Sample Efficient Reinforcement Learning with REINFORCE

Junzi Zhang[1], **Jongho Kim**[2], **Brendan O'Donoghue**[3], **Stephen Boyd**[2]
[1] Institute for Computational & Mathematical Engineering, Stanford University, USA
[2] Department of Electrical Engineering, Stanford University, USA
[3] DeepMind, Google
junziz@stanford.edu, jkim22@stanford.edu, bodonoghue85@gmail.com, boyd@stanford.edu

Abstract

Policy gradient methods are among the most effective methods for large-scale reinforcement learning, and their empirical success has prompted several works that develop the foundation of their global convergence theory. However, prior works have either required exact gradients or state-action visitation measure based mini-batch stochastic gradients with a *diverging* batch size, which limit their applicability in practical scenarios. In this paper, we consider classical policy gradient methods that compute an approximate gradient with a *single* trajectory or a *fixed size* mini-batch of trajectories under soft-max parametrization and log-barrier regularization, along with the widely-used REINFORCE gradient estimation procedure. By controlling the number of "bad" episodes and resorting to the classical doubling trick, we establish an anytime sub-linear high probability regret bound as well as almost sure global convergence of the average regret with an asymptotically sub-linear rate. These provide the first set of global convergence and sample efficiency results for the well-known REINFORCE algorithm and contribute to a better understanding of its performance in practice.

1 Introduction

In this paper, we study the global convergence rates of the REINFORCE algorithm (Williams 1992) for episodic reinforcement learning. REINFORCE is a vanilla policy gradient method that computes a stochastic approximate gradient with a single trajectory or a fixed size mini-batch of trajectories with particular choice of gradient estimator, where we use 'vanilla' here to disambiguate the method from more exotic variants such as natural policy gradient methods. REINFORCE and its variants are among the most widely used policy gradient methods in practice due to their good empirical performance and implementation simplicity (Mnih and Gregor 2014; Gu et al. 2015; Zoph and Le 2016; Rennie et al. 2017; Guu et al. 2017; Johnson et al. 2017; Yi et al. 2018; Kool, van Hoof, and Welling 2018, 2020). Related methods include the actor-critic family (Konda and Tsitsiklis 2003; Mnih et al. 2016) and deterministic and trust-region based variants (Silver et al. 2014; Schulman et al. 2017, 2015).

The theoretical results for policy gradient methods have, up to recently, been restricted to convergence to local stationary points (Agarwal et al. 2019). Lately, a series of works have established *global* convergence results. These recent developments cover a broad range of issues including global optimality characterization (Fazel et al. 2018; Bhandari and Russo 2019), convergence rates (Zhang et al. 2019; Mei et al. 2020; Bhandari and Russo 2020; Cen et al. 2020), the use of function approximation (Agarwal et al. 2019; Wang et al. 2019; Fu, Yang, and Wang 2020), and efficient exploration (Agarwal et al. 2020) (for more details, see the related work section, which we defer to the longer version of this paper (Zhang et al. 2020, Appendix E) due to space limits). Nevertheless, prior work on vanilla policy gradient methods either requires exact and deterministic policy gradients or only guarantees convergence up to $\Theta(1/M^p)$ with a fixed mini-batch size $M > 0$ of trajectories collected when performing a single update (where $p > 0$ is $1/2$ in most cases), while global convergence is only achieved when the batch size M goes to infinity. By contrast, practical implementations of policy gradient methods typically use either a single or a fixed number of sample trajectories, which tends to perform well. In addition, prior theoretical results (for general MDPs) have used the state-action visitation measure based gradient estimation (see *e.g.*, (Wang et al. 2019, (3.10))), which are typically not used in practice.

The main purpose of this paper is to bridge this gap between theory and practice. We do this in two major ways. First, we derive performance bounds for the case of a fixed mini-batch size, rather than requiring diverging size. Second, we remove the need for the state-action visitation measure based gradient, instead using the REINFORCE gradient estimator. It is nontrivial to go from a diverging mini-batch size to a fixed one. In fact, by allowing for an arbitrarily large batch size, existing works in the literature were able to make use of IID samples to decouple the analysis into deterministic gradient descent/ascent and error control of stochastic gradient estimations. In contrast, with a single trajectory or a fixed batch size, such a decoupling is no longer feasible. In addition, the state-action visitation measure based gradient estimations are unbiased and unbounded, while REINFORCE gradient estimations are biased and bounded. Hence a key to the analysis is to deal with the bias while making better use of the boundedness. Our analysis not only addresses these challenges, but also leads to convergence results in almost sure and high probability senses, which are stronger than the expected convergence results that dominate

the literature (for vanilla policy gradient methods). We also emphasize that the goal of this work is to provide a deeper understanding of a widely used algorithm, REINFORCE, with little or no modifications, rather than tweaking it to achieve near-optimal performance bounds. Lastly, our analysis is not the complete picture and several open questions about the performance of policy gradient methods remain. We discuss these issues in the conclusion.

1.1 Contribution

Our major contribution can be summarized as follows. We establish the first set of global convergence results for the REINFORCE algorithm. In particular, we establish an anytime sub-linear high probability regret bound as well as almost sure global convergence of the average regret with an asymptotically sub-linear rate for REINFORCE, showing that the algorithm is sample efficient (*i.e.*, with polynomial/non-exponential complexity). To our knowledge, these (almost sure and high probability) results are stronger than existing global convergence results for (vanilla) policy gradient methods in the literature. Moreover, our convergence results remove the non-vanishing $\Theta(1/M^p)$ term (with $M > 0$ being the mini-batch size of the trajectories and $p > 0$ being some constant exponent) and hence show for the first time that policy gradient estimations with a single or finite number of trajectories also enjoy global convergence properties. Finally, the widely-used REINFORCE gradient estimation procedure is studied, as opposed to the state-action visitation measure based estimators typically studied in the literature but rarely used in practice.

2 Problem Setting and Preliminaries

Below we begin with our problem setting and some preliminaries on MDPs and policy optimization. For brevity we restrict ourselves to the stationary infinite-horizon discounted setting. We briefly discuss potential extensions beyond this setting in §6.

2.1 Problem Setting

We consider a finite MDP $\mathcal{M}$, which is characterized by a finite state space $\mathcal{S} = \{1, \ldots, S\}$, a finite action space $\mathcal{A} = \{1, \ldots, A\}$, a transition probability p (with $p(s'|s,a)$ being the probability of transitioning to state s' given the current state s and action a), a reward function r (with $r(s,a)$ being the instantaneous reward when taking action a at state s), a discount factor $\gamma \in [0,1)$ and an initial state distribution $\rho \in \Delta(\mathcal{S})$. Here $\Delta(\mathcal{X})$ denotes the probability simplex over a finite set $\mathcal{X}$. A (stationary, stochastic) policy π is a mapping from $\mathcal{S}$ to $\Delta(\mathcal{A})$. We will use $\pi(a|s)$, $\pi(s,a)$ or $\pi_{s,a}$ alternatively to denote the probability of taking action a at state s following policy π. The policy π can also be viewed as an SA dimensional vector in

$$\Pi = \Big\{ \pi \in \mathbf{R}^{SA} \,\Big|\, \sum\nolimits_{a=1}^{A} \pi_{s,a} = 1 \,(\forall s \in \mathcal{S}), \; \pi_{s,a} \geq 0 \,(\forall s \in \mathcal{S},\, a \in \mathcal{A}) \Big\}. \tag{1}$$

Notice that here we use the double indices s and a for notational convenience. We use $\pi(s,\cdot) \in \mathbf{R}^A$ to denote the subvector $(\pi(s,1), \ldots, \pi(s,A))$. We also assume that $r(s,a)$ is deterministic for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$ for simplicity, although our results hold for any r with an almost sure uniform bound. Here r can be similarly viewed as an SA-dimensional vector. Without loss of generality, we assume that $r(s,a) \in [0,1]$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, which is a common assumption (Jaksch, Ortner, and Auer 2010; Agarwal et al. 2019; Mei et al. 2020; Even-Dar and Mansour 2003; Jin et al. 2018). We also assume that ρ is component-wise positive, as is assumed in (Bhandari and Russo 2019).

Given a policy $\pi \in \Pi$, the expected cumulative reward of the MDP is defined as

$$F(\pi) = \mathbf{E} \sum\nolimits_{t=0}^{\infty} \gamma^t r(s_t, a_t), \tag{2}$$

where $s_0 \sim \rho$, $a_t \sim \pi(\cdot|s_t)$, $s_{t+1} \sim p(\cdot|s_t, a_t)$, $\forall t \geq 0$, and the goal is to find a policy π which solves the following optimization problem:

$$\text{maximize}_{\pi \in \Pi} \quad F(\pi). \tag{3}$$

Any policy $\pi^\star \in \operatorname{argmax}_{\pi \in \Pi} F(\pi)$ is said to be optimal, and the corresponding optimal objective value is denoted as $F^\star = F(\pi^\star)$. Note that in the literature, $F(\pi)$ is also commonly written as V_ρ^π and referred to as the value function. Here we hide the dependency on ρ as it is fixed throughout the paper.

2.2 Vanilla Policy Gradient Method and REINFORCE Algorithm

When the transition probability p and reward r are fully known, problem (3) reduces to solving an MDP, in which case various classical algorithms are available, including value iteration and policy iteration (Bertsekas 2017). In this paper, we consider the episodic reinforcement learning setting in which the agent accesses p and r by interacting with the environment over successive episodes, *i.e.*, the agent accesses the environment in the form of a ρ-restart model (Shani, Efroni, and Mannor 2019), which is commonly adopted in the policy gradient literature (Kakade et al. 2003). In addition, we focus on the REINFORCE algorithm, a representative policy gradient method.

Policy parametrization and surrogate objectives. Here we consider parametrizing the policy with parameter $\theta \in \Theta$, *i.e.*, $\pi_\theta : \Theta \to \Pi$, and take the following (regularized) optimization problem as an approximation to (3):

$$\text{maximize}_{\theta \in \Theta} \quad L_\lambda(\theta) = F(\pi_\theta) + \lambda R(\theta), \tag{4}$$

where $\lambda \geq 0$ and $R : \Theta \to \mathbf{R}$ is a differentiable regularization term that improves convergence, to be specified later. Although our ultimate goal is still to solve the original problem (3) this regularized optimization problem is a useful surrogate and our approach will be to tackle problem (4) with progressively smaller λ regularization penalties, thereby converging to solving the actual problem we care about.

Policy gradient method. In each episode n, the policy gradient method directly performs an online stochastic gradient ascent update on a surrogate objective $L_{\lambda^n}(\theta)$, *i.e.*,

$$\theta^{n+1} = \theta^n + \alpha^n \widehat{\nabla}_\theta L_{\lambda^n}(\theta^n), \tag{5}$$

Algorithm 1 Policy Gradient Method with Single Trajectory Estimates

Input: initial parameter θ^0, step-sizes α^n and regularization parameters λ^n ($n \geq 0$).
for $n = 0, 1, \ldots$ **do**
  Choose H^n, sample trajectory τ^n from $\mathcal{M}$ following policy π_{θ^n}, and compute an approximate gradient $\widehat{\nabla}_\theta L_{\lambda^n}(\theta^n)$ of L_{λ^n} using trajectory τ^n.
  Update $\theta^{n+1} = \theta^n + \alpha^n \widehat{\nabla}_\theta L_{\lambda^n}(\theta^n)$.
end for

where α^n is the step-size and λ^n is the regularization parameter. Here the stochastic gradient $\widehat{\nabla}_\theta L_{\lambda^n}(\theta^n)$ is computed by sampling a *single* trajectory τ^n following policy π_{θ^n} from $\mathcal{M}$ with the initial state distribution ρ. Here $\tau^n = (s_0^n, a_0^n, r_0^n, s_1^n, a_1^n, r_1^n, \ldots, s_{H^n}^n, a_{H^n}^n, r_{H^n}^n)$, where H^n is a finite (and potentially random) stopping time of the trajectory (to be specified below), $s_0^n \sim \rho$, $a_t^n \sim \pi_{\theta^n}(\cdot|s_t^n)$, $s_{t+1}^n \sim p(\cdot|s_t^n, a_t^n)$ and $r_t^n = r(s_t^n, a_t^n)$ for all $t = 0, \ldots, H^n$. We summarize the generic policy gradient method (with single trajectory gradient estimates) in Algorithm 1. An extension to mini-batch scenarios will be discussed in §5. As is always (implicitly) assumed in the literature of episodic reinforcement learning (*e.g.*, *cf.* (Marbach and Tsitsiklis 2001)), given the current policy, we assume that the sampled trajectory is conditionally independent of all previous policies and trajectories.

REINFORCE algorithm. There are several ways of choosing the stochastic gradient operator $\widehat{\nabla}_\theta$ in the policy gradient method, and the well-known REINFORCE algorithm (Williams 1992) corresponds to a specific family of estimators based on the policy gradient theorem (Sutton et al. 2000) (*cf.* §3). Other common alternatives include zeroth order/random search (Fazel et al. 2018; Malik et al. 2018) and actor-critic (Konda and Tsitsiklis 2003) approximations. One may also choose to parametrize the policy as a mapping from the parameter space to a specific action, which would then result in deterministic policy gradient approximations (Silver et al. 2014).

Although our main goal is to study the REINFORCE algorithm, our analysis indeed holds for rather generic stochastic gradient estimates. In the next section, we introduce the (mild) assumptions needed for our convergence analysis and the detailed gradient estimation procedures in the REINFORCE algorithm, and then verify that the assumptions do hold for these gradient estimations.

2.3 Phased Learning and Performance Criteria

Phased learning. To facilitate the exposition below, we divide the optimization in Algorithm 1 into successive phases $l = 0, 1, \ldots$, each with length $T_l > 0$. We then fix the regularization coefficient λ_l within each phase $l \geq 0$. In addition, a post-processing step is enforced at the end of each phase to produce the initialization of the next phase. The resulting algorithm is described in Algorithm 2. Here the trajectory is denoted as $\tau^{l,k} = (s_0^{l,k}, a_0^{l,k}, r_0^{l,k}, \ldots, s_{H^{l,k},l}^{l,k}, a_{H^{k,l},k}^{l,k}, r_{H^{l,k}}^{l,k})$, and we will refer to $\theta^{l,k}$ as the (l,k)-th iterate hereafter. The post-processing function is required to guarantee that the resulting policy π_θ is lower bounded by a pre-specified tolerance $\epsilon_{\rm pp} \in (0, 1/A]$ to ensure that the regularization is bounded (*cf.* Algorithm 3 for a formal description and §3.1 for an example realization).

Algorithm 2 Phased Policy Gradient Method

Input: initial parameter $\tilde{\theta}^{0,0}$, step-sizes $\alpha^{l,k}$, regularization parameters λ^l, phase lengths T_l ($l,\ k \geq 0$) and post-processing tolerance $\epsilon_{\rm pp} \in (0, 1/A]$.
Set $\theta^{0,0} = \texttt{PostProcess}(\tilde{\theta}^{0,0}, \epsilon_{\rm pp})$.
for phase $l = 0, 1, 2, \ldots$ **do**
  for episode $k = 0, 1, \ldots, T_l - 1$ **do**
    Choose $H^{l,k}$, sample trajectory $\tau^{l,k}$ from $\mathcal{M}$ following policy $\pi_{\theta^{l,k}}$, and compute an approximate gradient $\widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})$ of L_{λ^l} using trajectory $\tau^{l,k}$.
    Update $\theta^{l,k+1} = \theta^{l,k} + \alpha^{l,k} \widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})$.
  end for
  Set $\theta^{l+1,0} = \texttt{PostProcess}(\theta^{l,T_l}, \epsilon_{\rm pp})$.
end for

Note that here the k-th episode in phase l corresponds to the n-th episode in the original indexing with $n = \sum_{j=0}^{l-1} T_j + k$. For notational compactness below, for $\mathcal{T} = \{T_j\}_{j=0}^{\infty}$, we define $B_{\mathcal{T}} : \mathbf{Z}_+ \times \mathbf{Z}_+ \to \mathbf{Z}_+$, where $B_{\mathcal{T}}(l,k) = \sum_{j=0}^{l-1} T_j + k$ maps the double index (l,k) to the corresponding original episode number, with $\mathbf{dom}\, B_{\mathcal{T}} = \{(l,k) \in \mathbf{Z}_+ \times \mathbf{Z}_+ \,|\, 0 \leq k \leq T_l - 1\}$. The mapping $B_{\mathcal{T}}$ is a bijection and we denote its inverse by $G_{\mathcal{T}}$.

Algorithm 3 $\texttt{PostProcess}(\theta, \epsilon_{\rm pp})$

Input: $\epsilon_{\rm pp} \in (0, 1/A]$, $\theta \in \Theta$.
Return θ' (near θ) such that $\pi_{\theta'}(s,a) \geq \epsilon_{\rm pp}$ for each $s,\ a \in \mathcal{S} \times \mathcal{A}$.

Performance criteria. The criterion we adopt to evaluate the performance of Algorithm 2 is *regret*. For any $N \geq 0$, the regret up to episode N is defined as the cumulative sub-optimality of the policy over the N episodes. Formally, we define

$$\mathbf{regret}(N) = \sum\nolimits_{\{(l,k)|B_{\mathcal{T}}(l,k) \leq N\}} F^\star - \hat{F}^{l,k}(\pi_{\theta^{l,k}}). \quad (6)$$

Here the summation is over all (l,k)-th iterates whose corresponding original episode numbers are smaller or equal to N, and $\hat{F}^{l,k}(\pi_{\theta^{l,k}}) = \mathbf{E}_{l,k} \sum_{t=0}^{H^{l,k}} \gamma^t r(s_t^{l,k}, a_t^{l,k})$, where $s_0 \sim \rho$, $a_t^{l,k} \sim \pi_{\theta^{l,k}}(\cdot|s_t^{l,k})$, $s_{t+1}^{l,k} \sim p(\cdot|s_t^{l,k}, a_t^{l,k})$,

$\forall t \geq 0$, and $\mathbf{E}_{l,k}$ denotes the conditional expectation given the (l,k)-th iteration $\theta^{l,k}$. Notice that the regret defined above takes into account the fact that the trajectories are stopped/truncated to have finite horizons $H^{l,k}$, which characterizes the actual loss caused by sampling the trajectories in line 5 of Algorithm 2. A similar regret definition for the episodic (discounted) reinforcement learning setting considered here is adopted in (Fu, Yang, and Wang 2020). We remark that all our regret bounds remain correct up to lower order terms when we replace $\hat{F}^{l,k}$ with F or an expectation-free version.

Similarly, we also define the single phase version of regret as follows. The regret up to episode $K \in \{0, \ldots, T_l - 1\}$ in phase l is defined as

$$\mathbf{regret}_l(K) = \sum\nolimits_{k=0}^{K} F^\star - \hat{F}^{l,k}(\pi_{\theta^{l,k}}). \tag{7}$$

Notice that (6) and (7) are connected via

$$\mathbf{regret}(N) = \sum_{l=0}^{l_N - 1} \mathbf{regret}_l(T_l - 1) + \mathbf{regret}_{l_N}(k_N), \tag{8}$$

where $(l_N, k_N) = G_{\mathcal{T}}(N)$.

We provide high probability regret bounds in §4. We remark that a regret bound of the form $\mathbf{regret}(N)/(N+1) \leq R$ (for some $R > 0$) immediately implies that $\min_{l,k:\, B_{\mathcal{T}}(l,k) \leq N} F^\star - F(\pi_{\theta^{l,k}}) \leq R$, where the latter is also a commonly adopted performance criteria in the literature (Agarwal et al. 2019; Wang et al. 2019).

3 Assumptions and REINFORCE Gradients

3.1 Assumptions

Here we list a few fundamental assumptions that we require for our analysis.

Assumption 1 (Setting)**.** *The regularization term is a log-barrier, i.e.,*

$$R(\theta) = \tfrac{1}{SA} \sum\nolimits_{s \in \mathcal{S}, a \in \mathcal{A}} \log(\pi_\theta(s,a)),$$

and the policy is parametrized to be a soft-max, i.e., $\pi_\theta(s,a) = \frac{\exp(\theta_{s,a})}{\sum_{a' \in \mathcal{A}} \exp(\theta_{s,a'})}$*, with* $\Theta = \mathbf{R}^{SA}$.

The first assumption concerns the form of the policy parameterization and the regularization. Notice that the regularization term here can also be seen as a relative entropy/KL regularization (with a uniform distribution policy reference) (Agarwal et al. 2019). Such kind of regularization terms are also widely adopted in practice (although typically with variations) (Peters, Mulling, and Altun 2010; Schulman, Chen, and Abbeel 2017).

With Assumption 1, the post-processing function in Algorithm 3 can be for example realized by first calculating $\hat{\pi} = \epsilon_{\text{pp}} \mathbf{1} + (1 - A\epsilon_{\text{pp}})\pi_\theta$, and then return θ' with $\theta'_{s,a} = \log \hat{\pi}_{s,a} + c_s$. Here $\mathbf{1}$ is an all-one vector and $c_s \in \mathbf{R}$ $(s = 1, \ldots, S)$ are arbitrary real numbers.

Assumption 2 (Policy gradient estimator)**.** *There exist constants* C, C_1, C_2, M_1, $M_2 > 0$*, such that for all* l, $k \geq 0$*, we have* $\|\widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})\|_2 \leq C_1$ *almost surely and that*

$$\nabla_\theta L_{\lambda^l}(\theta^{l,k})^T \mathbf{E}_{l,k} \widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k}) \geq C_2 \|\nabla_\theta L_{\lambda^l}(\theta^{l,k})\|_2^2 - \delta_{l,k},$$
$$\mathbf{E}_{l,k} \|\widehat{\nabla}_\theta L_{\lambda^k}(\theta^{l,k})\|_2^2 \leq M_1 + M_2 \|\nabla_\theta L_{\lambda^l}(\theta^{l,k})\|_2^2,$$

where $\sum_{k=0}^{T_l - 1} \delta_{l,k}^2 \leq C$*,* $\forall\ l \geq 0$*. In addition,* $H^{l,k} \geq \log_{1/\gamma}(k+1)$*,* $\forall\ l,\ k \geq 0$.

The second assumption requires that the gradient estimates are almost surely bounded, nearly unbiased and satisfy a bounded second-order moment growth condition. This is a slight generalization of standard assumptions in the stochastic gradient descent literature (Bottou, Curtis, and Nocedal 2018). Additionally, we also require that the trajectory lengths $H^{l,k}$ are at least logarithmically growing in k to control the loss of rewards due to truncation. For notational simplicity, hereafter we omit to mention the trajectory sampling (*i.e.*, $s_0 \sim \rho$, $a_t^{l,k} \sim \pi_{\theta^{l,k}}(\cdot | s_t^{l,k})$, $s_{t+1}^{l,k} \sim p(\cdot | s_t^{l,k}, a_t^{l,k})$, $\forall t \geq 0$) when we write down $\mathbf{E}_{l,k}$.

Notice that Assumption 2 immediately holds if $\widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})$ is unbiased and has a bounded second-order moment. We have implicitly assumed that L_λ is differentiable, which we can do due to the following lemma:

Proposition 1 ((Agarwal et al. 2019, Lemma E.4))**.** *Under Assumption 1,* L_λ *is strongly smooth with parameter* $\beta_\lambda = \frac{8}{(1-\gamma)^3} + \frac{2\lambda}{S}$*, i.e.,* $\|\nabla_\theta L_\lambda(\theta) - \nabla_\theta L_\lambda(\theta')\|_2 \leq \beta_\lambda \|\theta - \theta'\|_2$ *for any* θ, $\theta' \in \mathbf{R}^{SA}$.

3.2 REINFORCE Gradient Estimations

Now we introduce REINFORCE gradient estimation with baselines, and specify the hyper-parameters under which the technical Assumption 2 holds, when operating under the setting Assumption 1.

REINFORCE gradient estimation with log-regularization takes the following form:

$$\begin{aligned} \widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k}) = & \sum\nolimits_{t=0}^{\lfloor \beta H^{l,k} \rfloor} \gamma^t (\widehat{Q}^{l,k}(s_t^{l,k}, a_t^{l,k}) - b(s_t^{l,k})) \\ & \times \nabla_\theta \log \pi_{\theta^{l,k}}(a_t^{l,k} | s_t^{l,k}) \\ & + \tfrac{\lambda^l}{SA} \sum\nolimits_{s \in \mathcal{S}, a \in \mathcal{A}} \nabla_\theta \log \pi_{\theta^{l,k}}(a|s), \end{aligned} \tag{9}$$

where $\beta \in (0,1)$, $\widehat{Q}^{l,k}(s_t^{l,k}, a_t^{l,k}) = \sum_{t'=t}^{H^{l,k}} \gamma^{t'-t} r_{t'}^{l,k}$, and the second term above corresponds to the gradient of the regularization $R(\theta)$. Notice that here the outer summation is only up to $\lfloor \beta H^{l,k} \rfloor$, which ensures that $\hat{Q}^{l,k}(s_t^{l,k}, a_t^{l,k})$ is sufficiently accurate. Here $b : \mathcal{S} \to \mathbf{R}$ is called the baseline, and is required to be independent of the trajectory $\tau^{l,k}$ (Agarwal, Jiang, and Kakade 2019, §4.1). The purpose of subtracting b from the approximate Q-values is to (potentially) reduce the variance of the "plain" REINFORCE gradient estimation, which corresponds to the case when $b = 0$.

With this we have the following result, the proof of which can be found in the Appendix in the longer version of this paper (Zhang et al. 2020).

Lemma 2. *Suppose that Assumption 1 holds,* $\beta \in (0,1)$*, and that for all* l, $k \geq 0$, $\lambda^l \leq \bar{\lambda}$*,*

$$H^{l,k} \geq \tfrac{2 \log_{1/\gamma}\left(\frac{8(k+1)}{(1-\gamma)^3}\right)}{3 \min\{\beta, 1-\beta\}} (= \Theta(\log(k+1))). \tag{10}$$

Assume in addition that $|b(s)| \leq B$ *for any* $s \in \mathcal{S}$*, where* $B > 0$ *is a constant. Then for the gradient estimation* (9)*, Assumption 2 holds with*

$$C = 16\Big(\tfrac{1}{(1-\gamma)^2} + \bar{\lambda}\Big)^2, \quad C_1 = \tfrac{2(1+B(1-\gamma))}{(1-\gamma)^2} + 2\bar{\lambda},$$
$$C_2 = 1, \quad M_1 = \tfrac{32}{(1-\gamma)^4} + \bar{V}_b, \quad M_2 = 2.$$

and $\delta_{l,k} = \Big(\tfrac{2}{(1-\gamma)^2} + 2\bar{\lambda}\Big)(k+1)^{-\frac{2}{3}}$*,* $\forall\ l,\ k \geq 0$*. Here* $\bar{V}_b \in \Big[0, 4\Big(\tfrac{1+B(1-\gamma)}{(1-\gamma)^2} + \bar{\lambda}\Big)^2\Big]$ *is the uniform upper bound on variances of policy gradient estimations with form* (9)*.*

This result extends without modification to non-stationary baselines $b_t^{l,k}$, as long as each $b_t^{l,k}$ is independent of trajectory $\tau^{l,k}$ and $|b_t^{l,k}(s)| \leq B$ for any $t,\ l,\ k \geq 0$. Note that the explicit upper bound on $\bar{V}_b$ is pessimistic, and in practice $\bar{V}_b$ is usually much smaller than $\bar{V}_0$ with appropriate choices of baselines (*e.g.*, the adaptive reinforcement baselines (Williams 1992; Zhao et al. 2011)), although the latter has a smaller upper bound as stated in Lemma 2.

4 Main Convergence Results

4.1 Preliminary Tools

We first present some preliminary tools for our analysis.

Non-convexity and control of "bad" episodes. One of the key difficulties in applying policy gradient methods to solve an MDP problem towards global optimality is that problem (3) is in general non-convex (Agarwal et al. 2019). Fortunately, we have the following result, which connects the gradient of the surrogate objective L_λ with the global optimality gap of the original optimization problem (3).

Proposition 3 ((Agarwal et al. 2019, Theorem 5.3))**.** *Under Assumption 1, for any* $\epsilon > 0$*, suppose that we have* $\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \epsilon$ *and that* $\epsilon \leq \lambda/(2SA)$*. Then* $F^\star - F(\pi_\theta) \leq \frac{2\lambda}{1-\gamma} \left\| \frac{d_\rho^{\pi^\star}}{\rho} \right\|_\infty$.

Here for any policy $\pi \in \Pi$, $d_\rho^\pi = (1-\gamma)\sum_{t=0}^\infty \gamma^t \mathbf{Prob}_\pi(s_t = s | s_0 \sim \rho)$ is the discounted state visitation distribution, where $\mathbf{Prob}_\pi(s_t = s | s_0 \sim \rho)$ is the probability of arriving at s in step t starting from $s_0 \sim \rho$ following policy π in $\mathcal{M}$. In addition, the division in $d_\rho^{\pi^\star}/\rho$ is component-wise.

Now motivated by Proposition 3, when analyzing the regret up to episode K in phase l, we define the following set of "bad episodes":

$$I^+ = \{k \in \{0, \ldots, K\} \mid \|\nabla_\theta L_{\lambda^l}(\theta^{l,k})\|_2 \geq \lambda^l/(2SA)\}.$$

Then one can show that for any $\epsilon > 0$, if we choose $\lambda^l = \frac{\epsilon(1-\gamma)}{2}$, we have that $F^\star - F(\pi_{\theta^{l,k}}) \leq \|d_\rho^{\pi^\star}/\rho\|_\infty \epsilon$ for any $k \in \{0, \ldots, K\} \backslash I^+$, while $F^\star - F(\pi_{\theta^{l,k}}) \leq 1/(1-\gamma)$ holds trivially for $k \in I^+$ due to the assumption that the rewards are between 0 and 1. We then establish a sub-linear (in K) bound the size of I^+, which serves as the key stepping stone for the overall sub-linear regret bound. The details of these arguments can be found in the Appendix in the longer version of this paper (Zhang et al. 2020).

Doubling trick. The second tool is a classical doubling trick that is commonly adopted in the design of online learning algorithms (Besson and Kaufmann 2018; Basei, Guo, and Hu 2020), which can be used to stitch together the regret over multiple learning phases in Algorithm 2.

Notice that Proposition 3 suggests that for any prespecified tolerance ϵ, one can select λ proportional to ϵ and then run (stochastic) gradient ascent to drive $F^\star - F(\pi_\theta)$ below the tolerance. To obtain the eventual convergence and regret bound in the long run we apply the doubling trick, which specifies a growing phase length sequence with $T_{l+1} \approx 2T_l$ in Algorithm 2 and a suitably decaying sequence of regularization parameters $\{\lambda^l\}_{l=0}^\infty$.

From high probability to almost sure convergence. The last tool is an observation that an arbitrary anytime sublinear high probability regret bound with logarithmic dependency on $1/\delta$ immediately leads to almost sure convergence of the average regret with a corresponding asymptotic rate. Although such an observation seems to be informally well-known in the theoretical computer science community, we provide a compact formal discussion below for self-contained-ness.

Lemma 4. *Suppose that* $\forall\ \delta \in (0,1)$*, with probability at least* $1-\delta$*,* $\forall\ N \geq 0$*, we have*

$$\mathbf{regret}(N) \leq d_1(N+c)^{d_2}(\log(N/\delta))^{d_3} + d_4(\log N)^{d_5} \quad (11)$$

for some constants $c,\ d_1,\ d_3,\ d_4,\ d_5 \geq 0$ *and* $d_2 \in [0,1)$*. Then we also have*

$$\mathbf{Prob}\left(\exists\ \bar{N} \in \mathbf{Z}_+, \text{ such that } \forall\ N \geq \bar{N},\ A_N \text{ holds}\right) = 1,$$

where the events $A_N = \{\mathbf{regret}(N)/(N+1) \leq (*)\}$*, and*

$$(*) = d_1 N^{-(1-d_2)}\left(1 + \tfrac{c}{N}\right)^{d_2}(3\log N)^{d_3} + \tfrac{d_4(\log N)^{d_5}}{N}.$$

To put it another way, we have

$$\lim_{N\to\infty} \mathbf{regret}(N)/(N+1) = 0 \quad \textit{almost surely}$$

with an asymptotic rate of $(*)$*.*

Notice that here we restrict the right-hand side of (11) to a rather specific form simply because our bounds below are all of this form. However, similar results hold for much more general forms of bounds.

4.2 Regret Analysis

In this section, we establish the regret bound of Algorithm 2, when used with the REINFORCE gradient estimator from §3.2. We begin by bounding the regret of a single phase and then use the doubling trick to combine these into the overall regret bound.

Single phase analysis. We begin by bounding the regret defined in (7) of each phase in Algorithm 2. Note that a single phase in Algorithm 2 is exactly Algorithm 1 terminated in episode T_l, with $\lambda^n = \lambda^l$ for all $n \geq 0$ and $\theta^0 = \theta^{l,0}$. Also notice that for a given phase $l \geq 0$, in order for Theorem 5 below to hold, we actually only need the conditions in Assumption 2 to be satisfied for this specific l.

Theorem 5. *Under Assumptions 1 and 2, for phase $l \geq 0$ suppose that we choose $\alpha^{l,k} = C_{l,\alpha}\frac{1}{\sqrt{k+3}\log_2(k+3)}$ for some $C_{l,\alpha} \in (0, C_2/(M_2\beta_{\lambda^l})]$. Then for any $\epsilon > 0$, if we choose $\lambda^l = \frac{\epsilon(1-\gamma)}{2}$, then $\forall\ \delta \in (0,1)$, with probability at least $1-\delta$, for any $K \in \{0, \ldots, T_l - 1\}$, we have*

$$\begin{aligned}\mathbf{regret}_l(K) \leq U_1 \frac{\sqrt{K+1}\log_2(K+3)\sqrt{\log(2/\delta)}}{\epsilon^2} \\ + (K+1)\left\|\frac{d_\rho^{\pi^\star}}{\rho}\right\|_\infty \epsilon + \frac{2\gamma}{1-\gamma}\log(K+3).\end{aligned} \tag{12}$$

Here the constant U_1 only depends on the underlying MDP $\mathcal{M}$, phase initialization $\theta^{l,0}$ and the constants C, C_1, C_2, M_1, $C_{l,\alpha}$, λ^l.

The proof as well as a more formal statement of Theorem 5 with details of the constants (*cf.* Theorem 9) are deferred to the Appendix in the longer version of this paper (Zhang et al. 2020). Here the constant β_λ is the smoothness constant from Proposition 1. We remark that when ϵ is fixed, the regret bound (12) can be seen as a sub-linear (in K as $K \to \infty$) regret term plus an error term $(K+1)\epsilon + \frac{2\gamma}{1-\gamma}\log(K+3)$. Alternatively, one can interpret it as follows:

$$\begin{aligned}\frac{\mathbf{regret}_l(K)}{K+1} \leq & U_1 \frac{\log_2(K+3)\sqrt{\log(2/\delta)}}{\sqrt{K+1}\epsilon^2} \\ & + \frac{2\gamma}{1-\gamma}\frac{\log(K+3)}{K+1} + \left\|\frac{d_\rho^{\pi^\star}}{\rho}\right\|_\infty \epsilon.\end{aligned}$$

Namely, the average regret in episode l converges to a constant multiple of the pre-specified tolerance ϵ at a sub-linear rate (as $K \to \infty$).

Overall regret bound. Now we stitch together the single phase regret bounds established above to obtain the overall regret bound of Algorithm 2, with the help of the doubling trick. This leads to the following theorem.

Theorem 6 (Regret for REINFORCE)**.** *Under Assumption 1, suppose that for each $l \geq 0$, we choose $\alpha^{l,k} = C_{l,\alpha}\frac{1}{\sqrt{k+3}\log_2(k+3)}$, with $C_{l,\alpha} \in [1/(2\beta_{\bar\lambda}), 1/(2\beta_{\lambda^l})]$ and $\bar\lambda = \frac{1-\gamma}{2}$, and choose $T_l = 2^l$, $\epsilon^l = T_l^{-1/6} = 2^{-l/6}$, $\lambda^l = \frac{\epsilon^l(1-\gamma)}{2}$ and $\epsilon_{\mathrm{pp}} = 1/(2A)$. In addition, suppose that (9) is adopted to evaluate $\widehat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})$, with $\beta \in (0,1)$, $|b(s)| \leq B$ for any $s \in \mathcal{S}$ (where $B > 0$ is a constant), and that (10) holds for $H^{l,k}$ for all l, $k \geq 0$. Then we have for any $\delta \in (0,1)$, with probability at least $1-\delta$, for any $N \geq 0$, we have*

$$\mathbf{regret}(N) = O\Big(\Big(\tfrac{S^2A^2}{(1-\gamma)^7} + \left\|\tfrac{d_\rho^{\pi^\star}}{\rho}\right\|_\infty\Big) N^{\frac{5}{6}} (\log\tfrac{N}{\delta})^{\frac{5}{2}}\Big). \tag{13}$$

In addition, we have

$$\lim_{N\to\infty} \mathbf{regret}(N)/(N+1) = 0 \quad \textit{almost surely} \tag{14}$$

with asymptotic rate $O\Big(\Big(\tfrac{S^2A^2}{(1-\gamma)^7} + \left\|\tfrac{d_\rho^{\pi^\star}}{\rho}\right\|_\infty\Big) N^{-\frac{1}{6}} (\log N)^{\frac{5}{2}}\Big)$.

Note that the almost sure convergence (14) is immediately implied by the high probability bound (13) via Lemma 4. Here for clarity, we have restricted the statement to the case when we use the REINFORCE gradient estimation from §3.2. A more general counterpart result can be found in Appendix B.3 in the longer version of this paper (Zhang et al. 2020), from which Theorem 6 is immediately implied. See also (Zhang et al. 2020, Appendix C) for a more formal statement of the regret bound (*cf.* Corollary 11) for REINFORCE with detailed constants.

Notice that compared to the single phase regret bound in (12), the overall regret bound in (13) now gets rid of the dependency on a pre-specified tolerance $\epsilon > 0$. This should be attributed to the adaptivity in the regularization parameter sequence. Also notice that here we have followed the convention of the reinforcement learning literature to make all the problem dependent quantities (*e.g.*, γ, S, A, etc.) explicit in the big-O notation.

One crucial difference between our regret bound and those in the existing literature of vanilla policy gradient methods in the general MDP settings (which are sometimes not stated in the form of regret, but can be easily deduced from their proofs in those cases) is that the previous results either require exact and deterministic updates or contain a non-vanishing $\Theta(1/M^p)$ term, with M being the mini-batch size (of the trajectories) and $p > 0$ being some exponent (with a typical value of $1/2$). By removing such non-vanishing terms, we obtain the first sub-linear regret bound for model-free vanilla policy gradient methods with finite mini-batch sizes.

5 Extension to Mini-Batch Updates

We now consider extending our previous results to mini-batch settings, by modifying Algorithm 2 as follows. Firstly, in each inner iteration, instead of sampling only one trajectory in line 5, we sample $M \geq 1$ independent trajectories $\tau_1^{l,k}, \ldots, \tau_M^{l,k}$ from $\mathcal{M}$ following policy $\pi_{\theta^{l,k}}$ and then compute an approximate gradient $\widehat{\nabla}_\theta^{(i)} L_{\lambda^l}(\theta^{l,k})$ $(i = 1, \ldots, M)$ using each of these M trajectories. We then modify the update in line 6 as

$$\theta^{l,k+1} = \theta^{l,k} + \alpha^{l,k}\frac{1}{M}\sum_{i=1}^{M} \widehat{\nabla}_\theta^{(i)} L_{\lambda^l}(\theta^{l,k}).$$

See Algorithm 4 in Appendix D in the longer version of this paper (Zhang et al. 2020) for a formal description of the modified algorithm.

Regret with mini-batches. Notice that since each inner iteration (in Algorithm 4) now consists of M episodes, we need to slightly modify the definition of the regret up to episode N ($N \geq 0$) as follows:

$$\begin{aligned}&\mathbf{regret}(N; M) = \\ &\quad \sum\nolimits_{\{(l,k)\,|\,B_{\mathcal{T}}(l,k) \leq \lfloor \frac{N}{M} \rfloor - 1\}} M(F^\star - \hat F^{l,k}(\pi_{\theta^{l,k}})) \\ &\quad + \left(N - M\left\lfloor\frac{N}{M}\right\rfloor\right)(F^\star - \hat F^{l,k}(\pi_{\theta^{l_{N,M},k_{N,M}}})),\end{aligned} \tag{15}$$

where $(l_{N,M}, k_{N,M}) = G_{\mathcal{T}}(\lfloor N/M \rfloor)$ and $\hat F^{l,k}(\pi_{\theta^{l,k}})$ is the same as in (6). The above definition accounts for the fact

that each of the M episodes in an inner iteration/step (l,k) corresponds to the same iterate $\theta^{l,k}$ and hence has the same contribution to the regret. The second term on the right-hand side accounts for the contribution of the (remaining) $N - M\lfloor N/M \rfloor$ episodes (among a total of M episodes) in inner iteration/step $(l_{N,M}, k_{N,M})$.

Then the following regret bound can be established.

Corollary 7 (Regret for mini-batch REINFORCE)**.** *Under Assumption 1, suppose that for each $l \geq 0$, we choose $\alpha^{l,k} = C_{l,\alpha}\frac{1}{\sqrt{k+3}\log_2(k+3)}$, with $C_{l,\alpha} \in [1/(2\beta_{\bar{\lambda}}), 1/(2\beta_{\lambda^l})]$ and $\bar{\lambda} = \frac{1-\gamma}{2}$, and choose $T_l = 2^l$, $\epsilon^l = T_l^{-1/6} = 2^{-l/6}$, $\lambda^l = \frac{\epsilon^l(1-\gamma)}{2}$ and $\epsilon_{\rm pp} = 1/(2A)$. In addition, suppose that the assumptions in Lemma 12 hold (note that Assumption 1 and $\lambda^l \leq \bar{\lambda}$ already automatically hold by the other assumptions). Then we have for any $\delta \in (0,1)$, with probability at least $1-\delta$, jointly for all episodes N, we have (for the mini-batch Algorithm 4)*

$$\mathbf{regret}(N;M) = O\Big(\Big(\tfrac{S^2A^2}{(1-\gamma)^7} + \Big\|\tfrac{d_\rho^{\pi^\star}}{\rho}\Big\|_\infty\Big) \times (M^{\frac{1}{6}} + M^{-\frac{5}{6}})(N+M)^{\frac{5}{6}}(\log(N/\delta))^{\frac{5}{2}} + \tfrac{M(\log N)^2}{1-\gamma}\Big).$$

In addition, we also have

$$\lim_{N\to\infty} \mathbf{regret}(N;M)/(N+1) = 0 \quad \textit{almost surely}$$

with an asymptotic rate of

$$O\Big(\Big(\tfrac{S^2A^2}{(1-\gamma)^7} + \Big\|\tfrac{d_\rho^{\pi^\star}}{\rho}\Big\|_\infty\Big) \times (M^{\frac{1}{6}} + M^{-\frac{5}{6}})N^{-\frac{1}{6}}\left(1+\tfrac{M}{N}\right)^{\frac{5}{6}}(\log N)^{\frac{5}{2}} + \tfrac{M(\log N)^2}{(1-\gamma)N}\Big).$$

Again, we note that the almost sure convergence above is directly implied by the high probability bound via Lemma 4. The proof and a more formal statement of this corollary (*cf.* Corollary 13) can be found in Appendix D in the longer version of this paper (Zhang et al. 2020). In particular, when $M = 1$, the bound above reduces to (13). In addition, we can see that there might be a trade-off between the terms $M^{1/6}$ and $M^{-5/6}$. The intuition behind this is a trade-off between lower variance with larger batch sizes and more frequent updates with smaller batch sizes.

6 Conclusion and Open Problems

In this work, we establish the global convergence rates of practical policy gradient algorithms with a fixed size mini-batch of trajectories combined with REINFORCE gradient estimation.

Although in §4 and §5, we only instantiate the bounds for the REINFORCE gradient estimators, we note that our general results (in particular, Theorem 10 in Appendix B.3 in the longer version of this paper (Zhang et al. 2020)) can be easily applied to other gradient estimators (*e.g.*, actor-critic and state-action visitation measure based estimators) as well, as long as one can verify the existence of the constants in Assumption 2 in a similar way to Lemma 2. In addition, one can also easily derive sample complexity results as by-products of our analysis. In fact, our proof of Theorem 5 immediately implies a $\tilde{O}(1/\epsilon^4)$ sample complexity bound (for Algorithm 1 with REINFORCE gradient estimators and a constant regularization parameter) for any pre-specified tolerance $\epsilon > 0$, where we use $\tilde{O}$ to indicate the big-O notation with logarithmic terms suppressed. We have focused only on regret in this paper mainly for clarity purposes.

It is also relatively straightforward to extend our results to finite horizon non-stationary settings, in which the soft-max policy parametrization will have a dimension of SAH and different policy gradient estimators can be adopted (without trajectory truncation), with H being the horizon of each episode. In this case, it's also easy to rewrite the regret bound as a function of the total number of time steps $T \leq HN$, where N is the total number of episodes. Other straightforward extensions include refined convergence to stationary points (in both almost sure and high probability senses and with no requirement on large batch sizes), and inexact convergence results when $\delta^{l,k}$ (*cf.* Assumption 2) is not square summable (*e.g.*, when $H^{l,k}$ is fixed or not growing sufficiently fast).

There are also several open problems that may be resolved by combining the techniques introduced in this paper with existing results in the literature. Firstly, it would be desirable to remove the "exploration" assumption that the initial distribution ρ is component-wise positive. This may be achieved by combining our results with the policy cover technique in (Agarwal et al. 2020) or the optimistic bonus tricks in (Cai et al. 2019; Efroni et al. 2020). Secondly, the bounds in our paper are likely far from optimal (*i.e.*, sharp). Hence it would be desirable to either refine our analysis or apply our techniques to accelerated policy gradient methods (*e.g.*, IS-MBPG (Huang et al. 2020)) to obtain better global convergence rates and/or last-iterate convergence. Thirdly, it would be very interesting to see if global convergence results still hold for REINFORCE when the relative entropy regularization term used in this paper is replaced with the practically adopted entropy regularization term in the literature. The answer is affirmative when exact gradient estimations are available (Mei et al. 2020; Cen et al. 2020), but it remains unknown how these results might be generalized to the stochastic settings in our paper. We conjecture that entropy regularization leads to better global convergence rates and can help us remove the necessity of the `PostProcess` steps in Algorithm 2 as they are uniformly bounded. Finally, one may also consider relaxing the uniform bound assumption on the rewards r to instead being sub-Gaussian, introducing function approximation, and extending our results to natural policy gradient and actor-critic methods as well as more modern policy gradient methods like DPG, PPO and TRPO.

Acknowledgments

We would like to thank Anran Hu for pointing out a mistake in the proof of an early draft of this paper. We thank Guillermo Angeris, Shane Barratt, Haotian Gu, Xin Guo, Yusuke Kikuchi, Bennet Meyers, Xinyue Shen, Mahan Tajrobehkar, Jonathan Tuck, Jiaming Wang and Xiaoli Wei for their feedback on some preliminary results in this

paper. We thank Junyu Zhang for several detailed and fruitful discussions of the draft. We also thank the anonymous (meta-)reviewers for the great comments and suggestions. Jongho Kim is supported by Samsung Scholarship.

References

Agarwal, A.; Henaff, M.; Kakade, S.; and Sun, W. 2020. PC-PG: Policy Cover Directed Exploration for Provable Policy Gradient Learning. *arXiv preprint arXiv:2007.08459* .

Agarwal, A.; Jiang, N.; and Kakade, S. 2019. Reinforcement Learning: Theory and Algorithms. Technical report, Department of Computer Science, University of Washington.

Agarwal, A.; Kakade, S.; Lee, J.; and Mahajan, G. 2019. Optimality and approximation with policy gradient methods in Markov decision processes. *arXiv preprint arXiv:1908.00261* .

Basei, M.; Guo, X.; and Hu, A. 2020. Linear Quadratic Reinforcement Learning: Sublinear Regret in the Episodic Continuous-Time Framework. *arXiv preprint arXiv:2006.15316* .

Bertsekas, D. 2017. *Dynamic programming and optimal control*, volume II. Athena scientific Belmont, MA, 4th edition.

Besson, L.; and Kaufmann, E. 2018. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971* .

Bhandari, J.; and Russo, D. 2019. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786* .

Bhandari, J.; and Russo, D. 2020. A Note on the Linear Convergence of Policy Gradient Methods. *arXiv preprint arXiv:2007.11120* .

Bottou, L.; Curtis, F.; and Nocedal, J. 2018. Optimization methods for large-scale machine learning. *SIAM Review* 60(2): 223–311.

Cai, Q.; Yang, Z.; Jin, C.; and Wang, Z. 2019. Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830* .

Cen, S.; Cheng, C.; Chen, Y.; Wei, Y.; and Chi, Y. 2020. Fast Global Convergence of Natural Policy Gradient Methods with Entropy Regularization. *arXiv preprint arXiv:2007.06558* .

Efroni, Y.; Shani, L.; Rosenberg, A.; and Mannor, S. 2020. Optimistic Policy Optimization with Bandit Feedback. *arXiv preprint arXiv:2002.08243* .

Even-Dar, E.; and Mansour, Y. 2003. Learning rates for Q-learning. *Journal of machine learning Research* 5(Dec): 1–25.

Fazel, M.; Ge, R.; Kakade, S.; and Mesbahi, M. 2018. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039* .

Fu, Z.; Yang, Z.; and Wang, Z. 2020. Single-Timescale Actor-Critic Provably Finds Globally Optimal Policy. *arXiv preprint arXiv:2008.00483* .

Gu, S.; Levine, S.; Sutskever, I.; and Mnih, A. 2015. MuProp: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176* .

Guu, K.; Pasupat, P.; Liu, E.; and Liang, P. 2017. From language to programs: Bridging reinforcement learning and maximum marginal likelihood. *arXiv preprint arXiv:1704.07926* .

Huang, F.; Gao, S.; Pei, J.; and Huang, H. 2020. Momentum-Based Policy Gradient Methods. *arXiv preprint arXiv:2007.06680* .

Jaksch, T.; Ortner, R.; and Auer, P. 2010. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research* 11(Apr): 1563–1600.

Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. 2018. Is Q-learning provably efficient? In *Advances in Neural Information Processing Systems*, 4863–4873.

Johnson, J.; Hariharan, B.; Van Der Maaten, L.; Hoffman, J.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2989–2998.

Kakade, S.; et al. 2003. *On the sample complexity of reinforcement learning*. Ph.D. thesis, University of London London, England.

Konda, V.; and Tsitsiklis, J. 2003. On actor-critic algorithms. *SIAM journal on Control and Optimization* 42(4): 1143–1166.

Kool, W.; van Hoof, H.; and Welling, M. 2018. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475* .

Kool, W.; van Hoof, H.; and Welling, M. 2020. Estimating gradients for discrete random variables by sampling without replacement. *arXiv preprint arXiv:2002.06043* .

Malik, D.; Pananjady, A.; Bhatia, K.; Khamaru, K.; Bartlett, P.; and Wainwright, M. 2018. Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. *arXiv preprint arXiv:1812.08305* .

Marbach, P.; and Tsitsiklis, J. 2001. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control* 46(2): 191–209.

Mei, J.; Xiao, C.; Szepesvari, C.; and Schuurmans, D. 2020. On the Global Convergence Rates of Softmax Policy Gradient Methods. *arXiv preprint arXiv:2005.06392* .

Mnih, A.; and Gregor, K. 2014. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030* .

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

Peters, J.; Mulling, K.; and Altun, Y. 2010. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24(1).

Rennie, S.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7008–7024.

Schulman, J.; Chen, X.; and Abbeel, P. 2017. Equivalence between policy gradients and soft Q-learning. *arXiv preprint arXiv:1704.06440* .

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Shani, L.; Efroni, Y.; and Mannor, S. 2019. Adaptive trust region policy optimization: Global convergence and faster rates for regularized MDPs. *arXiv preprint arXiv:1909.02769* .

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic Policy Gradient Algorithms. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32(1), 387–395.

Sutton, R.; McAllester, D.; Singh, S.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1057–1063.

Wang, L.; Cai, Q.; Yang, Z.; and Wang, Z. 2019. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150* .

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

Yi, K.; Wu, J.; Gan, C.; Torralba, A.; Kohli, P.; and Tenenbaum, J. 2018. Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. In *Advances in neural information processing systems*, 1031–1042.

Zhang, J.; Kim, J.; O'Donoghue, B.; and Boyd, S. 2020. Sample efficient reinforcement learning with REINFORCE. Accessed March 23. [Online]. Available: https://stanford.edu/~boyd/papers/conv_reinforce.html.

Zhang, K.; Koppel, A.; Zhu, H.; and Başar, T. 2019. Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383* .

Zhao, T.; Hachiya, H.; Niu, G.; and Sugiyama, M. 2011. Analysis and improvement of policy gradient estimation. In *Advances in Neural Information Processing Systems*, 262–270.

Zoph, B.; and Le, Q. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* .