_{\mathcal{X}}$ the set of probability measures over $\mathcal{X}$, and by Δ_m the $(m+1)$-dimensional probability simplex. We denote by $\chi^2(P \mid Q)$ the chisquare divergence between distributions P and Q.

We consider independent episodes of interaction, of indefinite length, between an agent and an environment. In each timestep $h = 0, 1, \ldots$ of an episode, the agent observes a state S_h, performs an action A_h, and receives a reward R_{h+1}. We model this as a discounted Markov decision problem (Puterman, 2014), that is a 6-tuple $(\mathcal{S}, \mathcal{A}, P, r, \mu_0, \gamma)$. Here $\mathcal{S}$ is a state space, which is either very large or continuous; $\mathcal{A}$ is an action space, which may be discrete or continuous; $P : \mathcal{S} \times \mathcal{A} \to \Delta_{\mathcal{S}}$ is a transition probability kernel, such that $p(s' \mid s, a) = \mathbb{P}(S_{h+1} = s' \mid S_h = s, A_h = a)$ (or the corresponding probability density); $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, such that $R_{h+1} = R(S_h, A_h)$; $\gamma \in [0, 1)$ is a discount factor; and $\mu_0 \in \Delta_{\mathcal{S}}$ is the initial state distribution, such that $\mu_0(s) = \mathbb{P}(S_0 = s)$ (or the corresponding probability density). We assume the reward is bounded, namely $|R(s, a)| \leq R_{\max} < +\infty$ for all $s \in \mathcal{S}, a \in \mathcal{A}$.

We consider an episodic learning process where the agent has to commit, at the beginning of each episode, to a Markovian non-stationary (stochastic) policy $\pi : \mathcal{S} \to \Delta_{\mathcal{A}}$, such that $\pi(a \mid s) = \mathbb{P}(A_t = a \mid S_t = s)$ (or the corresponding probability density). The goal of the agent is to find a policy that maximizes the expected discounted return (sum of discounted rewards):

$$\max_{\pi} J(\pi) := \mathbb{E}\left[\sum_{h \geq 0} \gamma^h R_{h+1} \,\middle|\, \pi\right] \tag{2}$$

where the expectation is with respect to $S_0 \sim \mu_0, A_h \sim \pi(\cdot \mid S_h)$, and $S_{h+1} \sim P(\cdot \mid S_h, A_h)$.

In particular, we restrict our attention to parametric policies. Let $\Theta \subseteq \mathbb{R}^d$ be a space of d-dimensional parameters. We assume $\Theta = \mathbb{R}^d$ for simplicity, but all the sample-complexity results can be generalized to constrained parameter spaces as long as Θ is convex. The set of policies the agent can choose from (called the policy space) is defined as $\Pi_\Theta = \{\pi_\theta \mid \theta \in \Theta\}$, where each policy

is parametrized by an element of Θ. The policy optimization problem can then be characterized as an optimization problem over Θ, with slight abuse of notation:

$$\max_{\theta \in \Theta} J(\theta) := J\left(\pi_\theta\right) \tag{3}$$

In the following, J will always denote the parametric objective function, that is $J : \Theta \to \mathbb{R}$. We will call this the performance function. From the bounded reward assumption, we immediately note that the performance function is also bounded, namely $|J(\theta)| \leq \frac{R_{\max}}{1-\gamma}$ for all $\theta \in \Theta$.

In general, J is a non-convex function. However, it has a special structure. If we denote by $\tau = (S_0, A_0, R_1, S_1, A_1, \ldots)$ a trajectory, each policy $\pi \in \Pi_\Theta$ induces a parametric distribution over trajectories:

$$p(\tau \mid \boldsymbol{\theta}) = \mu_0\left(S_0\right) \pi_\theta\left(A_0 \mid S_0\right) P\left(S_1 \mid S_0, A_0\right) \ldots \tag{4}$$

Our objective function can then be written as:

$$J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot \boldsymbol{\theta})}[R(\tau)] \tag{5}$$

where, with abuse of notation, $R(\tau) = \sum_{h \geq 0} \gamma^h R_{h+1}$ denotes the discounted return. It is then an expectation of a fixed function R, which only depends on θ through the (non-convex) probability kernel $p(\cdot \mid \theta)$. Both the return function R and the mapping $\boldsymbol{\theta} \mapsto p(\cdot \mid \boldsymbol{\theta})$ are unknown to the agent.

We are interested in algorithms that run a sequence of policies $\pi_{\theta_1}, \pi_{\theta_2}, \ldots$, each for one or more independent episodes, collecting datasets of trajectories $\mathcal{D}_1, \mathcal{D}_2, \ldots$, where $\mathcal{D}_k = \left\{\tau_1^k, \ldots, \tau_{n_k}^k\right\}$ is a dataset of n_k independent trajectories such that $\tau_i^k \sim p\left(\cdot \mid \boldsymbol{\theta}_k\right)$, with the aim of maximizing the unknown performance function J from the collected data. Policy gradient algorithms [Wil92] do this via stochastic gradient ascent:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta \hat{\nabla} J\left(\boldsymbol{\theta}_k\right) \tag{6}$$

where $\alpha > 0$ is a step size and $\hat{\nabla} J(\theta)$ is an estimate of the gradient $\nabla_\theta J(\theta)$ of the performance function. In the following, we will omit the θ subscript since gradients are always with respect to the policy parameters. The simplest policy gradient algorithm is REINFORCE [Wil92], which we present here in its refined version for Markov decision processes, known as GPOMDP [BB01]. Here $\widehat{\nabla} J\left(\boldsymbol{\theta}_k\right)$ is computed from the batch of trajectories $\mathcal{D}_k$ collected with π_{θ_k} as follows:

$$\hat{\nabla} J\left(\boldsymbol{\theta}_k\right) = \frac{1}{n_k} \sum_{i=1}^{n_k} g\left(\boldsymbol{\theta}_k, \tau_i\right) \tag{7}$$

$$\text{where } g(\boldsymbol{\theta}, \tau) = \sum_{h \geq 0} \left(\sum_{\ell=0}^{k} \nabla \log \pi_\theta\left(A_\ell \mid S_\ell\right) \right) \gamma^h R_{h+1} \tag{8}$$

where the mapping $\boldsymbol{\theta} \mapsto \pi_\theta$ is assumed to be differentiable with respect to θ. The GPOMDP gradient estimator is unbiased, that is, $\mathbb{E}_{\tau \sim p(\cdot \mid \boldsymbol{\theta})}[g(\boldsymbol{\theta}, \tau)] = \nabla J(\boldsymbol{\theta})$, and the same holds for $\widehat{\nabla} J(\boldsymbol{\theta})$ since the trajectories are independent. Note how, for this to be true, the sample trajectory must be collected with the same policy parameters of which we are estimating the gradient. This is on-policy gradient estimation. Moreover, under standard assumptions on the policy class, the variance of the GPOMDP estimator is bounded [ZHNS12, PPR22], albeit infamously large. Common refinements of GPOMDP are variance-reducing baselines [PS08] and actor-critic algorithms [BSA83, KT99, SMSM99], but we will stick to the simple expression from (8) in our theoretical treatment.

Off-policy gradient estimation is also possible [DWS12]. If $\tau \sim p(\cdot \mid \theta')$ is collected with a different policy $\pi_{\theta'}$, an unbiased estimator of $\nabla J(\theta)$ is:

$$\widetilde{g}(\boldsymbol{\theta}, \tau) = \frac{p(\tau \mid \theta)}{p\left(\tau \mid \boldsymbol{\theta}'\right)} g(\boldsymbol{\theta}, \tau) \tag{9}$$

Note that $g(\boldsymbol{\theta}, \tau)$ is not an unbiased estimator for $\nabla J(\boldsymbol{\theta})$ in this case since τ is collected with a different policy $\pi_{\boldsymbol{\theta}'}$. However, it is easy to see that the ratio $p(\tau \mid \boldsymbol{\theta}) / p\left(\tau \mid \boldsymbol{\theta}'\right)$ ensures $\mathbb{E}_{\tau \sim p(\theta')}[\widetilde{g}(\boldsymbol{\theta}, \tau)] = \nabla J(\boldsymbol{\theta})$. This is called importance weighting. The importance weight is usually easy to compute: since the reward R_{h+1} and the next-state S_{h+1} are independent of $\boldsymbol{\theta}$ conditionally on the action A_h, the importance weight for $\tau \sim p(\cdot \mid \theta')$ is simply:

$$\frac{p(\tau \mid \boldsymbol{\theta})}{p\left(\tau \mid \boldsymbol{\theta}'\right)} = \prod_{h \geq 0} \frac{\pi_\theta\left(A_h \mid S_h\right)}{\pi_{\theta'}\left(A_h \mid S_h\right)} =: \omega(\boldsymbol{\theta}, \tau) \tag{10}$$

Of course, when $\boldsymbol{\theta} = \boldsymbol{\theta}', \omega(\boldsymbol{\theta}, \tau) = 1$ for any $\tau \sim p(\cdot \mid \theta)$ and we recover the on-policy case. Unfortunately, the variance of the off-policy estimator is not bounded in general since the variance of importance weights can be infinite even for simple policy classes [MPMR20, ZNS+21]. It is still possible to characterize the variance of importance weights in terms of the chi-square divergence [CMM10]:

$$\operatorname{Var}_{\tau \sim p(\cdot \mid \boldsymbol{\theta}')}[\omega(\boldsymbol{\theta}, \tau)] = \int_{\mathcal{T}} \frac{p(\tau \mid \boldsymbol{\theta})^2}{p\left(\tau \mid \boldsymbol{\theta}'\right)} \mathrm{d}\tau - 1 = \chi^2\left(p(\cdot \mid \boldsymbol{\theta}) \| p\left(\cdot \mid \boldsymbol{\theta}'\right)\right) \tag{11}$$

where $\mathcal{T}$ denotes the set of feasible trajectories. However, the chi-square divergence is unbounded in general, and it is even larger than the Kullback-Liebler divergence that appears more frequently in the policy optimization literature [SLA+15]. For this reason, most works on the sample complexity of policy gradient [PBC+18, XGG20b, YLLZ20, GZM+22]) resort to the following strong assumption, a uniform upper bound on the variance of the importance weights:

Assumption 1. *There exists a finite constant $W > 0$ such that*

$$\sup_{\theta_1, \theta_2 \in \Theta} \operatorname{Var}_{\tau \sim p(\cdot \mid \theta_2)}\left[\omega\left(\boldsymbol{\theta}_1, \tau\right)\right] \leq W$$

As thoroughly discussed by [ZNS+21], this assumption is often violated in practice, as shown below.

Example 1. Consider the class of 1-dimensional Gaussian policies with constant mean and where $\theta \in \mathbb{R}$ parametrizes the variance as $\sigma_\theta = e^\theta$. Consider $\theta_1 \neq \theta_2$, recalling the relation between the chi-square divergence and the variance of the importance weights and exploiting the closed-form expression for Gaussian distributions [GAL13], we have:

$$\operatorname{Var}_{\tau \sim p(\cdot \mid \theta_2)}\left[\omega\left(\theta_1, \tau\right)\right] = \frac{\sigma_{\theta_2}^2}{\sigma_{\theta_1} \sqrt{2\sigma_{\theta_2}^2 - \sigma_{\theta_1}^2}} - 1 \tag{12}$$

when $2\sigma_{\theta_2}^2 - \sigma_{\theta_1}^2 > 0$, otherwise, the variance is infinite. Thus, it suffices to select the target parameter θ_1 such that:

$$2e^{2\theta_2} - e^{2\theta_1} \leq 0 \Longrightarrow \theta_1 \geq \theta_2 + \log\sqrt{2} \tag{13}$$

to make the variance infinite. We remark that this result holds for every $\theta_2 \in \mathbb{R}$, i.e., we do not need to send the Gaussian standard deviation to zero to make the variance of the importance weights diverge.

3 Algorithm

In this section, we present our modified policy gradient algorithm. Like SVRPG [PBC+18] and its successors [XGG20b, YLLZ20, GZM+22], our algorithm is based on the technique of stochastic variance reduction from finite-sum optimization [RSB12, JZ13]. The idea of stochastic variance-reduction is to alternate between "full gradient" and "stochastic gradient" updates. In the context of machine learning [LJ17], a full gradient is a gradient estimate computed from a large batch of data, and a stochastic gradient is computed from a smaller batch or a single sample. The full gradient estimate is more precise and allows more stable updates, but has a larger impact on the sample complexity. The key idea is to combine infrequent full gradient estimates with frequent stochastic gradients, using the former to stabilize the latter.

As a concrete example, consider the modified gradient estimator used by the PAGE algorithm [LBZR21]:

$$v_t = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} g\left(x_t, z_i\right) & \text{with probability } p \\ \frac{1}{B}\sum_{i=1}^{B} g\left(x_t, z_i\right) + v_{t-1} - \frac{1}{B}\sum_{i=1}^{B} g\left(x_{t-1}, z_i\right) & \text{with probability } 1-p \end{cases} \tag{14}$$

where the z_i are i.i.d. samples from a fixed data distribution ρ, the objective function is $F(x) = \mathbb{E}_{z\sim\rho}[f(x,z)]$, g is an unbiased gradient estimator, $\mathbb{E}_{z\sim\rho}[g(x,z)] = \nabla_x F(x)$, $B \ll N$, $0 < p \ll 1$, and $v_0 = 0$. Since p is small, the full gradient is computed infrequently. In most iterations, instead, it is combined, in a recursive fashion, with a stochastic gradient computed from a smaller number of samples. The negative term in Eq. (14) is a correction term, which is fundamental to control the bias-variance tradeoff.

Applying this and similar techniques to policy optimization comes with an additional challenge that is peculiar to RL, as observed by [PBC+18]: distribution shift. Here, the trajectories are not sampled from a fixed distribution, but from $p\left(\cdot \mid \boldsymbol{\theta}_t\right)$, the induced trajectory distribution, which changes with time and is itself a function of the optimization variable $\boldsymbol{\theta}$. So far, this issue has been bypassed by applying importance weighting to the correction term, at the cost of unrealistically assuming a uniform upper bound (Assumption 1) on the variance of the importance weights [ZNS+21]. For example, the modified policy gradient estimator for PAGE-PG [GZM+22] is:

$$v_t = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} g\left(\boldsymbol{\theta}_t, \tau_i\right) & \text{prob. } p \\ \frac{1}{B}\sum_{i=1}^{B} g\left(\boldsymbol{\theta}_t, \tau_i\right) + v_{t-1} - \frac{1}{B}\sum_{i=1}^{B} \omega\left(\boldsymbol{\theta}_{t-1}, \tau_i\right) g\left(\boldsymbol{\theta}_{t-1}, \tau_i\right) & \text{prob. } 1-p \end{cases} \tag{15}$$

where g is the REINFORCE or the GPOMDP estimator and the trajectories τ_i are always sampled from $p\left(\cdot \mid \theta_t\right)$ by running the current policy π_{θ_i}. Note how the importance weight is needed because the correction term is an off-policy estimate of the policy gradient of $\boldsymbol{\theta}_{t-1}$ using trajectories from a different policy $\boldsymbol{\theta}_t$. We refer to the former as the target policy and to the latter as the behavior policy.

Our key observation to avoid Assumption 1 is the following: although this use of importance weighting is common in offline RL [LKTF20], where data from the target policy is not accessible, we find ourselves in a situation where obtaining data from the target policy is actually very easy. Indeed, the target policy with parameters θ_{t-1} is just an older policy we ran in the previous iteration. This allows us to use defensive importance sampling [OZ00], a special kind of mixture importance sampling that guarantees that the variance of the importance weights is always bounded.

In mixture importance sampling, to estimate $\bar{f} = \mathbb{E}_{z\sim p}[f(z)]$ for $p \in \Delta_{\mathcal{Z}}$ and $f : \mathcal{Z} \to \mathbb{R}$, we draw n samples $Z_1, \ldots, Z_n$ from the mixture distribution $q_\alpha = \sum_{j=1}^{m} \alpha_j q_j$, where $\{q_1, \ldots, q_m\}$ is

a set of m distributions in $\Delta_{\mathcal{Z}}$, and $\boldsymbol{\alpha} \in \Delta_m$ is a vector of convex-combination coefficients. The mean $\bar{f}$ is estimated as:

$$\hat{f} = \frac{1}{n} \sum_{i=1}^{n} \frac{p\left(Z_i\right)}{\sum_{j=1}^{m} \alpha_j q_j\left(Z_i\right)} f\left(Z_i\right) \tag{16}$$

It is easy to see that this estimate is unbiased. Defensive importance sampling is a special case where the mixture distribution is simply:

$$q_\alpha(x) = \alpha p(x) + (1-\alpha) q(x) \tag{17}$$

for some scalar $\alpha \in [0,1]$. Note that it recovers vanilla importance sampling for $\alpha = 0$, and Monte Carlo estimation for $\alpha = 1$. Most importantly, it is easy to see that, for $\alpha < 1$:

$$\mathrm{Var}_{z \sim q_\alpha}\left[\frac{p(z)}{q_\alpha(z)}\right] \leq \frac{1-\alpha}{\alpha} \tag{18}$$

We will refer to α as the defensive parameter: the larger the α, the smaller the variance of the importance weights $p(z)/q_\alpha(z)$, and call the estimator " α-defensive".

In our algorithm, we apply defensive importance sampling to the PAGE technique, achieving the same $O\left(\epsilon^{-3}\right)$ rate of PAGE-PG without Assumption 1 , thanks to the bounded-variance property of α-defensive estimators (Eq. 18). Our modified policy gradient estimator is:

$$v_t = \begin{cases} \frac{1}{N} \sum_{i=1}^{N} g\left(\boldsymbol{\theta}_t, \bar{\tau}_i\right) & \text{with probability } p \\ \frac{1}{B} \sum_{i=1}^{B} \omega\left(\boldsymbol{\theta}_t, \tau_i\right) g\left(\boldsymbol{\theta}_t, \tau_i\right) + v_{t-1} - \frac{1}{B} \sum_{i=1}^{B} \omega\left(\boldsymbol{\theta}_{t-1}, \tau_i\right) g\left(\boldsymbol{\theta}_{t-1}, \tau_i\right) & \text{otherwise} \end{cases} \tag{19}$$

where the N trajectories $\bar{\tau}_i$ for the full gradient are sampled from $p\left(\cdot \mid \theta_t\right)$, but the B trajectories τ_i are sampled from the mixture distribution $q_{\alpha,t} = \alpha p\left(\cdot \mid \theta_t\right) + (1-\alpha) p\left(\cdot \mid \theta_{t-1}\right)$. As a consequence of $\tau_i \sim q_{\alpha,t}$, the importance weights become:

$$\omega\left(\boldsymbol{\theta}_t, \tau_i\right) = \frac{p\left(\tau_i \mid \boldsymbol{\theta}_t\right)}{\alpha p\left(\tau_i \mid \boldsymbol{\theta}_t\right) + (1-\alpha) p\left(\tau_i \mid \boldsymbol{\theta}_{t-1}\right)} = \frac{1}{\alpha + (1-\alpha) \frac{p(\tau_i \mid \theta_{t-1})}{p(\tau_i \mid \theta_t)}} \tag{20}$$

$$\omega\left(\boldsymbol{\theta}_{t-1}, \tau_i\right) = \frac{p\left(\tau_i \mid \boldsymbol{\theta}_{t-1}\right)}{\alpha p\left(\tau_i \mid \boldsymbol{\theta}_t\right) + (1-\alpha) p\left(\tau_i \mid \theta_{t-1}\right)} = \frac{1}{\alpha \frac{p(\tau_i \mid \theta_t)}{p(\tau_i \mid \theta_{t-1})} + 1 - \alpha} \tag{21}$$

that is, defensive importance weights. In practice, this can be done by flipping a coin at the beginning of each episode and running the current policy π_{θ_θ} with probability α or the previous policy $\pi_{\theta_{t-1}}$ otherwise. Notice that, differently from PAGE-PG (Eq. 14), both $g\left(\boldsymbol{\theta}_t, \tau_i\right)$ and $g\left(\boldsymbol{\theta}_{t-1}, \tau_i\right)$ are weighted, since both are partially off-policy: the first is an α-defensive estimate of $\nabla J\left(\boldsymbol{\theta}_t\right)$ with behavior distribution $p\left(\cdot \mid \theta_{t-1}\right)$. The second is an $(1-\alpha)$-defensive estimate of $\nabla J\left(\boldsymbol{\theta}_{t-1}\right)$ with behavior distribution $p\left(\cdot \mid \boldsymbol{\theta}_t\right)$. We will show in §4 how the defensive importance-sampling components eliminate the need for any requirements on the variance of the importance weights. The full pseudocode can be seen in Algorithm 1.

The per-iteration time complexity is polynomial in N and d. The overall space complexity is polynomial in d and independent of T since we only need to store v_t for at most one extra iteration. Also note that it is not necessary to store $\theta_1, \ldots, \theta_T$ to compute θ_{OUT} : we can equivalently stop at a random iteration t and return $\boldsymbol{\theta}_t$.

Algorithm 1 DEF-PG

Input: Initial parameter $\boldsymbol{\theta}_0$, large batch-size N, small batch-size B, step-size $\eta > 0$, probability $p \in (0,1)$, defensive parameter $\alpha \in (0,1)$
Collect N trajectories $\{\bar{\tau}_1, \ldots, \bar{\tau}_N\}$ with policy $\boldsymbol{\theta}_0$
$v_0 \leftarrow \frac{1}{N}\sum_{i=1}^{N} g\left(\boldsymbol{\theta}_0, \bar{\tau}_i\right)$
for $t = 0$ **to** $T-1$ **do**
    $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \eta v_t$
    if with probability p **then** ▷ Full gradient
        Collect N trajectories $\{\bar{\tau}_1, \ldots, \bar{\tau}_N\}$ with $\pi_{\boldsymbol{\theta}_t}$
        $v_{t+1} \leftarrow \frac{1}{N}\sum_{i=1}^{N} g\left(\boldsymbol{\theta}_t, \bar{\tau}_i\right)$
    else ▷ Stochastic gradient
        for $i = 1$ **to** B **do**
            if with probability α **then**
                Collect trajectory τ_i with $\pi_{\boldsymbol{\theta}_t}$
            else
                Collect trajectory τ_i with $\pi_{\theta_{t-1}}$
            end if
        end for
        $v_{t+1} \leftarrow \frac{1}{B}\sum_{i=1}^{B} \omega\left(\boldsymbol{\theta}_t, \tau_i\right) g\left(\boldsymbol{\theta}_t, \tau_i\right) + v_{t-1} - \frac{1}{B}\sum_{i=1}^{B} \omega\left(\boldsymbol{\theta}_{t-1}, \tau_i\right) g\left(\boldsymbol{\theta}_{t-1}, \tau_i\right)$
    end if
end for
Output: $\boldsymbol{\theta}_{\mathrm{OUT}}$: chosen uniformly at random from $\{\boldsymbol{\theta}_t\}_{t=1}^{T}$

4 Sample complexity upper bounds

In this section, we recall the sample complexity upper bound of GPOMDP [BB01] and establish improved guarantees for Algorithm 1, matching the best-known result (e.g., [GZM$^+$22]) without the restrictive Assumption 1.

The key assumption for GPOMDP is the following:

Assumption 2 (E-LS, [PPR22, YGL22])**.** *There exist two finite constants $c_1, c_2 > 0$, such that for all $\boldsymbol{\theta} \in \Theta$:*

$$\sup_{s \in \mathcal{S}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left[\|\nabla \log \pi_\theta(a \mid s)\|^2 \right] \leq c_1^2 \tag{22}$$

$$\sup_{s \in \mathcal{S}} \mathbb{E}_{a \sim \pi_\theta(\cdot|s)} \left\| \nabla^2 \log \pi_\theta(a \mid s) \right\| \leq c_2 \tag{23}$$

Under this assumption, GPOMDP (and REINFORCE) has $O\left(\epsilon^{-4}\right)$ sample complexity [YGL22]. There are two fundamental steps to establish this result, and both of them will be crucial for our analysis of Algorithm 1 as well. We report them here for completeness. The first is an upper bound on the variance of the policy gradient estimator:

Proposition 1 (Lemma 29 of [PPR22]; Lemma 4.2 of [YGL22])**.** *Let g denote the GPOMDP estimator (Eq. 8). Under Assumption 2, for all $\boldsymbol{\theta} \in \Theta$:*

$$\mathrm{Var}_{\tau \sim p(\cdot|\theta)}[g(\boldsymbol{\theta}, \tau)] \leq \frac{c_1^2 R_{\max}^2}{(1-\gamma)^3} =: v^2$$

The second is a smoothness result on the performance function:

Proposition 2 (Lemma 4.4 of [YGL22]). *Under Assumption 2, for all $\boldsymbol{\theta} \in \Theta$:*

$$\left\|\nabla^2 J(\theta)\right\| \leq \frac{\left(c_1^2 + c_2\right) R_{\max}}{(1-\gamma)^2} =: L \tag{24}$$

For completeness, we report the best-known result on the sample complexity of GPOMDP:

Proposition 3 (Sample complexity of GPOMDP; Corollary 4.7 of [YGL22]). *Under Assumption 2, the following total number of trajectories is sufficient for GPOMDP to output an ϵ-FOSP in expectation:*

$$N_{\mathrm{TOT}} \leq \frac{8\left(J\left(\boldsymbol{\theta}^*\right) - J\left(\boldsymbol{\theta}_0\right)\right) L v^2}{\epsilon^4} = O\left((1-\gamma)^{-5}\epsilon^{-4}\right)$$

that is, $\mathbb{E}\left[\nabla J\left(\boldsymbol{\theta}_{OUT}\right)\right] \leq \epsilon$, where v^2 is from Proposition 1 and L is from Proposition 2.

In works on stochastic variance-reduced policy gradient, the following stronger assumption is usually made:

Assumption 3 (LS, [PBC+18, YGL22]). *There exist two finite constants $C_1, C_2 > 0$ such that for all $\theta \in \Theta$:*

$$\sup_{s \in \mathcal{S}, a \in \mathcal{A}} \|\nabla \log \pi_\theta(a \mid s)\| \leq C_1, \tag{25}$$

$$\sup_{s \in \mathcal{S}, a \in \mathcal{A}} \left\|\nabla^2 \log \pi_\theta(a \mid s)\right\| \leq C_2 \tag{26}$$

Remark. Trivially, Assumption 3 is stronger than Assumption 2, thus Propositions 1, 2, and 3 continue to hold under Assumption 3 [YGL22].

While Assumption 2 is naturally satisfied by Softmax and Gaussian policies[PPR22], Assumption 3 fails to hold for Gaussians unless some form of clipping is applied to the actions [YGL22]. Several works have demonstrated $O\left(\epsilon^{-3}\right)$ sample complexity under Assumption 3 (e.g., [XGG20b, GZM+22]) and Assumption 1, the unrealistic requirement on the variance of importance weights. Whether the same can be achieved with the weaker Assumption 2 was not clear so far. In §5, we answer this open question in the negative, proving a $\Omega\left(\epsilon^{-4}\right)$ lower bound under Assumption 2.

We now present the main result of this section, an $O\left(\epsilon^{-3}\right)$ sample-complexity upper bound under Assumption 3, but without Assumption 1. We start with a technical lemma, the key to exploiting the favorable properties of defensive importance sampling. The proofs of this and the other novel results from this section are in Appendix C.1

Lemma 1. *Under Assumption 3, the variance of defensive importance weights is bounded as follows. For any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$ and $\alpha \in (0,1)$, let q_α be the mixture distribution $q_\alpha(\tau) = \alpha p\left(\tau \mid \boldsymbol{\theta}_1\right) + (1-\alpha) p\left(\tau \mid \boldsymbol{\theta}_2\right)$. Then:*

$$\mathrm{Var}_{\tau \sim q_\alpha}\left[\omega\left(\boldsymbol{\theta}_1, \tau\right)\right] \leq C_{\omega,\alpha} \left\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\right\|^2 \tag{27}$$

where $C_{\omega,\alpha} = \left(C_1^2 + C_2\right)\left(\frac{4}{\alpha} + \frac{6(1-\alpha)}{\alpha^2}\right) + C_1^2\left(2 + 2\alpha^2 + 4\alpha(1-\alpha)\right)$.

From a technical standpoint, our Lemma 4 replaces Lemma 6.1 by [XGG20a], the key tool that allowed to establish the sample complexity of SVRPG and its successors under Assumption 1. We were able to remove the latter only thanks to defensive samples. Note how, for $\alpha = 0$, the LHS of Equation (27) reduces to the variance of vanilla importance weights, but the upper bound on the RHS becomes vacuous since $C_{\omega,\alpha} \to \infty$ as $\alpha \to 0$. Armed with this powerful technical tool, we can prove the following:

Theorem 1 (Sample complexity of DEF-PG)**.** *For any $\alpha \in (0,1)$ and $\theta_0 \in \Theta$ such that $\Delta = J(\boldsymbol{\theta}^*) - J(\boldsymbol{\theta}_0) < \infty$, under Assumption 3, for an appropriate choice of hyperparameters η, p, and $N \geq \frac{3\sigma^2}{\epsilon^2}$, the following total number of trajectories is sufficient for Algorithm 1 to output an ϵ-FOSP in expectation:*

$$N_{TOT} \leq \left(\frac{6\Delta\sqrt{2C_\alpha}}{\epsilon^2} \frac{\sqrt{N}}{\sqrt{B}} + \frac{3\sigma^2}{\epsilon^2} \right) \left(1 + B - \frac{B}{N} \right)$$

Setting $B = O(1)$ and $N = O\left(\epsilon^{-2}\right)$, the sample complexity is $O\left(\epsilon^{-3}\right)$.

The above result also provides a natural choice for the defensive parameter:

Corollary 1. *The choice of the defensive parameter α that minimizes the upper bound from Theorem 4 is $\alpha = \frac{1}{2}$, yielding a sample complexity for Algorithm 1 of*

$$N_{TOT} \leq \left(\frac{6\Delta\sqrt{47C_1^2 + 40C_2}\big)}{\epsilon^2} \frac{\sqrt{N}}{\sqrt{B}} + \frac{3\sigma^2}{\epsilon^2} \right) \left(1 + B - \frac{B}{N} \right) = O\left(\epsilon^{-3}\right)$$

Remark. The choice $\alpha = 1/2$ agrees with intuition, by the symmetry of the gradient estimator v_{t+1} in Algorithm 1: any smaller choice of the defensive parameter would inject extra variance in the first term; any larger choice, in the third.

This sample complexity matches the best rate known so far (e.g., [GZM+22]), but without the restrictive Assumption 1 or access to second-order information, and for a wider class of policies compared to [ZNS+21]. This rate is also considered optimal after a related result on non-convex stochastic optimization [ACD+23]. In the following section, we prove that it is indeed optimal. Furthermore, we show that Assumption 3 is necessary and just Assumption 2 does not suffice to obtain an $O\left(\epsilon^{-3}\right)$ sample complexity. We summarize our and related results in Table 1, including the lower bounds from the next section.

5 Sample complexity lower bounds

In this section, we prove lower bounds on the sample complexity of first-order policy gradient algorithms to find an ϵ-FOSP. Depending on the assumptions on the policy class, they are matched by GPOMDP or by our DEF-PG (Algorithm 1).

Our lower bounds are based on the work by [ACD+23] on non-convex stochastic optimization. However, they are not simple reductions due to the special structure of the performance function J, which does not allow us to construct arbitrary objective functions, and to the different protocol, summarized in Algorithm 2.

It is easy to see that GPOMDP complies with this protocol. We further restrict the class of algorithms of interest with the following:

Table 1. First-order sample complexity upper and lower bounds for different policy gradient algorithms under different assumptions on the policy class. Rows are ordered from weaker to stronger sets of assumptions (recall that Asm. 3 implies Asm. 2)

Assumptions	Upper bound	Algorithm	Lower bound
Asm. 2	$O\left(\epsilon^{-4}\right)$ [YGL22]	GPOMDP	$\Omega\left(\epsilon^{-4}\right)$ (Thm. 2)
Asm. 3	$O\left(\epsilon^{-3}\right)$ (Thm. 4)	DEF-PG (ours)	$\Omega\left(\epsilon^{-3}\right)$ (Thm. 3)
Asm. 3 and 1	$O\left(\epsilon^{-3}\right)$ (e.g., [GZM+22])	PAGE-PG	$\Omega\left(\epsilon^{-3}\right)$ (Thm. 3)

Algorithm 2 First-order Policy Gradient

for $t = 1, \dots, N$ **do**
  Select policy parameter $\boldsymbol{\theta}_t$
  Execute policy $\pi_{\boldsymbol{\theta}_t}$ and collect trajectory $\tau_t \sim p(\cdot|\boldsymbol{\theta}_t)$
  Obtain (on-policy) gradient estimate $g(\tau_t|\boldsymbol{\theta}_t)$
  if Assumption 5 holds **then**
    Obtain probability ratios $\frac{p(\tau_t|\boldsymbol{\theta}_k)}{p(\tau_t|\boldsymbol{\theta}_t)}$ for all $k \leq t$
  end if
end for

Assumption 4. *We consider algorithms that comply with Algorithm 2, such that* $\mathbb{E}_{\tau\sim p(\cdot|\theta_t)}\left[g\left(\tau \mid \theta_t\right)\right] = \nabla J\left(\boldsymbol{\theta}_t\right)$ *and* $\mathrm{Var}_{\tau\sim p(\cdot|\theta_t)}\left[g\left(\tau \mid \theta_t\right)\right] \leq \sigma^2$.

By Proposition 1, this reduces to Assumption 2 when g is the GPOMDP gradient estimator, but it allows us to be more general in stating our lower bounds.

In our construction, we will make use of a "hard-to-optimize" function $f_d : \mathbb{R}^d \times \{0, 1\}$ by [ACD+23, Section 5.1, Equation 30] that we will report in Appendix C.2. We will use f_d to design two policy classes. The first, satisfying Assumption 2, will lead to a $\Omega\left(\epsilon^{-4}\right)$ lower bound. The second, satisfying the stronger Assumption 3 but not Assumption 2, will lead to a better $\Omega\left(\epsilon^{-3}\right)$ lower bound. This separation mirrors the one established by [ACD+23] for non-convex stochastic optimization with or without mean-squared smoothness. To achieve the ϵ^{-3} result, we will also need the following amendment to the interaction protocol:

Assumption 5. *We consider algorithms that satisfy Assumption 4, but additionally can obtain probability ratios* $p\left(\tau_t \mid \boldsymbol{\theta}_k\right) / p\left(\tau_t \mid \boldsymbol{\theta}_t\right)$ *for all* $t \geq 1$ *and* $k \leq t$ *(cf. Algorithm 2).*

It is easy to see that our DEF-PG satisfies Assumption 5 since defensive importance weights can easily be computed from probability ratios (Eq. 20). The access to proability ratios involving the distributions of previous policies is realistic, as discussed in §2 and §3. This extra requirement mirrors the one by [ACD+23] of being able to make multiple concurrent calls to the first-order stochastic oracle.

We defer all proofs to Appendix C.2 and just state here our two lower bounds. In the following, we abbreviate $\Delta = J\left(\boldsymbol{\theta}^*\right) - J\left(\boldsymbol{\theta}_0\right)$, while σ^2 is the constant from Assumption 4.

Theorem 2. *For any algorithm AlG satisfying Assumption 4 there exist a policy class* Π_Θ *satisfying Assumption 2 and an MDP* $\mathcal{M}$ *such that, to find an* ϵ*-FOSP, AlG needs a total number of trajectories*

$$N_{TOT} \geq L\left(\frac{\Delta\sigma^2}{\epsilon^4} + \frac{\Delta}{\epsilon^2}\right) = \Omega\left(\epsilon^{-4}\right)$$

for some $L > 0$.

Remark. The construction for this and the following theorem are realized by using the GPOMDP gradient estimator (Eq. 8) in the role of g from Assumption 4.

This lower bound is matched by GPOMDP [BB01], since it satisfies Assumption 4 (Proposition 1) and has $O\left(\epsilon^{-4}\right)$ sample complexity under Assumption 2 (Proposition 3). The same can be shown for REINFORCE [Wil92] with a different σ^2 [YGL22]. This also establishes that a stronger assumption on the policy class, like Assumption 3, is necessary to obtain a better sample complexity. Hence, perhaps surprisingly, REINFORCE is already optimal for Gaussian policies without action clipping, which satisfy Assumption 2 but not Assumption 3 [PPR22].

Theorem 3. *For any algorithm AlG satisfying Assumptions 4 and 5 there exists a policy class Π_Θ satisfying Assumption 3 and an MDP $\mathcal{M}$ such that, to find an ϵ-FOSP, ALG needs a total number of trajectories*

$$N_{\mathrm{TOT}} \geq \bar{L}\left(\frac{\Delta\sigma}{\epsilon^3} + \frac{\Delta}{\epsilon^2} + \frac{\sigma^2}{\epsilon^2}\right) = \Omega\left(\epsilon^{-3}\right)$$

for some $\bar{L} > 0$.

This lower bound is matched by our DEF-PG (Algorithm 1). It satisfies Assumption 5 by using the same gradient estimator as GPOMDP and computing importance weights w.r.t. previous policies, and has $O\left(\epsilon^{-3}\right)$ sample complexity under Assumption 3. This also establishes that adopting Assumption 1, as done by several previous works, cannot further improve the sample complexity.

6 Conclusion

To summarize our contributions, we designed a first-order policy gradient algorithm based on defensive importance sampling and proved for it a $O\left(\epsilon^{-3}\right)$ sample complexity upper bound to find an ϵ-stationary point under a common, reasonable assumption on the policy class (Assumption 3), but without the unrealistic, albeit equally common, Assumption 1 on the variance of the importance weights. This removes the main disadvantage of algorithms based on importance sampling, which motivated the development of less practical second-order algorithms (e.g. [SRH$^+$19]), and for a much larger family of policies compared to [ZNS$^+$21]. We empirically demonstrated the practical advantages of our algorithm w.r.t. to similar ones based on vanilla importance weighting, corroborating the idea that the variance of importance weights can be counterproductive if not controlled, and that Assumption 1 was hiding a concrete problem. We also showed that the $O\left(\epsilon^{-3}\right)$ rate is optimal with a matching $\Omega\left(\epsilon^{-3}\right)$ lower bound, proving a well-accepted conjecture that was more subtle than expected. We also proved a worse $\Omega\left(\epsilon^{-4}\right)$ under the weaker Assumption 2 on the policy class, which matches a known upper bound for GPOMDP/REINFORCE. This indicates that a stronger assumption on the score function like Assumption 3 is indeed necessary to break the ϵ^{-4} barrier, and that REINFORCE is already optimal for Gaussian policies without action clipping.

These results leave a few open questions in this setting. Our lower bounds, however, could be further refined. First, we did not establish the optimal dependence on the effective horizon $(1-\gamma)^{-1}$. Moreover, our constructions involved policy spaces that are rather artificial. A more involved analysis may achieve the same result for more realistic classes like deep neural policies. Finally, following [ACD$^+$20], we may extend the lower bounds to second-order policy gradient algorithms.

A more exciting direction for future work is to study the global-convergence properties of DEF-PG under Fisher non-degeneracy, or to apply the idea of defensive importance sampling to the design of novel algorithms with faster global convergence.

References

[ACD$^+$20] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In *Conference on Learning Theory*, pages 242–299. PMLR, 2020.

[ACD$^+$23] Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *Mathematical Programming*, 199:165–214, 2023.

[ACG+24] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

[AKLM21] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

[AZ18] Zeyuan Allen-Zhu. How to make the gradients small stochastically: Even faster convex and nonconvex sgd. *Advances in Neural Information Processing Systems*, 31, 2018.

[BB01] Jonathan Baxter and Peter Bartlett. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.

[BSA83] Andrew G Barto, Richard S Sutton, and Charles W Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE transactions on systems, man, and cybernetics*, (5):834–846, 1983.

[BT96] Dimitri Bertsekas and John N Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

[CMM10] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. *Advances in Neural Information Processing Systems*, 23:442–450, 2010.

[CO19] Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. *Advances in Neural Information Processing Systems*, 32:15236–15245, 2019.

[DCL+17] Simon S Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In *International Conference on Machine Learning*, pages 1049–1058. PMLR, 2017.

[DWS12] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. In *International Conference on Machine Learning*, 2012.

[FBKH23] Ilyas Fatkhullin, Anas Barakat, Anastasia Kireeva, and Niao He. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. In *International Conference on Machine Learning*, volume 202, pages 9827–9869. PMLR, 2023.

[FLLZ18] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31:687–697, 2018.

[GAL13] Manuel Gil, Fady Alajaji, and Tamas Linder. Rényi divergence measures for commonly used univariate continuous distributions. *Information Sciences*, 249:124–131, 2013.

[GAMK13] Mohammad Gheshlaghi Azar, Rémi Munos, and Hilbert J Kappen. Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91:325–349, 2013.

[GZM+22] Matilde Gargiani, Andrea Zanelli, Andrea Martinelli, Tyler Summers, and John Lygeros. PAGE-PG: A simple and loopless variance-reduced policy gradient method with probabilistic gradient estimation. In *International Conference on Machine Learning*, volume 162, pages 7223–7240. PMLR, 2022.

[HGH22] Feihu Huang, Shangqian Gao, and Heng Huang. Bregman gradient policy optimization. In *International Conference on Learning Representations*, 2022.

[HGPH20] Feihu Huang, Shangqian Gao, Jian Pei, and Heng Huang. Momentum-based policy gradient methods. In *International conference on machine learning*, volume 119, pages 4422–4433. PMLR, 2020.

[JZ13] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems*, 26:315–323, 2013.

[Kak01] Sham M Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 14:1531–1538, 2001.

[KT99] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in Neural Information Processing Systems*, 12:1008–1014, 1999.

[LBZR21] Zhize Li, Hongyan Bao, Xiangliang Zhang, and Peter Richtárik. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International conference on machine learning*, volume 139, pages 6286–6295. PMLR, 2021.

[LJ17] Lihua Lei and Michael Jordan. Less than a single pass: Stochastically controlled stochastic gradient. In *International Conference on Artificial Intelligence and Statistics*, volume 54, pages 148–156. PMLR, 2017.

[LKTF20] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[LZBY20] Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33:7624–7636, 2020.

[MPMR20] Alberto Maria Metelli, Matteo Papini, Nico Montali, and Marcello Restelli. Importance sampling techniques for policy optimization. *Journal of Machine Learning Research*, 21(141):1–75, 2020.

[NLST17] Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, volume 70, pages 2613–2621. PMLR, 2017.

[Ope23] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[OZ00] Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[PBC+18] Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, volume 80, pages 4026–4035. PMLR, 2018.

[PNP+20] Nhan Pham, Lam Nguyen, Dzung Phan, Phuong Ha Nguyen, Marten Dijk, and Quoc Tran-Dinh. A hybrid stochastic policy gradient algorithm for reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 374–385. PMLR, 2020.

[PPR22] Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients. *Machine Learning*, 111(11):4081–4137, 2022.

[PS08] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008.

[RHRH22] Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, volume 164, pages 91–100. PMLR, 2022.

[RHS+16] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, volume 48, pages 314–323. PMLR, 2016.

[RSB12] Nicolas Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. *Advances in Neural Information Processing Systems*, 25:2672–2680, 2012.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[SKK+22] Saber Salehkaleybar, Sadegh Khorasani, Negar Kiyavash, Niao He, and Patrick Thiran. Momentum-based policy gradient with second-order information. *arXiv preprint arXiv:2205.08253*, 2022.

[SLA+15] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, volume 37, pages 1889–1897. PMLR, 2015.

[SMSM99] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, pages 1057–1063, 1999.

[SRH+19] Zebang Shen, Alejandro Ribeiro, Hamed Hassani, Hui Qian, and Chao Mi. Hessian aided policy gradient. In *International conference on machine learning*, volume 97, pages 5729–5738. PMLR, 2019.

[TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[WBK+22] Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, 2022.

[Wil92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[XGG20a] Pan Xu, Felicia Gao, and Quanquan Gu. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Uncertainty in Artificial Intelligence*, volume 115, pages 541–551. PMLR, 2020.

[XGG20b] Pan Xu, Felicia Gao, and Quanquan Gu. Sample efficient policy gradient methods with recursive variance reduction. In *International Conference on Learning Representations*, 2020.

[XLP17] Tianbing Xu, Qiang Liu, and Jian Peng. Stochastic variance reduction for policy gradient estimation. *arXiv preprint arXiv:1710.06034*, 2017.

[YGL22] Rui Yuan, Robert M Gower, and Alessandro Lazaric. A general sample complexity analysis of vanilla policy gradient. In *International Conference on Artificial Intelligence and Statistics*, volume 151, pages 3332–3380. PMLR, 2022.

[YLLZ20] Huizhuo Yuan, Xiangru Lian, Ji Liu, and Yuren Zhou. Stochastic recursive momentum for policy gradient methods. *arXiv preprint arXiv:2003.04302*, 2020.

[YZZ+22] Long Yang, Yu Zhang, Gang Zheng, Qian Zheng, Pengfei Li, Jianhang Huang, and Gang Pan. Policy optimization with stochastic mirror descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8823–8831, 2022.

[ZHNS12] Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. *Neural Networks*, 26:118–129, 2012.

[ZNS+21] Junyu Zhang, Chengzhuo Ni, Csaba Szepesvari, Mengdi Wang, et al. On the convergence and sample efficiency of variance-reduced policy gradient method. *Advances in Neural Information Processing Systems*, 34:2228–2240, 2021.

A Related works

In this section, we discuss and compare the most related results from the reinforcement learning literature.

Local convergence of first-order algorithms. The $O\left(\epsilon^{-4}\right)$ convergence of REINFORCE/GPOMDP to a stationary point was known for a long time as it follows from standard stochastic approximation results [BT96], but a detailed and refined analysis can be found in the recent work by [YGL22]. For instance, [SMSM99] studied asymptotic convergence under stronger assumptions on the policy space than Assumption 2. The first applications of stochastic variance reduction to RL were for policy evaluation [DCL+17] and trust-region methods [XLP17]. [PBC+18] were the first to apply stochastic variance reduction to a REINFORCE-like algorithm with the aim of improving the sample complexity for finding an ϵ-FOSP, although they did not establish a better rate than $O\left(\epsilon^{-4}\right)$ for their SVRPG algorithm. SVRPG was based on SVRG [JZ13], and its analysis on previous works on nonconvex finite-sum optimization [RHS+16], while SCSG [LJ17], sharing with SVRPG the infinite sum nature of the objective function, was developed independently for supervised learning. To analyze SVRPG, [PBC+18] introduced Assumptions 1 and 3, which later became standard in a line of work adapting results from stochastic optimization to policy gradient methods. They also made an assumption on the variance of the on-policy gradient estimator, although it is already implied by Assumption 3 (cf. Proposition 1). [XGG20a] later showed the sample complexity of SVRPG to be $O\left(\epsilon^{-10/3}\right)$. The same authors [XGG20b] proposed a new first-order algorithm, SRVR-PG, based on SARAH [NLST17] and SPIDER [FLLZ18], with $O\left(\epsilon^{-3}\right)$ sample complexity, also considering constrained parameter spaces and parameter-based exploration. The $O\left(\epsilon^{-3}\right)$ result was matched several times: by [YLLZ20] with STORMPG, based on STORM [CO19], which has the additional advantage of being single loop (so is our algorithm); by [PNP+20] with ProxHSPGA, also considering constrained and regularized problems; by [HGPH20] with IS-MBPG, based on STORM, considering adaptive learning rates; by [HGH22], that in their VR-BGPO algorithm applied stochastic variance reduction to a mirror-descent scheme; and by [GZM+22] with PAGE-PG, based on PAGE, a more recent algorithm for nonconvex optimization [LBZR21]. The last four algorithms have the advantage of using an $O(1)$ batch size for the stochastic-gradient updates, compared to $O\left(\epsilon^{-1/2}\right)$ for SRVR-PG (so does our algorithm). All of this works establish $O\left(\epsilon^{-3}\right)$ sample complexity under Assumption 1 . To see why this assumption is unrealistic, consider the detailed discussion on the variance of importance weights for policy optimization by [MPMR20]. This issue was addressed by [ZNS+21]. Their TSIVR-PG algorithm, based on SARAH/SPIDER, achieves $O\left(\epsilon^{-3}\right)$ sample complexity without any assumption on the variance of importance weights, by truncating the gradients. They also consider general utility functions that cannot be written as a sum of rewards. However, their analysis is tailored to softmax policies, a class of policies that satisfies Assumption 3 but is unfit for continuous control. Our algorithm achieves $O\left(\epsilon^{-3}\right)$ sample complexity without any assumption on the variance of importance weights, like TSIVR-PG, but for general policy classes satisfying Assumption 3, and without gradient truncation. The version we presented is most similar to PAGE-PG, but the same defensive sampling technique can be equally applied to other algorithms based on importance weighting, preserving their sample complexity, but without Assumption 1 (see Appendix D).

Local convergence of second order algorithms. Another way to get rid of Assumption 1 is to use second-order information, which allows to avoid the use of importance weights entirely.[1] The first to do so were [SRH+19]. Their HAPG algorithm enjoys $O\left(\epsilon^{-3}\right)$ sample complexity for ϵ-FOSP convergence and was developed independently from SRVR-PG. Efficient implementations allow to keep the periteration computational complexity linear in the dimension d and the horizon T. A similar solution was proposed by [SKK+22]. Their SHARP algorithm has the advantage of using $O(1)$ batch sizes for the stochastic gradient updates, compared to $O\left(\epsilon^{-1}\right)$ for HAPG.

Lower bounds for local convergence. The $O\left(\epsilon^{-3}\right)$ rate of SRVR-PG and its successors was believed to be optimal due to a $\Omega\left(\epsilon^{-3}\right)$ lower bound for first-order nonconvex stochastic optimization by [ACD+23] which is matched, for instance, by SPIDER. However, the validity of this lower bound for first-order policy gradient methods was less obvious than sometimes stated for the following reasons. On the one hand, the performance objective is not an arbitrary nonconvex function but has a special structure, being defined as the expectation of a parameterindependent function (the return) under a parametric measure (the trajectory distribution). On the other hand, the sampling protocol is different, as data (trajectories) are not drawn from a fixed distribution. Besides working out the details of lower bounds for policy gradient algorithms and establishing the necessity of Assumption 3, we were able to match the lower bound with upper bounds that hold under the same assumptions. This was only possible by removing Assumption 1 from the analysis of first-order PG algorithms. Interestingly, [ACD+20] proved that, in the stochastic optimization framework, the $\Omega\left(\epsilon^{-3}\right)$ continues to hold even with access to second-order information.

Global convergence. Several recent works study the convergence of policy gradient algorithms to a globally optimal policy, that is, $J\left(\boldsymbol{\theta}^*\right) - J(\boldsymbol{\theta}) \leq \widetilde{\epsilon}$. Additional assumptions are needed since the performance function is, in general, nonconvex. A common such assumption is Fisher nondegeneracy, introduced by [LZBY20]. Under this assumption, the latter established $O\left(\widetilde{\epsilon}^{-4}\right)$ sample complexity for GPOMDP, and $O\left(\widetilde{\epsilon}^{-3}\right)$ for Natural Policy Gradient (NPG, [Kak01]), SRVR-PG, and their own algorithm SRVR-NPG, that achieves a better dependence on variance upper bounds by combining NPG with SARAH-style variance reduction. Notably, Assumption 1 is still required by their analysis. Note that these sample complexity results are not comparable with the ones for ϵ-FOSP, since ϵ and $\widetilde{\epsilon}$ measure two different things (gradient norms and performance gaps, respectively). Moreover, it is a counter-intuitive but well-known fact of optimization that fast convergence to $\widetilde{\epsilon}$-globaloptima does not necessarily imply fast convergence to ϵ-FOSP (e.g., [AZ18]). Indeed, [FBKH23] designed a first-order algorithm (N-PG-IGT) that, under Fisher non-degeneracy, converges to a globally optimal policy at a rate of $O\left(\widetilde{\epsilon}^{-2.5}\right)$ without using importance sampling. However, its rate to ϵ-FOSP is $O\left(\epsilon^{-3.5}\right)$, even worse than SVRPG. They also proposed a second-order algorithm (N-HARPG) with $O\left(\widetilde{\epsilon}^{-2}\right)$ global convergence that matches the $O\left(\epsilon^{-3}\right)$ FOSP convergence of HAPG and SHARP. Several other works consider global convergence, but a further discussion would be out of scope for this paper. Refer to [FBKH23] for recent coverage of the topic. The classic $\Omega\left(\widetilde{\epsilon}^{-2}\right)$ lower bound by [GAMK13] holds for this setting.

[1] Curiously, [YZZ+22] claim $O\left(\epsilon^{-3}\right)$ sample complexity for their VRMPO algorithm without using importance weights nor second-order information. Our understanding is that their proof is affected by a subtle mistake where the distribution shift is ignored without any further explanation (see Equation 51 from the proof of Theorem 4.1 in their paper).