NEURAL POLICY GRADIENT METHODS: GLOBAL OPTIMALITY AND RATES OF CONVERGENCE

Lingxiao Wang[*†] **Qi Cai**[*‡] **Zhuoran Yang**[§] **Zhaoran Wang**[¶]

ABSTRACT

Policy gradient methods with actor-critic schemes demonstrate tremendous empirical successes, especially when the actors and critics are parameterized by neural networks. However, it remains less clear whether such "neural" policy gradient methods converge to globally optimal policies and whether they even converge at all. We answer both the questions affirmatively under the overparameterized two-layer neural-network parameterization. In detail, assuming independent sampling, we prove that neural natural policy gradient converges to a globally optimal policy at a sublinear rate. Also, we show that neural vanilla policy gradient converges sublinearly to a stationary point. Meanwhile, by relating the suboptimality of the stationary points to the representation power of neural actor and critic classes, we prove the global optimality of all stationary points under mild regularity conditions. Particularly, we show that a key to the global optimality and convergence is the "compatibility" between the actor and critic, which is ensured by sharing neural architectures and random initializations across the actor and critic. To the best of our knowledge, our analysis establishes the first global optimality and convergence guarantees for neural policy gradient methods. [1]

1 INTRODUCTION

In reinforcement learning (Sutton and Barto, 2018), an agent aims to maximize its expected total reward by taking a sequence of actions according to a policy in a stochastic environment, which is modeled as a Markov decision process (MDP) (Puterman, 2014). To obtain the optimal policy, policy gradient methods (Williams, 1992; Baxter and Bartlett, 2000; Sutton et al., 2000) directly maximize the expected total reward via gradient-based optimization. As policy gradient methods are easily implementable and readily integrable with advanced optimization techniques such as variance reduction (Johnson and Zhang, 2013; Papini et al., 2018) and distributed optimization (Mnih et al., 2016; Espeholt et al., 2018), they enjoy wide popularity among practitioners. In particular, when the policy (actor) and action-value function (critic) are parameterized by neural networks, policy gradient methods achieve significant empirical successes in challenging applications, such as playing Go (Silver et al., 2016; 2017), real-time strategy gaming (Vinyals et al., 2019), robot manipulation (Peters and Schaal, 2006; Duan et al., 2016), and natural language processing (Wang et al., 2018). See Li (2017) for a detailed survey.

In stark contrast to the tremendous empirical successes, policy gradient methods remain much less well understood in terms of theory, especially when they involve neural networks. More specifically, most existing work analyzes the REINFORCE algorithm (Williams, 1992; Sutton et al., 2000), which estimates the policy gradient via Monte Carlo sampling. Based on the recent progress in nonconvex optimization, Papini et al. (2018); Shen et al. (2019); Xu et al. (2019a); Karimi et al. (2019);

[*]equal contribution
[†]Northwestern University; lingxiaowang2022@u.northwestern.edu
[‡]Northwestern University; qicai2022@u.northwestern.edu
[§]Princeton University; zy6@princeton.edu
[¶]Northwestern University; zhaoranwang@gmail.com
[1]See https://arxiv.org/abs/1909.01150 for the full version.

Zhang et al. (2019) establish the rate of convergence of REINFORCE to a first- or second-order stationary point. However, the global optimality of the attained stationary point remains unclear. A more commonly used class of policy gradient methods is equipped with the actor-critic scheme (Konda and Tsitsiklis, 2000), which alternatingly estimates the action-value function in the policy gradient via a policy evaluation step (critic update), and performs a policy improvement step using the estimated policy gradient (actor update). The global optimality and rate of convergence of such a class are even more challenging to analyze than that of REINFORCE. In particular, the policy evaluation step itself may converge to an undesirable stationary point or even diverge (Tsitsiklis and Van Roy, 1997), especially when it involves both nonlinear action-value function approximator, such as neural network, and temporal-difference update (Sutton, 1988). As a result, the estimated policy gradient may be biased, which possibly leads to divergence. Even if the algorithm converges to a stationary point, due to the nonconvexity of the expected total reward with respect to the policy as well as its parameter, the global optimality of such a stationary point remains unclear. The only exception is the linear-quadratic regulator (LQR) setting (Fazel et al., 2018; Malik et al., 2018; Tu and Recht, 2018; Yang et al., 2019a; Bu et al., 2019), which is, however, more restrictive than the general MDP setting that possibly involves neural networks.

To bridge the gap between practice and theory, we analyze neural policy gradient methods equipped with actor-critic schemes, where the actors and critics are represented by overparameterized two-layer neural networks. In detail, we study two settings, where the policy improvement steps are based on vanilla policy gradient and natural policy gradient, respectively. In both settings, the policy evaluation steps are based on the TD(0) algorithm (Sutton, 1988) with independent sampling. In the first setting, we prove that neural vanilla policy gradient converges to a stationary point of the expected total reward at a $1/\sqrt{T}$-rate in the expected squared norm of the policy gradient, where T is the number of policy improvement steps. Meanwhile, through a geometric characterization that relates the suboptimality of the stationary points to the representation power of the neural networks parameterizing the actor and critic, we establish the global optimality of all stationary points under mild regularity conditions. In the second setting, through the lens of Kullback-Leibler (KL) divergence regularization, we prove that neural natural policy gradient converges to a globally optimal policy at a $1/\sqrt{T}$-rate in the expected total reward. In particular, a key to such global optimality and convergence guarantees is a notion of compatibility between the actor and critic, which connects the accuracy of policy evaluation steps with the efficacy of policy improvement steps. We show that such a notion of compatibility is ensured by using shared neural architectures and random initializations for both the actor and critic, which is often used as a practical heuristic (Mnih et al., 2016). To our best knowledge, our analysis gives the first global optimality and convergence guarantees for neural policy gradient methods, which corroborate their significant empirical successes.

Related Work. In contrast to the huge body of empirical literature on policy gradient methods, theoretical results on their convergence remain relatively scarce. In particular, Sutton et al. (2000) and Kakade (2002) analyze vanilla policy gradient (REINFORCE) and natural policy gradient with compatible action-value function approximators, respectively, which are further extended by Konda and Tsitsiklis (2000); Peters and Schaal (2008); Castro and Meir (2010) to incorporate actor-critic schemes. Most of this line of work only establishes the asymptotic convergence based on stochastic approximation techniques (Kushner and Yin, 2003; Borkar, 2009) and requires the actor and critic to be parameterized by linear functions. Another line of work (Papini et al., 2018; Xu et al., 2019a;b; Shen et al., 2019; Karimi et al., 2019; Zhang et al., 2019) builds on the recent progress in nonconvex optimization to establish the nonasymptotic rates of convergence of REINFORCE (Williams, 1992; Baxter and Bartlett, 2000; Sutton et al., 2000) and its variants, but only to first- or second-order stationary points, which, however, lacks global optimality guarantees. Moreover, when actor-critic schemes are involved, due to the error of policy evaluation steps and its impact on policy improvement steps, the nonasymptotic rates of convergence of policy gradient methods, even to first- or second-order stationary points, remain rather open.

Compared with the convergence of policy gradient methods, their global optimality is even less explored in terms of theory. Fazel et al. (2018); Malik et al. (2018); Tu and Recht (2018); Yang et al. (2019a); Bu et al. (2019) prove that policy gradient methods converge to globally optimal policies in the LQR setting, which is more restrictive. In very recent work, Bhandari and Russo (2019) establish the global optimality of vanilla policy gradient (REINFORCE) in the general MDP setting. However, they require the policy class to be convex, which restricts its applicability to the tabular

and LQR settings. In independent work, Agarwal et al. (2019) prove that vanilla policy gradient and natural policy gradient converge to globally optimal policies at $1/\sqrt{T}$-rates in the tabular and linear settings. In the tabular setting, their rate of convergence of vanilla policy gradient depends on the size of the state space. In contrast, we focus on the nonlinear setting with the actor-critic scheme, where the actor and critic are parameterized by neural networks. It is worth mentioning that when such neural networks have linear activation functions, our analysis also covers the linear setting, which is, however, not our focus. In addition, Liu et al. (2019) analyze the proximal policy optimization (PPO) and trust region policy optimization (TRPO) algorithms (Schulman et al., 2015; 2017), where the actors and critics are parameterized by neural networks, and establish their $1/\sqrt{T}$-rates of convergence to globally optimal policies. However, they require solving a subproblem of policy improvement in the functional space using multiple stochastic gradient steps in the parameter space, whereas vanilla policy gradient and natural policy gradient only require a single stochastic (natural) gradient step in the parameter space, which makes the analysis even more challenging.

There is also an emerging body of literature that analyzes the training and generalization error of deep supervised learning with overparameterized neural networks (Daniely, 2017; Jacot et al., 2018; Wu et al., 2018; Allen-Zhu et al., 2018a;b; Du et al., 2018a;b; Zou et al., 2018; Chizat and Bach, 2018; Jacot et al., 2018; Li and Liang, 2018; Cao and Gu, 2019a;b; Arora et al., 2019; Lee et al., 2019), especially when they are trained using stochastic gradient. See Fan et al. (2019) for a detailed survey. In comparison, our focus is on deep reinforcement learning with policy gradient methods. In particular, the policy evaluation steps are based on the TD(0) algorithm, which uses stochastic semigradient (Sutton, 1988) rather than stochastic gradient. Moreover, the interplay between the actor and critic makes our analysis even more challenging than that of deep supervised learning.

Notation. For distribution μ on Ω and $p > 0$, we define $\|f(\cdot)\|_{\mu,p} = (\int_\Omega |f|^p \mathrm{d}\mu)^{1/p}$ as the $L_p(\mu)$ norm of f. We define $\|f(\cdot)\|_{\mu,\infty} = \inf\{C \geq 0 : |f(x)| \leq C \text{ for } \mu\text{-almost every } x\}$ as the $L_\infty(\mu)$-norm of f. We write $\|f\|_{\mu,p}$ for notational simplicity when the variable of f is clear from the context. We further denote by $\|\cdot\|_\mu$ the $L_2(\mu)$-norm for notational simplicity. For a vector $\phi \in \mathbb{R}^n$ and $p > 0$, we denote by $\|\phi\|_p$ the ℓ_p-norm of ϕ. We denote by $x = ([x]_1^\top, \ldots, [x]_m^\top)^\top$ a vector in $\mathbb{R}^{md}$, where $[x]_i \in \mathbb{R}^d$ is the i-th block of x for $i \in [m]$.

2 Background

In this section, we introduce the background of reinforcement learning and policy gradient methods.

Reinforcement Learning. A discounted Markov decision process (MDP) is defined by tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \zeta, r, \gamma)$. Here $\mathcal{S}$ and $\mathcal{A}$ are the sets of all possible states and actions, respectively. Meanwhile, $\mathcal{P}$ is the Markov transition kernel and r is the reward function, which is possibly stochastic. Specifically, when taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$, the agent receives reward $r(s, a)$ and the environment transits into a new state according to transition probability $\mathcal{P}(\cdot \,|\, s, a)$. Meanwhile, ζ is the distribution of initial state $S_0 \in \mathcal{S}$ and $\gamma \in (0, 1)$ is the discount factor. In addition, policy $\pi(a \,|\, s)$ gives the probability of taking action a at state s. We denote the state- and action-value functions associated with π by $V^\pi \colon \mathcal{S} \to \mathbb{R}$ and $Q^\pi \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which are defined respectively as

$$V^\pi(s) = (1-\gamma) \cdot \mathbb{E}\bigg[\sum_{t=0}^{\infty} \gamma^t \cdot r(S_t, A_t) \,\bigg|\, S_0 = s\bigg], \quad \forall s \in \mathcal{S}, \tag{2.1}$$

$$Q^\pi(s, a) = (1-\gamma) \cdot \mathbb{E}\bigg[\sum_{t=0}^{\infty} \gamma^t \cdot r(S_t, A_t) \,\bigg|\, S_0 = s, A_0 = a\bigg], \quad \forall (s, a) \in \mathcal{S} \times \mathcal{A}, \tag{2.2}$$

where $S_0 \sim \zeta(\cdot)$, $A_t \sim \pi(\cdot \,|\, S_t)$, and $S_{t+1} \sim \mathcal{P}(\cdot \,|\, S_t, A_t)$ for all $t \geq 0$. Also, we define the advantage function of policy π as the difference between Q^π and V^π, i.e., $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$. By the definitions in (2.1) and (2.2), V^π and Q^π are related via

$$V^\pi(s) = \mathbb{E}_\pi\big[Q^\pi(s, a)\big] = \langle Q^\pi(s, \cdot), \pi(\cdot \,|\, s)\rangle,$$

where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathbb{R}^{|\mathcal{A}|}$. Here we write $\mathbb{E}_{a \sim \pi(\cdot \,|\, s)}[Q^\pi(s, a)]$ as $\mathbb{E}_\pi[Q^\pi(s, a)]$ for notational simplicity. Note that policy π together with transition probability $\mathcal{P}$ induces a Markov chain over state space $\mathcal{S}$. We denote by ϱ_π the stationary state distribution of the Markov chain

induced by π. We further define $\varsigma_\pi(s,a) = \pi(a\,|\,s)\cdot\varrho_\pi(s)$ as the stationary state-action distribution over $\mathcal{S}\times\mathcal{A}$. Meanwhile, policy π induces a state visitation measure over $\mathcal{S}$ and a state-action visitation measure over $\mathcal{S}\times\mathcal{A}$, which are denoted by ν_π and σ_π, respectively. Specifically, for all $(s,a)\in\mathcal{S}\times\mathcal{A}$, we define

$$\nu_\pi(s) = (1-\gamma)\cdot\sum_{t=0}^{\infty}\gamma^t\cdot\mathbb{P}(S_t = s), \qquad \sigma_\pi(s,a) = (1-\gamma)\cdot\sum_{t=0}^{\infty}\gamma^t\cdot\mathbb{P}(S_t = s, A_t = a), \tag{2.3}$$

where $S_0\sim\zeta(\cdot)$, $A_t\sim\pi(\cdot\,|\,S_t)$, and $S_{t+1}\sim\mathcal{P}(\cdot\,|\,S_t,A_t)$ for all $t\geq 0$. By definition, we have $\sigma_\pi(s,a) = \pi(a\,|\,s)\cdot\nu_\pi(s)$. We define the expected total reward function $J(\pi)$ by

$$J(\pi) = (1-\gamma)\cdot\mathbb{E}\Big[\sum_{t=0}^{\infty}\gamma^t\cdot r(S_t,A_t)\Big] = \mathbb{E}_\zeta\big[V^\pi(s)\big] = \mathbb{E}_{\sigma_\pi}\big[r(s,a)\big], \quad \forall\pi, \tag{2.4}$$

where we write $\mathbb{E}_{\sigma_\pi}[r(s,a)] = \mathbb{E}_{(s,a)\sim\sigma_\pi(\cdot,\cdot)}[r(s,a)]$ for notational simplicity. The goal of reinforcement learning is to find the optimal policy that maximizes $J(\pi)$, which is denoted by π^*. When state space $\mathcal{S}$ is large, a popular approach is to find the maximizer of $J(\pi)$ over a class of parameterized policies $\{\pi_\theta\colon\theta\in\mathcal{B}\}$, where $\theta\in\mathcal{B}$ is the parameter and $\mathcal{B}$ is the parameter space. In this case, we obtain the optimization problem $\max_{\theta\in\mathcal{B}} J(\pi_\theta)$.

Policy Gradient Methods. Policy gradient methods maximize $J(\pi_\theta)$ using $\nabla_\theta J(\pi_\theta)$. These methods are based on the policy gradient theorem (Sutton and Barto, 2018), which states that

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\sigma_{\pi_\theta}}\big[Q^{\pi_\theta}(s,a)\cdot\nabla_\theta\log\pi_\theta(a\,|\,s)\big], \tag{2.5}$$

where σ_{π_θ} is the state-action visitation measure defined in (2.3). Based on (2.5), (vanilla) policy gradient maximizes the expected total reward via gradient ascent. Specifically, we generate a sequence of policy parameters $\{\theta_i\}_{i\in[T]}$ via

$$\theta_{i+1}\leftarrow\theta_i+\eta\cdot\nabla_\theta J(\pi_{\theta_i}), \tag{2.6}$$

where $\eta>0$ is the learning rate. Meanwhile, natural policy gradient (Kakade, 2002) utilizes natural gradient ascent (Amari, 1998), which is invariant to the parameterization of policies. Specifically, let $F(\theta)$ be the Fisher information matrix corresponding to policy π_θ, which is given by

$$F(\theta) = \mathbb{E}_{\sigma_{\pi_\theta}}\Big[\nabla_\theta\log\pi_\theta(a\,|\,s)\big[\nabla_\theta\log\pi_\theta(a\,|\,s)\big]^\top\Big]. \tag{2.7}$$

At each iteration, natural policy gradient performs

$$\theta_{i+1}\leftarrow\theta_i+\eta\cdot F^{-1}(\theta_i)\cdot\nabla_\theta J(\pi_{\theta_i}), \tag{2.8}$$

where $F^{-1}(\theta_i)$ is the inverse of $F(\theta_i)$ and η is the learning rate. In practice, both Q^{π_θ} in (2.5) and $F(\theta)$ in (2.7) remain to be estimated, which yields approximations of the policy improvement steps in (2.6) and (2.8).

3 Neural Policy Gradient Methods

In this section, we represent π_θ by a two-layer neural network and study neural policy gradient methods, which estimate the policy gradient and natural policy gradient using the actor-critic scheme (Konda and Tsitsiklis, 2000).

3.1 Overparameterized Neural Policy

We now introduce the parameterization of policies. For notational simplicity, we assume that $\mathcal{S}\times\mathcal{A}\subseteq\mathbb{R}^d$ with $d\geq 2$. Without loss of generality, we further assume that $\|(s,a)\|_2 = 1$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$. A two-layer neural network $f((s,a);W,b)$ with input (s,a) and width m takes the form of

$$f\big((s,a);W,b\big) = \frac{1}{\sqrt{m}}\sum_{r=1}^{m} b_r\cdot\mathrm{ReLU}\big((s,a)^\top[W]_r\big), \quad \forall(s,a)\in\mathcal{S}\times\mathcal{A}. \tag{3.1}$$

Here $\mathrm{ReLU}\colon\mathbb{R}\to\mathbb{R}$ is the rectified linear unit (ReLU) activation function, which is defined as $\mathrm{ReLU}(u) = \mathbb{1}\{u>0\}\cdot u$. Also, $\{b_r\}_{r\in[m]}$ and $W = ([W]_1^\top,\ldots,[W]_m^\top)^\top\in\mathbb{R}^{md}$ in (3.1) are the parameters. When training the two-layer neural network, we initialize the parameters via

$[W_{\text{init}}]_r \sim N(0, I_d/d)$ and $b_r \sim \text{Unif}(\{-1, 1\})$ for all $r \in [m]$. Note that the ReLU activation function satisfies $\text{ReLU}(c \cdot u) = c \cdot \text{ReLU}(u)$ for all $c > 0$ and $u \in \mathbb{R}$. Hence, without loss of generality, we keep b_r fixed at the initial parameter throughout training and only update W in the sequel. See, e.g., Allen-Zhu et al. (2018b) for a detailed argument. For notational simplicity, we write $f((s,a); W, b)$ as $f((s,a); W)$ hereafter.

Using the two-layer neural network in (3.1), we define

$$\pi_\theta(a \,|\, s) = \frac{\exp\big[\tau \cdot f\big((s,a);\theta\big)\big]}{\sum_{a' \in \mathcal{A}} \exp\big[\tau \cdot f\big((s,a');\theta\big)\big]}, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{3.2}$$

where $f((\cdot,\cdot);\theta)$ is defined in (3.1) with $\theta \in \mathbb{R}^{md}$ playing the role of W. Note that π_θ defined in (3.2) takes the form of an energy-based policy (Haarnoja et al., 2017). With a slight abuse of terminology, we call τ the temperature parameter and $f((\cdot,\cdot);\theta)$ the energy function in the sequel.

In the sequel, we investigate policy gradient methods for the class of neural policies defined in (3.2). We define the feature mapping $\phi_\theta = ([\phi_\theta]_1^\top, \ldots, [\phi_\theta]_m^\top)^\top : \mathbb{R}^d \to \mathbb{R}^{md}$ of a two-layer neural network $f((\cdot,\cdot);\theta)$ as

$$[\phi_\theta]_r(s,a) = \frac{b_r}{\sqrt{m}} \cdot \mathbb{1}\big\{(s,a)^\top [\theta]_r > 0\big\} \cdot (s,a), \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A},\ \forall r \in [m]. \tag{3.3}$$

By (3.1), it holds that $f((\cdot,\cdot);\theta) = \phi_\theta(\cdot,\cdot)^\top \theta$. Meanwhile, $f((\cdot,\cdot);\theta)$ is almost everywhere differentiable with respect to θ, and it holds that $\nabla_\theta f((\cdot,\cdot);\theta) = \phi_\theta(\cdot,\cdot)$. In the following proposition, we calculate the closed forms of the policy gradient $\nabla_\theta J(\pi_\theta)$ and the Fisher information matrix $F(\theta)$ for π_θ defined in (3.2).

Proposition 3.1 (Policy Gradient and Fisher Information Matrix)**.** For π_θ defined in (3.2), we have

$$\nabla_\theta J(\pi_\theta) = \tau \cdot \mathbb{E}_{\sigma_{\pi_\theta}} \Big[Q^{\pi_\theta}(s,a) \cdot \Big(\phi_\theta(s,a) - \mathbb{E}_{\pi_\theta}\big[\phi_\theta(s,a')\big] \Big) \Big], \tag{3.4}$$

$$F(\theta) = \tau^2 \cdot \mathbb{E}_{\sigma_{\pi_\theta}} \Big[\Big(\phi_\theta(s,a) - \mathbb{E}_{\pi_\theta}\big[\phi_\theta(s,a')\big] \Big) \Big(\phi_\theta(s,a) - \mathbb{E}_{\pi_\theta}\big[\phi_\theta(s,a')\big] \Big)^\top \Big], \tag{3.5}$$

where $\phi_\theta(\cdot,\cdot)$ is the feature mapping defined in (3.3), τ is the temperature parameter, and σ_{π_θ} is the state-action visitation measure defined in (2.3). Here we write $\mathbb{E}_{\pi_\theta}[\phi_\theta(s,a')] = \mathbb{E}_{a' \sim \pi_\theta(\cdot \,|\, s)}[\phi_\theta(s,a')]$ for notational simplicity.

Proof. See §H.1 for a detailed proof. □

Since the action-value function Q^{π_θ} in (3.4) is unknown, to obtain the policy gradient, we use another two-layer neural network to track the action-value function of policy π_θ. Specifically, we use a two-layer neural network $Q_\omega(\cdot,\cdot) = f((\cdot,\cdot);\omega)$ defined in (3.1) to represent the action-value function Q^{π_θ}, where ω plays the same role as W in (3.1). Such an approach is known as the actor-critic scheme (Konda and Tsitsiklis, 2000). We call π_θ and Q_ω the actor and critic, respectively. We highlight that in the overparameterized regime where the width of two-layer neural networks m is large, a shared architecture and random initialization between the actor Q_ω and the energy function of critic π_θ ensures approximate compatible function approximations. See §B for details.

3.2 Neural Policy Gradient Methods

Now we present neural policy gradient and neural natural policy gradient. Following the actor-critic scheme, they generate a sequence of policies $\{\pi_{\theta_i}\}_{i \in [T+1]}$ and action-value functions $\{Q_{\omega_i}\}_{i \in [T]}$.

3.2.1 Actor Update

As introduced in §2, we aim to solve the optimization problem $\max_{\theta \in \mathcal{B}} J(\pi_\theta)$ iteratively via gradient-based methods, where $\mathcal{B}$ is the parameter space. We set $\mathcal{B} = \{\alpha \in \mathbb{R}^{md} : \|\alpha - W_{\text{init}}\|_2 \le R\}$, where $R > 1$ and W_{init} is the initial parameter defined in §3.1. For all $i \in [T]$, let θ_i be the policy parameter at the i-th iteration. For notational simplicity, in the sequel, we denote by σ_i and ς_i the state-action visitation measure $\sigma_{\pi_{\theta_i}}$ and the stationary state-action distribution $\varsigma_{\pi_{\theta_i}}$, respectively, which are defined in §2. Similarly, we write $\nu_i = \nu_{\pi_{\theta_i}}$ and $\varrho_i = \varrho_{\pi_{\theta_i}}$. To update θ_i, we set

$$\theta_{i+1} \leftarrow \Pi_{\mathcal{B}}\big(\theta_i + \eta \cdot G(\theta_i) \cdot \widehat{\nabla}_\theta J(\pi_{\theta_i})\big), \tag{3.6}$$

where we define $\Pi_{\mathcal{B}}\colon \mathbb{R}^{md} \to \mathcal{B}$ as the projection operator onto the parameter space $\mathcal{B} \subseteq \mathbb{R}^{md}$. Here $G(\theta_i) \in \mathbb{R}^{md\times md}$ is a matrix specific to each algorithm. Specifically, we have $G(\theta_i) = I_{md}$ for policy gradient and $G(\theta_i) = (F(\theta_i))^{-1}$ for natural policy gradient, where $F(\theta_i)$ is the Fisher information matrix in (3.5). Meanwhile, η is the learning rate and $\widehat{\nabla}_\theta J(\pi_{\theta_i})$ is an estimator of $\nabla_\theta J(\pi_{\theta_i})$, which takes the form of

$$\widehat{\nabla}_\theta J(\pi_{\theta_i}) = \frac{1}{B} \cdot \sum_{\ell=1}^{B} Q_{\omega_i}(s_\ell, a_\ell) \cdot \nabla_\theta \log \pi_{\theta_i}(a_\ell \,|\, s_\ell). \tag{3.7}$$

Here τ_i is the temperature parameter of π_{θ_i}, $\{(s_\ell, a_\ell)\}_{\ell\in[B]}$ is sampled from the state-action visitation measure σ_i corresponding to the current policy π_{θ_i}, and $B > 0$ is the batch size. Also, Q_{ω_i} is the critic obtained by Algorithm 2. Here we omit the dependency of $\widehat{\nabla}_\theta J(\pi_{\theta_i})$ on ω_i for notational simplicity. See §C for the sampling from visitation measures.

Inverting Fisher Information Matrix. Recall that $G(\theta_i)$ is the inverse of the Fisher information matrix used in natural policy gradient. In the overparameterized regime, inverting an estimator $\widehat{F}(\theta_i)$ of $F(\theta_i)$ can be infeasible as $\widehat{F}(\theta_i)$ is a high-dimensional matrix, which is possibly not invertible. To resolve this issue, we estimate the natural policy gradient $G(\theta_i) \cdot \nabla_\theta J(\pi_{\theta_i})$ by solving

$$\min_{\alpha\in\mathcal{B}} \|\widehat{F}(\theta_i) \cdot \alpha - \tau_i \cdot \widehat{\nabla}_\theta J(\pi_{\theta_i})\|_2, \tag{3.8}$$

where $\widehat{\nabla}_\theta J(\pi_{\theta_i})$ is defined in (3.7), τ_i is the temperature parameter in π_{θ_i}, and $\mathcal{B}$ is the parameter space. Meanwhile, $\widehat{F}(\theta_i)$ is an unbiased estimator of $F(\theta_i)$ based on $\{(s_\ell, a_\ell)\}_{\ell\in[B]}$ sampled from σ_i, which is defined as

$$\widehat{F}(\theta_i) = \frac{\tau_i^2}{B} \cdot \sum_{\ell=1}^{B} \Big(\phi_{\theta_i}(s_\ell, a_\ell) - \mathbb{E}_{\pi_{\theta_i}}\big[\phi_{\theta_i}(s_\ell, a'_\ell)\big]\Big)\Big(\phi_{\theta_i}(s_\ell, a_\ell) - \mathbb{E}_{\pi_{\theta_i}}\big[\phi_{\theta_i}(s_\ell, a'_\ell)\big]\Big)^\top, \tag{3.9}$$

where $a'_\ell \sim \pi_{\theta_i}(\cdot \,|\, s_\ell)$ and ϕ_{θ_i} is defined in (3.3) with $\theta = \theta_i$. The actor update of neural natural policy gradient takes the form of

$$\tau_{i+1} \leftarrow \tau_i + \eta, \qquad \tau_{i+1} \cdot \theta_{i+1} \leftarrow \tau_i \cdot \theta_i + \eta \cdot \operatorname*{argmin}_{\alpha\in\mathcal{B}} \|\widehat{F}(\theta_i) \cdot \alpha - \tau_i \cdot \widehat{\nabla}_\theta J(\pi_{\theta_i})\|_2, \tag{3.10}$$

where we use an arbitrary minimizer of (3.8) if it is not unique. Note that we also update the temperature parameter by $\tau_{i+1} \leftarrow \tau_i + \eta$, which ensures $\theta_{i+1} \in \mathcal{B}$. It is worth mentioning that up to minor modifications, our analysis allows for approximately solving (3.8), which is the common practice of approximate second-order optimization (Martens and Grosse, 2015; Wu et al., 2017).

To summarize, at the i-th iteration, neural policy gradient obtains θ_{i+1} via projected gradient ascent using $\widehat{\nabla}_\theta J(\pi_{\theta_i})$ defined in (3.7). Meanwhile, neural natural policy gradient solves (3.8) and obtains θ_{i+1} according to (3.10).

3.2.2 Critic Update

To obtain $\widehat{\nabla}_\theta J(\pi_\theta)$, it remains to obtain the critic Q_{ω_i} in (3.7). For any policy π, the action-value function Q^π is the unique solution to the Bellman equation $Q = \mathcal{T}^\pi Q$ (Sutton and Barto, 2018). Here $\mathcal{T}^\pi$ is the Bellman operator that takes the form of

$$\mathcal{T}^\pi Q(s,a) = \mathbb{E}\big[(1-\gamma)\cdot r(s,a) + \gamma \cdot Q(s', a')\big], \quad \forall (s,a) \in \mathcal{S}\times\mathcal{A},$$

where $s' \sim \mathcal{P}(\cdot \,|\, s, a)$ and $a' \sim \pi(\cdot \,|\, s')$. Correspondingly, we aim to solve the following optimization problem

$$\omega_i \leftarrow \operatorname*{argmin}_{\omega\in\mathcal{B}} \mathbb{E}_{\varsigma_i}\Big[\big(Q_\omega(s,a) - \mathcal{T}^{\pi_{\theta_i}} Q_\omega(s,a)\big)^2\Big], \tag{3.11}$$

where ς_i and $\mathcal{T}^{\pi_{\theta_i}}$ are the stationary state-action distribution and the Bellman operator associated with π_{θ_i}, respectively, and $\mathcal{B}$ is the parameter space. We adopt neural temporal-difference learning (TD) studied in Cai et al. (2019), which solves the optimization problem in (3.11) via stochastic

semigradient descent (Sutton, 1988). Specifically, an iteration of neural TD takes the form of

$$\omega(t+1/2) \leftarrow \omega(t) - \eta_{\mathrm{TD}} \cdot \big(Q_{\omega(t)}(s,a) - (1-\gamma)\cdot r(s,a) - \gamma Q_{\omega(t)}(s',a')\big)\cdot \nabla_\omega Q_{\omega(t)}(s,a), \quad (3.12)$$

$$\omega(t+1) \leftarrow \operatorname*{argmin}_{\alpha\in\mathcal{B}} \|\alpha - \omega(t+1/2)\|_2, \quad (3.13)$$

where $(s,a)\sim\varsigma_i(\cdot)$, $s'\sim\mathcal{P}(\cdot\,|\,s,a)$, $a'\sim\pi(\cdot\,|\,s')$, and η_{TD} is the learning rate of neural TD. Here (3.12) is the stochastic semigradient step, and (3.13) projects the parameter obtained by (3.12) back to the parameter space $\mathcal{B}$. Meanwhile, the state-action pairs in (3.12) are sampled from the stationary state-action distribution ς_i, which is achieved by sampling from the Markov chain induced by π_{θ_i} until it mixes. See Algorithm 2 in §F for details. Finally, combining the actor updates and the critic update described in (3.6), (3.10), and (3.11), respectively, we obtain neural policy gradient and natural policy gradient, which are described in Algorithm 1.

Algorithm 1 Neural Policy Gradient Methods

Require: Number of iterations T, number of TD iterations T_{TD}, learning rate η, learning rate η_{TD} of neural TD, temperature parameters $\{\tau_i\}_{i\in[T+1]}$, batch size B.

1: **Initialization:** Initialize $b_r\sim\mathrm{Unif}(\{-1,1\})$ and $[W_{\mathrm{init}}]_r\sim N(0,I_d/d)$ for all $r\in[m]$. Set $\mathcal{B}\leftarrow\{\alpha\in\mathbb{R}^{md}:\|\alpha-W_{\mathrm{init}}\|_2\le R\}$ and $\theta_1\leftarrow W_{\mathrm{init}}$.

2: **for** $i\in[T]$ **do**

3: Update ω_i using Algorithm 2 with π_{θ_i} as the input, $\omega(0)\leftarrow W_{\mathrm{init}}$ and $\{b_r\}_{r\in[m]}$ as the initialization, T_{TD} as the number of iterations, and η_{TD} as the learning rate.

4: Sample $\{(s_\ell,a_\ell)\}_{\ell\in[B]}$ from the visitation measure σ_i, and estimate $\widehat{\nabla}_\theta J(\pi_\theta)$ and $\widehat{F}(\theta_i)$ using (3.7) and (3.9), respectively.

5: If using policy gradient, update θ_{i+1} by

$$\theta_{i+1}\leftarrow\Pi_{\mathcal{B}}\big(\theta_i+\eta\cdot\widehat{\nabla}_\theta J(\pi_{\theta_i})\big).$$

If using natural policy gradient, update θ_{i+1} and τ_{i+1} by

$$\tau_{i+1}\leftarrow\tau_i+\eta,\qquad \tau_{i+1}\cdot\theta_{i+1}\leftarrow\tau_i\cdot\theta_i+\eta\cdot\operatorname*{argmin}_{\alpha\in\mathcal{B}}\|\widehat{F}(\theta_i)\cdot\alpha-\tau_i\cdot\widehat{\nabla}_\theta J(\pi_{\theta_i})\|_2.$$

6: **end for**

7: **Output:** $\{\pi_{\theta_i}\}_{i\in[T+1]}$.

4 Main Results

In this section, we establish the global optimality and convergence for neural policy gradient methods. Hereafter, we assume that the absolute value of the reward function r is upper bounded by an absolute constant $Q_{\max}>0$. As a result, we obtain from (2.1) and (2.2) that $|V^\pi(s,a)|\le Q_{\max}$, $|Q^\pi(s,a)|\le Q_{\max}$, and $|A^\pi(s,a)|\le 2Q_{\max}$ for all π and $(s,a)\in\mathcal{S}\times\mathcal{A}$. In what follows, we show that neural policy gradient converges to a stationary point of $J(\pi_\theta)$ with respect to θ at a sublinear rate. We further characterize the geometry of $J(\pi_\theta)$ and establish the global optimality of the obtained stationary point. We defer the global optimality and convergence of neural natural policy gradient to §A.

In the sequel, we study the convergence of neural policy gradient, i.e., Algorithm 1 with (3.6) as the actor update, where $G(\theta)=I_{md}$. In what follows, we lay out a regularity condition on the action-value function Q^π.

Assumption 4.1 (Action-Value Function Class)**.** We define

$$\mathcal{F}_{R,\infty}=\bigg\{f(s,a)=f_0(s,a)+\int\mathbb{1}\big\{w^\top(s,a)>0\big\}\cdot(s,a)^\top\iota(w)\mathrm{d}\mu(w):\|\iota(w)\|_\infty\le R/\sqrt{d}\bigg\},$$

where $\mu\colon\mathbb{R}^d\to\mathbb{R}$ is the density function of the Gaussian distribution $N(0,I_d/d)$, $f_0(\cdot,\cdot)=f((\cdot,\cdot);W_{\mathrm{init}})$ is the two-layer neural network corresponding to the initial parameter W_{init}, and $\iota:\mathbb{R}^d\to\mathbb{R}^d$ together with f_0 parameterizes the element of $\mathcal{F}_{R,\infty}$. We assume that $Q^\pi\in\mathcal{F}_{R,\infty}$ for all π.

Assumption 4.1 is a mild regularity condition on Q^π, as $\mathcal{F}_{R,\infty}$ captures a sufficiently general family of functions, which constitute a subset of the reproducing kernel Hilbert space (RKHS) induced by the random feature $\mathbb{1}\{w^\top(s,a) > 0\} \cdot (s,a)$ with $w \sim N(0, I_d/d)$ (Rahimi and Recht, 2008; 2009) up to the shift of f_0. Similar assumptions are imposed in the analysis of batch reinforcement learning in RKHS (Farahmand et al., 2016).

In what follows, we lay out a regularity condition on the state visitation measure ν_π and the stationary state distribution ϱ_π.

Assumption 4.2 (Regularity Condition on ν_π and ϱ_π)**.** Let π and $\widetilde{\pi}$ be two arbitrary policies. We assume that there exists an absolute constant $c > 0$ such that

$$\mathbb{E}_{\widetilde{\pi}\cdot\nu_\pi}\Big[\mathbb{1}\big\{|y^\top(s,a)| \le u\big\}\Big] \le c \cdot u/\|y\|_2,$$
$$\mathbb{E}_{\widetilde{\pi}\cdot\varrho_\pi}\Big[\mathbb{1}\big\{|y^\top(s,a)| \le u\big\}\Big] \le c \cdot u/\|y\|_2, \quad \forall y \in \mathbb{R}^d,\ \forall u > 0.$$

Here the expectations are taken over the joint distributions $\widetilde{\pi}(\cdot\,|\,\cdot) \cdot \nu_\pi(\cdot)$ and $\widetilde{\pi}(\cdot\,|\,\cdot) \cdot \varrho_\pi(\cdot)$ over $\mathcal{S} \times \mathcal{A}$, respectively.

Assumption 4.2 essentially imposes a regularity condition on the Markov transition kernel $\mathcal{P}$ of the MDP as $\mathcal{P}$ determines ν_π and ϱ_π for all π. Such a regularity condition holds if both ν_π and ϱ_π have upper-bounded density functions for all π.

After introducing these regularity conditions, we present the following proposition adapted from Cai et al. (2019), which characterizes the convergence of neural TD for the critic update.

Proposition 4.3 (Convergence of Critic Update)**.** We set $\eta_{\mathrm{TD}} = \min\{(1-\gamma)/8, 1/\sqrt{T_{\mathrm{TD}}}\}$ in Algorithm 1. Let Q_{ω_i} be the output of the i-th critic update in Line 3 of Algorithm 1, which is an estimator of $Q^{\pi_{\theta_i}}$ obtained by Algorithm 2 with T_{TD} iterations. Under Assumptions 4.1 and 4.2, it holds for $T_{\mathrm{TD}} = \Omega(m)$ that

$$\mathbb{E}_{\mathrm{init}}\big[\|Q_{\omega_i} - Q^{\pi_{\theta_i}}\|_{\varsigma_i}^2\big] = \mathcal{O}(R^3 \cdot m^{-1/2} + R^{5/2} \cdot m^{-1/4}), \tag{4.1}$$

where ς_i is the stationary state-action distribution corresponding to π_{θ_i}. Here the expectation is taken over the random initialization.

Proof. See §F.1 for a detailed proof. □

Cai et al. (2019) show that the error of the critic update consists of two parts, namely the approximation error of two-layer neural networks and the algorithmic error of neural TD. The former decays as the width m grows, while the latter decays as the number of neural TD iterations T_{TD} in Algorithm 2 grows. By setting $T_{\mathrm{TD}} = \Omega(m)$, the algorithmic error in (4.1) of Proposition 4.3 is dominated by the approximation error. In contrast with Cai et al. (2019), we obtain a more refined convergence characterization under the more restrictive assumption that $Q^\pi \in \mathcal{F}_{R,\infty}$. Specifically, such a restriction allows us to obtain the upper bound of the mean squared error in (4.1) of Proposition 4.3.

It now remains to establish the convergence of the actor update, which involves the estimator $\widehat{\nabla}_\theta J(\pi_{\theta_i})$ of the policy gradient $\nabla_\theta J(\pi_{\theta_i})$ based on $\{(s_\ell, a_\ell)\}_{\ell\in[B]}$. We introduce the following regularity condition on the variance of $\widehat{\nabla}_\theta J(\pi_{\theta_i})$.

Assumption 4.4 (Variance Upper Bound)**.** Recall that σ_i is the state-action visitation measure corresponding to π_{θ_i} for all $i \in [T]$. Let $\xi_i = \widehat{\nabla}_\theta J(\pi_{\theta_i}) - \mathbb{E}[\widehat{\nabla}_\theta J(\pi_{\theta_i})]$, where $\widehat{\nabla}_\theta J(\pi_{\theta_i})$ is defined in (3.7). We assume that there exists an absolute constant $\sigma_\xi > 0$ such that $\mathbb{E}[\|\xi_i\|_2^2] \le \tau_i^2 \cdot \sigma_\xi^2/B$ for all $i \in [T]$. Here the expectations are taken over σ_i given θ_i and ω_i.

Assumption 4.4 is a mild regularity condition. Such a regularity condition holds if the Markov chain that generates $\{(s_\ell, a_\ell)\}_{\ell\in[B]}$ mixes sufficiently fast and $Q_{\omega_i}(s,a)$ with $(s,a) \sim \sigma_i$ have upper bounded second moments for all $i \in [T]$. Zhang et al. (2019) verify that under certain regularity conditions, similar unbiased policy gradient estimators have almost surely upper bounded norms, which implies Assumption 4.4. Similar regularity conditions are also imposed in the analysis of policy gradient methods by Xu et al. (2019a;b).

In what follows, we impose a regularity condition on the discrepancy between the state-action visitation measure and the stationary state-action distribution corresponding to the same policy.

Assumption 4.5 (Regularity Condition on σ_i and ς_i)**.** We assume that there exists an absolute constant $\kappa > 0$ such that

$$\left\{\mathbb{E}_{\varsigma_i}\left[\left(\frac{\mathrm{d}\sigma_i}{\mathrm{d}\varsigma_i}(s,a)\right)^2\right]\right\}^{1/2} \leq \kappa, \quad \forall i \in [T]. \tag{4.2}$$

Here $\mathrm{d}\sigma_i/\mathrm{d}\varsigma_i$ is the Radon-Nikodym derivative of σ_i with respect to ς_i.

We highlight that if the MDP is initialized at the stationary distribution ς_i, the state-action visitation measure σ_i is the same as ς_i. Meanwhile, if the induced Markov state-action chain mixes sufficiently fast, such an assumption also holds. A similar regularity condition is imposed by Scherrer (2013), which assumes that the L_∞-norm of $\mathrm{d}\sigma_i/\mathrm{d}\varsigma_i$ is upper bounded, whereas we only assume that its L_2-norm is upper bounded.

Meanwhile, we impose the following regularity condition on the smoothness of the expected total reward $J(\pi_\theta)$ with respect to θ.

Assumption 4.6 (Lipschitz Continuous Policy Gradient)**.** We assume that $\nabla_\theta J(\pi_\theta)$ is L-Lipschitz continuous with respect to θ, where $L > 0$ is an absolute constant.

Such an assumption holds when the transition probability $\mathcal{P}(\cdot \mid s, a)$ and the reward function r are both Lipschitz continuous with respect to their inputs (Pirotta et al., 2015). Also, Karimi et al. (2019); Zhang et al. (2019); Xu et al. (2019b); Agarwal et al. (2019) verify the Lipschitz continuity of the policy gradient under certain regularity conditions.

Note that we restrict θ to the parameter space $\mathcal{B}$. Here we call $\widehat{\theta} \in \mathcal{B}$ a stationary point of $J(\pi_\theta)$ if it holds for all $\theta \in \mathcal{B}$ that $\nabla_\theta J(\pi_{\widehat{\theta}})^\top(\theta - \widehat{\theta}) \leq 0$. We now show that the sequence $\{\theta_i\}_{i\in[T+1]}$ generated by neural policy gradient converges to a stationary point at a sublinear rate.

Theorem 4.7 (Convergence to Stationary Point)**.** We set $\tau_i = 1$, $\eta = 1/\sqrt{T}$, $\eta_{\mathrm{TD}} = \min\{(1-\gamma)/8, 1/\sqrt{T_{\mathrm{TD}}}\}$, $T_{\mathrm{TD}} = \Omega(m)$, and $\mathcal{B} = \{\alpha : \|\alpha - W_{\mathrm{init}}\|_2 \leq R\}$ by Algorithm 1, where the actor update is given in (3.6) with $G(\theta) = I_{md}$. For all $i \in [T]$, we define

$$\rho_i = \eta^{-1} \cdot \Big[\Pi_{\mathcal{B}}\big(\theta_i + \eta \cdot \nabla_\theta J(\pi_{\theta_i})\big) - \theta_i\Big] \in \mathbb{R}^{md}, \tag{4.3}$$

where $\Pi_{\mathcal{B}} \colon \mathbb{R}^{md} \to \mathcal{B}$ is the projection operator onto $\mathcal{B} \subseteq \mathbb{R}^{md}$. Under the assumptions of Proposition 4.3 and Assumptions 4.4-4.6, for $T \geq 4L^2$ we have

$$\min_{i\in[T]} \mathbb{E}\big[\|\rho_i\|_2^2\big] \leq 8/\sqrt{T} \cdot \mathbb{E}\big[J(\pi_{\theta_{T+1}}) - J(\pi_{\theta_1})\big] + 8\sigma_\xi^2/B + \varepsilon_Q(T),$$

where κ is defined in (4.2) of Assumption 4.2 and $\varepsilon_Q(T) = \kappa \cdot \mathcal{O}(R^{5/2} \cdot m^{-1/4} \cdot T^{1/2} + R^{9/4} \cdot m^{-1/8} \cdot T^{1/2})$. Here the expectations are taken over all the randomness.

Proof. See §D.1 for a detailed proof. □

By Theorem 4.7 with $m = \Omega(T^8 \cdot R^{18})$ and $B = \Omega(\sqrt{T})$, we obtain $\min_{i\in[T]} \mathbb{E}[\|\rho_i\|_2^2] = \mathcal{O}(1/\sqrt{T})$. Therefore, when the two-layer neural networks are sufficiently wide and the batch size B is sufficiently large, neural policy gradient achieves a $1/\sqrt{T}$-rate of convergence. Moreover, ρ_i defined in (4.3) is known as the gradient mapping at θ_i (Nesterov, 2018). It is known that $\widehat{\theta} \in \mathcal{B}$ is a stationary point if and only if the gradient mapping at $\widehat{\theta}$ is a zero vector. Therefore, (a subsequence of) $\{\theta_i\}_{i\in[T+1]}$ converges to a stationary point $\widehat{\theta} \in \mathcal{B}$ as $\min_{i\in[T]} \mathbb{E}[\|\rho_i\|_2^2]$ converges to zero. In other words, neural policy gradient converges to a stationary point at a $1/\sqrt{T}$-rate. Also, we remark that the projection operator in the actor update is adopted only for the purpose of simplicity, which can be removed with more refined analysis. Moreover, the projection-free version of neural policy gradient converges to a stationary point at a similar sublinear rate. See §G for details.

We now characterize the global optimality of the obtained stationary point $\widehat{\theta}$. To this end, we compare the expected total reward of $\pi_{\widehat{\theta}}$ with that of the global optimum π^* of $J(\pi)$.

Theorem 4.8 (Global Optimality of Stationary Point)**.** Let $\widehat{\theta} \in \mathcal{B}$ be a stationary point of $J(\pi_\theta)$. It holds that

$$(1-\gamma) \cdot \big(J(\pi^*) - J(\pi_{\widehat{\theta}})\big) \leq 2Q_{\max} \cdot \inf_{\theta \in \mathcal{B}} \|u_{\widehat{\theta}}(\cdot,\cdot) - \phi_{\widehat{\theta}}(\cdot,\cdot)^\top \theta\|_{\sigma_{\pi_{\widehat{\theta}}}},$$

where $Q_{\max}$ is the upper bound of $|r|$ and $u_{\widehat{\theta}}\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined as

$$u_{\widehat{\theta}}(s,a) = \frac{\mathrm{d}\sigma_{\pi^*}}{\mathrm{d}\sigma_{\pi_{\widehat{\theta}}}}(s,a) - \frac{\mathrm{d}\nu_{\pi^*}}{\mathrm{d}\nu_{\pi_{\widehat{\theta}}}}(s) + \phi_{\widehat{\theta}}(s,a)^\top \widehat{\theta}, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}. \tag{4.4}$$

Here $\mathrm{d}\sigma_{\pi^*}/\mathrm{d}\sigma_{\pi_{\widehat{\theta}}}$ and $\mathrm{d}\nu_{\pi^*}/\mathrm{d}\nu_{\pi_{\widehat{\theta}}}$ are the Radon-Nikodym derivatives, and $\|\cdot\|_{\sigma_{\pi_{\widehat{\theta}}}}$ is the $L_2(\sigma_{\pi_{\widehat{\theta}}})$-norm.

Proof. See §D.2 for a detailed proof. □

To understand Theorem 4.8, we highlight that for $\theta, \widehat{\theta} \in \mathcal{B}$, the function $\phi_{\widehat{\theta}}(\cdot,\cdot)^\top \theta$ is well approximated by the overparameterized two-layer neural network $f((\cdot,\cdot);\theta)$. See Corollary E.4 for details. Therefore, the global optimality of $\pi_{\widehat{\theta}}$ depends on the error of approximating $u_{\widehat{\theta}}$ with an overparameterized two-layer neural network. Specifically, if $u_{\widehat{\theta}}$ is well approximated by an overparameterized two-layer neural network, then $\pi_{\widehat{\theta}}$ is nearly as optimal as π^*. In the following corollary, we formally establish a sufficient condition for any stationary point $\widehat{\theta}$ to be globally optimal.

Theorem 4.9 (Global Optimality of Stationary Point)**.** Let $\widehat{\theta} \in \mathcal{B}$ be a stationary point of $J(\pi_\theta)$. We assume that $u_{\widehat{\theta}} \in \mathcal{F}_{R,\infty}$ in Theorem 4.8. Under Assumption 4.2, it holds that

$$(1-\gamma) \cdot \mathbb{E}_{\text{init}}\big[J(\pi^*) - J(\pi_{\widehat{\theta}})\big] = \mathcal{O}(R^{3/2} \cdot m^{-1/4}).$$

More generally, without assuming $u_{\widehat{\theta}} \in \mathcal{F}_{R,\infty}$ in Theorem 4.8, under Assumption 4.2, it holds that

$$(1-\gamma) \cdot \mathbb{E}_{\text{init}}\big[J(\pi^*) - J(\pi_{\widehat{\theta}})\big] = \mathcal{O}(R^{3/2} \cdot m^{-1/4}) + \mathbb{E}_{\text{init}}\big[\|\Pi_{\mathcal{F}_{R,\infty}} u_{\widehat{\theta}} - u_{\widehat{\theta}}\|_{\sigma_{\pi_{\widehat{\theta}}}}\big].$$

Here the expectations are taken over the random initialization, and $\Pi_{\mathcal{F}_{R,\infty}}$ is the projection operator onto $\mathcal{F}_{R,\infty}$ with respect to the $L_2(\sigma_{\pi_{\widehat{\theta}}})$-norm.

Proof. See §H.2 for a detailed proof. □

By Theorem 4.9, a stationary point $\widehat{\theta}$ is globally optimal if $u_{\widehat{\theta}} \in \mathcal{F}_{R,\infty}$ and $m \to \infty$. Moreover, following from the definition of ρ_i in (4.3) of Theorem 4.7, we obtain that

$$\nabla_\theta J(\pi_{\theta_i})^\top (\theta - \theta_i) \leq (2R + 2\eta \cdot Q_{\max}) \cdot \|\rho_i\|_2, \quad \forall \theta \in \mathcal{B}. \tag{4.5}$$

See §H.3 for a detailed proof of (4.5). Since $\|\rho_i\|_2 = 0$ implies that θ_i is a stationary point, the right-hand side of (4.5) quantifies the deviation of θ_i from a stationary point $\widehat{\theta}$. Following similar analysis to §D.2 and §H.2, if $u_{\theta_i} \in \mathcal{F}_{R,\infty}$ for all $i \in [T]$, we obtain that

$$(1-\gamma) \cdot \min_{i \in [T]} \mathbb{E}\big[J(\pi^*) - J(\pi_{\theta_i})\big] = \mathcal{O}(R^{3/2} \cdot m^{-1/4}) + (2R + 2\eta \cdot Q_{\max}) \cdot \min_{i \in [T]} \mathbb{E}\big[\|\rho_i\|_2\big].$$

Thus, by invoking Theorem 4.7, it holds for sufficiently large m and B that the expected total reward $J(\pi_{\theta_i})$ converges to the global optimum $J(\pi^*)$ at a $1/T^{1/4}$-rate.

REFERENCES

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

Allen-Zhu, Z., Li, Y. and Liang, Y. (2018a). Learning and generalization in overparameterized neural networks, going beyond two layers. *arXiv preprint arXiv:1811.04918*.

Allen-Zhu, Z., Li, Y. and Song, Z. (2018b). A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*.

Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, **10** 251–276.

Antos, A., Szepesvári, C. and Munos, R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, **71** 89–129.

Arora, S., Du, S. S., Hu, W., Li, Z. and Wang, R. (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*.

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, **14** 115–133.

Baxter, J. and Bartlett, P. L. (2000). Direct gradient-based reinforcement learning. In *International Symposium on Circuits and Systems*.

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

Borkar, V. S. (2009). *Stochastic Approximation: A Dynamical Systems Viewpoint*, vol. 48. Springer.

Bu, J., Mesbahi, A., Fazel, M. and Mesbahi, M. (2019). LQR through the lens of first order methods: Discrete-time case. *arXiv preprint arXiv:1907.08921*.

Cai, Q., Yang, Z., D. Lee, J. and Wang, Z. (2019). Neural temporal-difference learning converges to global optima. *arXiv preprint arXiv:1905.10027*.

Cao, Y. and Gu, Q. (2019a). Generalization bounds of stochastic gradient descent for wide and deep neural networks. *arXiv preprint arXiv:1905.13210*.

Cao, Y. and Gu, Q. (2019b). A generalization theory of gradient descent for learning over-parameterized deep ReLU networks. *arXiv preprint arXiv:1902.01384*.

Castro, D. D. and Meir, R. (2010). A convergent online single time scale actor critic algorithm. *Journal of Machine Learning Research*, **11** 367–410.

Chen, J. and Jiang, N. (2019). Information-theoretic considerations in batch reinforcement learning. *arXiv preprint arXiv:1905.00360*.

Chizat, L. and Bach, F. (2018). A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*.

Daniely, A. (2017). SGD learns the conjugate kernel class of the network. In *Advances in Neural Information Processing Systems*.

Du, S. S., Lee, J. D., Li, H., Wang, L. and Zhai, X. (2018a). Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*.

Du, S. S., Zhai, X., Poczos, B. and Singh, A. (2018b). Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*.

Duan, Y., Chen, X., Houthooft, R., Schulman, J. and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I. et al. (2018). IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*.

Fan, J., Ma, C. and Zhong, Y. (2019). A selective overview of deep learning. *arXiv preprint arXiv:1904.05526*.

Farahmand, A.-m., Ghavamzadeh, M., Szepesvári, C. and Mannor, S. (2016). Regularized policy iteration with nonparametric function spaces. *Journal of Machine Learning Research*, **17** 4809–4874.

Farahmand, A.-m., Szepesvári, C. and Munos, R. (2010). Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*.

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

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, **2** 183–192.

Haarnoja, T., Tang, H., Abbeel, P. and Levine, S. (2017). Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*.

Jacot, A., Gabriel, F. and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*.

Kakade, S. and Langford, J. (2002). Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*.

Kakade, S. M. (2002). A natural policy gradient. In *Advances in Neural Information Processing Systems*.

Karimi, B., Miasojedow, B., Moulines, E. and Wai, H.-T. (2019). Non-asymptotic analysis of biased stochastic approximation scheme. *arXiv preprint arXiv:1902.00629*.

Klusowski, J. M. and Barron, A. R. (2016). Risk bounds for high-dimensional ridge function combinations including neural networks. *arXiv preprint arXiv:1607.01434*.

Konda, V. (2002). *Actor-Critic Algorithms*. Ph.D. thesis, Massachusetts Institute of Technology.

Konda, V. R. and Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in Neural Information Processing Systems*.

Kushner, H. and Yin, G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*, vol. 35. Springer Science & Business Media.

Lazaric, A., Ghavamzadeh, M. and Munos, R. (2016). Analysis of classification-based policy iteration algorithms. *Journal of Machine Learning Research*, **17** 583–612.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J. and Sohl-Dickstein, J. (2018). Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Sohl-Dickstein, J. and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*.

Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

Li, Y. and Liang, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*.

Liu, B., Cai, Q., Yang, Z. and Wang, Z. (2019). Neural proximal/trust region policy optimization attains globally optimal policy. *arXiv preprint arXiv:1906.10306*.

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

Martens, J. and Grosse, R. (2015). Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*.

Mnih, V., Badia, A. P., 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*.

Munos, R. and Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, **9** 815–857.

Nesterov, Y. (2018). *Lectures on Convex Optimization*. Springer.

Pan, X. and Srikumar, V. (2016). Expressiveness of rectifier networks. In *International Conference on Machine Learning*.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M. and Restelli, M. (2018). Stochastic variance-reduced policy gradient. *arXiv preprint arXiv:1806.05618*.

Peters, J. and Schaal, S. (2006). Policy gradient methods for robotics. In *International Conference on Intelligent Robots and Systems*.

Peters, J. and Schaal, S. (2008). Natural actor-critic. *Neurocomputing*, **71** 1180–1190.

Pirotta, M., Restelli, M. and Bascetta, L. (2015). Policy gradient in Lipschitz Markov decision processes. *Machine Learning*, **100** 255–283.

Puterman, M. L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*.

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*.

Scherrer, B. (2013). On the performance bounds of some policy search dynamic programming algorithms. *arXiv preprint arXiv:1306.0539*.

Scherrer, B., Ghavamzadeh, M., Gabillon, V., Lesner, B. and Geist, M. (2015). Approximate modified policy iteration and its application to the game of Tetris. *Journal of Machine Learning Research*, **16** 1629–1676.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*.

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

Shen, Z., Ribeiro, A., Hassani, H., Qian, H. and Mi, C. (2019). Hessian aided policy gradient. In *International Conference on Machine Learning*.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, **529** 484.

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

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, **3** 9–44.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT press.

Sutton, R. S., McAllester, D. A., Singh, S. P. and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

Szepesvári, C. and Munos, R. (2005). Finite time bounds for sampling based fitted value iteration. In *International Conference on Machine Learning*.

Tsitsiklis, J. N. and Van Roy, B. (1997). Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*.

Tu, S. and Recht, B. (2018). The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint. *arXiv preprint arXiv:1812.03565*.

van Handel, R. (2014). *Probability in High Dimension*. Princeton University.

Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Wu, Y., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D. and Silver, D. (2019). AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.

Wagner, P. (2011). A reinterpretation of the policy oscillation phenomenon in approximate policy iteration. In *Advances in Neural Information Processing Systems*.

Wagner, P. (2013). Optimistic policy iteration and natural actor-critic: A unifying view and a non-optimality result. In *Advances in Neural Information Processing Systems*.

Wang, W. Y., Li, J. and He, X. (2018). Deep reinforcement learning for NLP. In *Association for Computational Linguistics*.

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

Wu, L., Ma, C. and Weinan, E. (2018). How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S. and Ba, J. (2017). Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in Neural Information Processing Systems*.

Xu, P., Gao, F. and Gu, Q. (2019a). An improved convergence analysis of stochastic variance-reduced policy gradient. *arXiv preprint arXiv:1905.12615*.

Xu, P., Gao, F. and Gu, Q. (2019b). Sample efficient policy gradient methods with recursive variance reduction. *arXiv preprint arXiv:1909.08610*.

Yang, Z., Chen, Y., Hong, M. and Wang, Z. (2019a). On the global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. *arXiv preprint arXiv:1907.06246*.

Yang, Z., Xie, Y. and Wang, Z. (2019b). A theoretical analysis of deep Q-learning. *arXiv preprint arXiv:1901.00137*.

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*.

Zou, D., Cao, Y., Zhou, D. and Gu, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *arXiv preprint arXiv:1811.08888*.