THE IMPACT OF APPROXIMATION ERRORS ON WARM-START REINFORCEMENT LEARNING: A FINITE-TIME ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Warm-Start reinforcement learning (RL), aided by a prior policy obtained from offline training, is emerging as a promising RL approach for practical applications. Recent empirical studies have demonstrated that the performance of Warm-Start RL can be improved *quickly* in some cases but become *stagnant* in other cases, calling for a fundamental understanding, especially when the function approximation is used. To fill this void, we take a finite time analysis approach to quantify the impact of approximation errors on the learning performance of Warm-Start RL. Specifically, we consider the widely used Actor-Critic (A-C) method with a prior policy. We first quantify the approximation errors in the Actor update and the Critic update, respectively. Next, we cast the Warm-Start A-C algorithm as Newton's method with perturbation, and study the impact of the approximation errors on the finite-time learning performance with inaccurate Actor/Critic updates. Under some general technical conditions, we obtain lower bounds on the sub-optimality gap of the Warm-Start A-C algorithm to quantify the impact of the bias and error propagation. We also derive the upper bounds, which provide insights on achieving the desired finite-learning performance in the Warm-Start A-C algorithm.

1 INTRODUCTION

Online reinforcement learning (RL) (Kaelbling et al., 1996; Sutton & Barto, 2018) often faces the formidable challenge of high sample complexity and intensive computational cost (Kumar et al., 2020; Xie et al., 2021), which hinders its applicability in real-world tasks. Indeed, this is the case in portfolio management (Choi et al., 2009), vehicles control (Wu et al., 2017; Shalev-Shwartz et al., 2016) and other time-sensitive settings (Li, 2017; Garcıa & Fernández, 2015). To tackle this challenge, Warm-Start RL has recently garnered much attention (Nair et al., 2020; Gelly & Silver, 2007; Uchendu et al., 2022), by enabling online policy adaptation from an initial policy pre-trained using offline data (e.g., via behavior cloning or offline RL). One main insight of Warm-Start RL is that online learning can be significantly accelerated, thanks to the bootstrapping by an initial policy.

Despite the encouraging empirical successes (Silver et al., 2017; 2018; Uchendu et al., 2022), a fundamental understanding of the learning performance of Warm-Start RL is lacking, especially in the practical settings with function approximation by neural networks. In this work, we focus on the widely used Actor-Critic (A-C) method (Grondman et al., 2012; Peters & Schaal, 2008), which combines the merits of both policy iteration and value iteration approaches (Sutton & Barto, 2018) and has great potential for RL applications (Uchendu et al., 2022). Notably, in the framework of abstract dynamic programming (ADP) (Bertsekas, 2022a), the policy iteration method (Sutton et al., 1999) has been studied extensively, for warm-start learning under the assumption of accurate updates. In such a setting, policy iteration can be regarded as a second-order method in convex optimization (Grand-Clément, 2021) from the perspective of ADP, and can achieve *super-linear* convergence rate (Santos & Rust, 2004; Puterman & Brumelle, 1979; Boyd et al., 2004). Nevertheless, when the A-C method is implemented in practical applications, the approximation errors are inevitable in the Actor/Critic updates due to many implementation issues, including function approximation using neural networks, the finite sample size, and the finite number of gradient iterations. Moreover, the error propagation from iteration to iteration may exacerbate the 'slowing down' of the convergence and have intricate impact therein. Clearly, the (stochastic) accumulated errors may throttle the

convergence rate significantly and degrade the learning performance dramatically (Fujimoto et al., 2018; Uehara et al., 2021; Dalal et al., 2020; Doan et al., 2019). Thus, it is of great importance to characterize the learning performance of Warm-Start RL in practical scenarios; and the primary objective of this study is to take steps to build a fundamental understanding of the impact of the approximation errors on the finite-time sub-optimality gap for the Warm-Start A-C algorithm, i.e., the performance gap between the learnt policy and the optimal policy.

Main Contributions.

• **We first focus on the characterization of the approximation errors via finite time analysis, based on which we quantify its impact on the sub-optimality gap of the A-C algorithm in Warm-Start RL.** In particular, we analyze the A-C algorithm in a more realistic setting where the samples are Markovian in the rollout trajectories for the Critic update (different from the widely used i.i.d. assumption). Further, we consider that the Actor update and the Critic update take place on the *single-time scale*, indicating that the time-scale decomposition is not applicable to the finite-time analysis here. We tackle these challenges using recent advances on Bernstein's Inequality for Markovian samples (Jiang et al., 2018; Fan et al., 2021b). By delving into the coupling due to the interleaved updates of the Actor and the Critic, we provide upper bounds on the approximation errors in the Critic update and the Actor update of online exploration, respectively, from which we pinpoint the root causes of the approximation errors.

• **We analyze the impact of the approximation errors on the finite-time learning performance of Warm-Start A-C.** Based on the approximation error characterization, we treat the Warm-Start A-C algorithm as Newton's method with perturbation, and study the impact of the approximation errors on the finite-time learning performance of Warm-Start A-C. For the case when the approximation errors are biased, we derive lower bounds on the sub-optimality gap, which reveals that even with a sufficiently good warm-start, the performance gap of online policy adaptation to the optimal policy is still bounded away from zero when the biases are not negligible. Further, we also derive the upper bounds, which shed light on designing Warm-Start A-C to achieve desired finite-time learning performance. We present the experiments results to further elucidate our findings in Appendix K.

Related Work. (Warm-Start RL) AlphaZero (Silver et al., 2017) is one of the most remarkable successes in Warm-Start RL. In a line of very recent works (Gupta et al., 2020)(Ijspeert et al., 2002)(Kim et al., 2013) on Warm-Start RL, the policy is initialized via behavior cloning from offline data and then is fine-tuned with online reinforcement learning. A variant of this scheme is proposed in Advanced Weighted Actor Critic (Nair et al., 2020) which enables quick learning of skills across a suite of benchmark tasks. In the same spirit, Offline-Online Ensemble (Lee et al., 2022) leverages multiple Q-functions trained pessimistically offline as the initial function approximation for online learning. Jump-start RL (Uchendu et al., 2022) utilizes a guided-policy to initialize online RL in the early phase with a separate online exploration-policy. The guided-policy will be abandoned as the online exploration-policy improves. However, a fundamental characterization of the finite-time performance of Warm-Start RL is still lacking. Recent work (Xie et al., 2021) provides a quantitative understanding on the policy fine-tuning problem in episodic Markov Decision Processes (MDPs) and establishes the lower bound for the sample complexity, where no function approximation is used. Our work aims to take steps to quantify the impact of approximation error on online RL when a warm-start policy is given.

(Actor-Critic as Newton's Method) The intrinsic connection between the A-C method and Newton's method can be traced back to the convergence analysis of policy iteration in MDPs with continuous action spaces (Puterman & Brumelle, 1979). The connection is further examined later in a special MDP with discretized continuous state space (Santos & Rust, 2004). Recent work (Bertsekas, 2022b) points out that the success of Warm-Start RL, e.g., AlphaZero, can be attributed to the equivalence between policy iteration and Newton's method in the ADP framework, which leads to the superlinear convergence rate for online policy adaptation. Under the generalized differentiable assumption, it has also been proved theoretically that policy iteration is the instances of semi-smooth Newton-type methods to solve the Bellman equation (Gargiani et al., 2022). While some prior works (Grand-Clément, 2021) have provided theoretical investigation of the connections between policy iteration and Newton's Method, the studies are carried out in the abstract dynamic programming (ADP) framework, assuming accurate updates in iterations. Departing from the ADP framework, this work treats the A-C algorithm as Newton's method in the presence of approximation errors, and focuses on the finite-time learning performance of Warm-Start RL.

(Finite-time analysis for Actor-Critic methods) Among the existing works on the finite time analysis of A-C methods with function approximation, (Yang et al., 2019) establishes the global convergence under the linear quadratic regulator. (Wang et al., 2020) proves the convergence behavior when both Actor and Critic are approximated by overparameterized neural networks. (Kumar et al., 2019) considers the sample complexity under i.i.d. assumptions where the Actor update and Critic update can be 'decoupled'. (Fu et al., 2020) focuses on the more general single-time scale setting but constrains the policy function approximation in the energy based function class.

2 BACKGROUND

Markov Decision Processes. We consider a MDP defined by a tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S} = \{1, 2, \cdots, n\}$, $n < \infty$ and $\mathcal{A} = \{1, 2, \cdots, A\}$, $A < \infty$ represent the finite state space and finite action space, respectively. $P(s'|s, a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the probability of the transition from state s to state s' by applying action a and $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the corresponding reward. $\gamma \in (0, 1]$ is the discount factor. At each step t, an agent moves from the current state s_t to next state s_{t+1} by taking an action a_t following the policy $\pi : \mathcal{S} \to \mathcal{A}$ and receives the reward r_t. In the Warm-Start RL, we assume that the initial policy π_0 is given, e.g., in the form of a neural network (Li, 2017), and obtained by offline training. For brevity, we use bold symbols $\boldsymbol{r}_\pi \in \mathbb{R}^n : [r_\pi]_s = r(s, \pi(s))$ and $\boldsymbol{P}_\pi \in \mathbb{R}^{n \times n} : [P^\pi]_{s,s'} \triangleq P(s'|s, \pi(s))$ to denote the reward vector and the transition matrix induced by policy π. We further denote by $d^\pi : \mathcal{S} \to [0, 1]$ and $\rho^\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ the stationary state distribution and state-action transition distribution induced by policy π. We use ρ_0 to represent the initial state distribution. We use $\| \cdot \|$ or $\| \cdot \|_2$ to represent the 2-norm.

Value Functions. For any policy π, define the value function $v^\pi(s) : \mathcal{S} \to \mathbb{R}$ as $v^\pi(s) = \mathbf{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)} [\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s]$ to measure the average accumulative reward staring from state s by following policy π. We define Q-function $Q^\pi(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as $Q^\pi(s, a) = \mathbf{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a]$ to represent the expected return when the action a is chosen at the state s. By using the transition matrix and reward vector defined above, we have the compact form of the value function $\boldsymbol{v}^\pi = (\boldsymbol{I} - \gamma \boldsymbol{P}_\pi)^{-1} \boldsymbol{r}_\pi$, where $\boldsymbol{I} \in \mathbb{R}^{n \times n}$ is the identity matrix and $\boldsymbol{v}^\pi \in \mathbb{R}^n$ is the value vector with the component-wise values $[v^\pi]_s \triangleq v^\pi(s)$, with

$$v^\pi(s) \triangleq \mathbf{E}_{a \sim \pi(\cdot|s)} [Q^\pi(s, a)]. \tag{1}$$

The main objective is to find an optimal policy π^* such that the value function is maximized, i.e.,

$$\max_\pi \mathbf{E}_{s \sim \rho_0} [v^\pi(s)] \triangleq \max_\pi \mathbf{E}_{s \sim \rho_0, a \sim \pi(\cdot|s)} [Q^\pi(s, a)]. \tag{2}$$

In what follows, we use both Q-function and value function $v(s)$ for convenience, and the relation between the two is given in Eqn. (2).

Bellman Operator. For $\boldsymbol{v} \in \mathbb{R}^n$, define the Bellman evaluation operator $T^\pi : \mathbb{R}^n \to \mathbb{R}^n$ and the Bellman operator $T : \mathbb{R}^n \to \mathbb{R}^n$ as

$$T^\pi(\boldsymbol{v}) = \boldsymbol{r}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{v}, \; T(\boldsymbol{v}) = \max_\pi \{\boldsymbol{r}_\pi + \gamma \boldsymbol{P}_\pi \boldsymbol{v}\} = \max_\pi T^\pi(\boldsymbol{v}).$$

It is well known that the Bellman operator T is a contraction mapping and has order-preserving property. Note that the Bellman operator T may not be differentiable everywhere due to the $\max$ operator, and the value $\boldsymbol{v}^*$ of the optimal policy π^* is the only fixed point of the Bellman operator T (Puterman, 2014). From the definition of the Bellman Evaluation Operator T^π, we have $\boldsymbol{v}^\pi$ to be the fixed point of T^π, i.e., $\boldsymbol{v}^\pi = T^\pi(\boldsymbol{v}^\pi)$.

2.1 POLICY ITERATION AS NEWTON'S METHOD IN ABSTRACT DYNAMIC PROGRAMMING

Policy iteration carries out policy learning by alternating between two steps: policy improvement and policy evaluation. At time t, the policy evaluation step seeks to learn the value function $\boldsymbol{v}^{\pi_t}$ for the current policy π_t by solving the fixed point equation of the Bellman evaluation operator:

$$\boldsymbol{v} = T^{\pi_t}(\boldsymbol{v}). \tag{3}$$

Denote $\boldsymbol{v}_t = \boldsymbol{v}^{\pi_t}$ for simplicity. Then in the policy improvement step, a new policy π_{t+1} is obtained by maximizing the learnt value function $\boldsymbol{v}_t$ in the policy evaluation step, in a greedy manner, i.e.,

$$\pi_{t+1} = \arg\max_\pi T^\pi(\boldsymbol{v}_t). \tag{4}$$

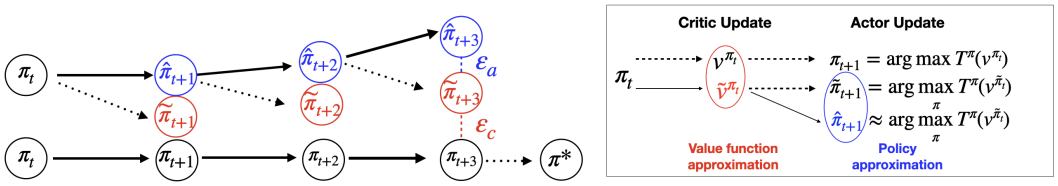

Figure 1: Illustration of error propagation effect in the A-C method: The approximation errors from Critic update ($\mathcal{E}_c$) and Actor update ($\mathcal{E}_a$) are carried forward and may get amplified due to accumulation. (To distinguish the approximation errors between Critic update and Actor update, we use tilde symbol ($\tilde{}$) above variables, such as policy $\widetilde{\pi}$ and value vector $\widetilde{\boldsymbol{v}}$, to represent the policy and the value vector obtained in the presence of Critic update error. We use hat symbol ($\hat{}$) above the variables to represent the results with approximation error in Actor update.)

To introduce the connection between policy iteration and Newton's Method, we first define operator $F : \boldsymbol{v} \to \boldsymbol{v} - T(\boldsymbol{v})$ for convenience. As in (Grand-Clément, 2021; Puterman, 2014), F can be viewed as the gradient of an unknown function. Under the assumption that $F(\boldsymbol{v})$ is differentiable at $\boldsymbol{v}$, the Jacobian $\boldsymbol{J}_{\boldsymbol{v}}$ of F at $\boldsymbol{v}$ can be obtained as $\boldsymbol{J}_{\boldsymbol{v}} = I - \gamma \boldsymbol{P}_{\pi(\boldsymbol{v})}$, where $\pi(\boldsymbol{v}) \triangleq \arg\max_{\pi} T^{\pi}(\boldsymbol{v})$. Note that $\boldsymbol{J}_{\boldsymbol{v}}^{-1} = \sum_{i=1}^{\infty} (\gamma \boldsymbol{P}_{\pi(\boldsymbol{v})})^i$ is invertible (Puterman, 2014). Since it can be shown that $\boldsymbol{v}^{\pi_{t+1}} = (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{t+1}})^{-1} \boldsymbol{r}_{\pi_{t+1}}$ for the policy evaluation of π_{t+1}, we have that,

$$\boldsymbol{v}^{\pi_{t+1}} = \boldsymbol{v}^{\pi_t} - \boldsymbol{J}_{\boldsymbol{v}^{\pi_t}}^{-1} \left(\boldsymbol{v}^{\pi_t} - T\left(\boldsymbol{v}^{\pi_t}\right) \right) = \boldsymbol{v}^{\pi_t} - \boldsymbol{J}_{\boldsymbol{v}^{\pi_t}}^{-1} F(\boldsymbol{v}^{\pi_t}), \tag{5}$$

which indicates that the analytic representation of policy iteration in the abstract dynamic programming framework reduces to Newton's Method. It is worth mentioning that the convergence behavior of policy iteration near the optimal value $\boldsymbol{v}^*$ cannot be directly obtained by using the results from convex optimization (Boyd et al., 2004) since the Bellman operator T may not be differentiable at any given value vector $\boldsymbol{v}$. The full proof is included in Appendix A.

2.2 An Illustrative Example of the Error Propagation in Actor-Critic Updates

The A-C method can be viewed as a generalization of policy iteration in ADP, where the Critic update corresponds to the policy evaluation of the current policy and the Actor update performs the policy improvement. In practice, function approximation (e.g., via neural networks) is often used to approximate both the Critic and the Actor, which inevitably incurs approximation errors for the policy update and evaluation. More importantly, the approximation errors could propagate along with the iterative updates in the A-C method. We have the illustrative example to get a more concrete sense of the impact of the approximation errors on the policy update.

As illustrated in Figure 1, for a given policy π_t with the underlying true policy value $\boldsymbol{v}^{\pi_t}$, we denote $\widetilde{\boldsymbol{v}}^{\pi_t}$ as the learnt value estimation of $\boldsymbol{v}^{\pi_t}$ in the Critic step. We further denote π_{t+1} and $\widetilde{\pi}_{t+1}$ as the greedy policy obtained in the Actor update Eqn. (4) by using $\boldsymbol{v}^{\pi_t}$ and $\widetilde{\boldsymbol{v}}^{\pi_t}$, respectively. Let $\hat{\pi}_{t+1}$ be the policy estimation of $\widetilde{\pi}_{t+1}$ with function approximation in the Actor step. Intuitively, π_{t+1} is the underlying true policy update from π_t using one step policy iteration without any error, $\widetilde{\pi}_{t+1}$ is the policy update from π_t with approximation errors in the Critic update, and $\hat{\pi}_{t+1}$ is the policy update from π_t with approximation errors in both the Critic step and the Actor step. To characterize the impact of the approximation errors on the policy update, i.e., the difference between $\boldsymbol{v}^{\pi_{t+1}}$ and $\boldsymbol{v}^{\hat{\pi}_{t+1}}$, we evaluate the Critic error, i.e., the difference between $\boldsymbol{v}^{\pi_{t+1}}$ and $\boldsymbol{v}^{\widetilde{\pi}_{t+1}}$, and the Actor error, i.e., the difference between $\boldsymbol{v}^{\widetilde{\pi}_{t+1}}$ and $\boldsymbol{v}^{\hat{\pi}_{t+1}}$, in a separate manner. More specifically, to quantify the Critic error, we can first have the following update based on the same reasoning with Eqn. (5):

$$\boldsymbol{v}^{\widetilde{\pi}_{t+1}} = \boldsymbol{v}_t - \boldsymbol{J}_{\widetilde{\boldsymbol{v}}_t}^{-1} \left(\boldsymbol{v}_t - (\boldsymbol{r}_{\widetilde{\pi}_{t+1}} + \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}} \boldsymbol{v}_t) \right) \triangleq \boldsymbol{v}_t - \boldsymbol{J}_{\widetilde{\boldsymbol{v}}_t}^{-1} \left(\boldsymbol{v}_t - \widetilde{T}\left(\boldsymbol{v}_t\right) \right),$$

where $\widetilde{T}(\boldsymbol{v}_t) = \boldsymbol{r}_{\widetilde{\pi}_{t+1}} + \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}} \boldsymbol{v}_t$ and $\boldsymbol{J}_{\widetilde{\boldsymbol{v}}_t} = \boldsymbol{I} - \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}}$. Denote the approximation error in the Bellman operator T and the Jacobian $\boldsymbol{J}_{\boldsymbol{v}}$ by $\mathcal{E}_{T,t}$ and $\mathcal{E}_{J,t}$, i.e.,

$$\begin{aligned} \widetilde{T}(\boldsymbol{v}_t) - T(\boldsymbol{v}_t) =& (\boldsymbol{r}_{\widetilde{\pi}_{t+1}} + \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}} \boldsymbol{v}_t) - (\boldsymbol{r}_{\pi_{t+1}} + \gamma \boldsymbol{P}_{\pi_{t+1}} \boldsymbol{v}_t) \triangleq \mathcal{E}_{T,t}, \\ \boldsymbol{J}_{\widetilde{\boldsymbol{v}}_t}^{-1} - \boldsymbol{J}_{\boldsymbol{v}_t}^{-1} =& (\boldsymbol{I} - \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}})^{-1} - (\boldsymbol{I} - \gamma \boldsymbol{P}_{\pi_{t+1}})^{-1} \triangleq \mathcal{E}_{J,t}, \end{aligned}$$

where it is clear that both error terms stem from the function approximation errors in the Critic update. To quantify the Actor error, we assume that

$$\boldsymbol{v}^{\hat{\pi}_{t+1}} = \boldsymbol{v}^{\widetilde{\pi}_{t+1}} + \mathcal{E}_{a,t}, \tag{6}$$

where $\mathcal{E}_{a,t}$ is the error term. Therefore, by casting the A-C method as Newton's method with perturbation, we can characterize the approximation errors on the policy update:

$$\boldsymbol{v}^{\hat{\pi}_{t+1}} = \boldsymbol{v}^{\pi_{t+1}} + \mathcal{E}_{c,t} + \mathcal{E}_{a,t},$$

where $\mathcal{E}_{c,t} \triangleq -\mathcal{E}_{J,t}(\boldsymbol{v}_t - T(\boldsymbol{v}_t)) + (\boldsymbol{J}_{\boldsymbol{v}_t}^{-1} + \mathcal{E}_{J,t})\mathcal{E}_{T,t}$ and $\mathcal{E}_{a,t}$ capture the impact of the approximation error from Critic update step and Actor update step, respectively. Intuitively, as illustrated in Figure 1, both errors from the previous update in the A-C method may propagate to the next update and thus affect the convergence behavior of the algorithm substantially, in contrast to idealized policy iteration without approximation errors. This phenomenon has also been observed in the empirical results (Fujimoto et al., 2018; Thrun & Schwartz, 1993). In this work, we strive to systematically analyze the impact of the approximation errors, through (1) a detailed characterization of the approximation errors in the Critic update and the Actor update in Section 3 and (2) a thorough analysis of the error propagation effect and biases in Section 4.

3 Characterization of Approximation Errors

Actor-Critic Methods with Function Approximation. In what follows, we consider that the policy is parameterized by $\theta \in \Theta$, which in general corresponds to a non-linear function class. Following (Konda & Tsitsiklis, 1999; Peters & Schaal, 2008; Kumar et al., 2019; 2020; Santos & Rust, 2004), the Q-function is parameterized by a linear function class with base function $\phi(s,a)$ and parameter $\omega \in \Omega \subset \mathbb{R}^d$, i.e., $Q_\omega(s,a) = \omega^\top \phi(s,a)$. We note that the modeling of the Q-function via linear value function is often used to extract insight in the A-C method. Similar to the policy iteration, the update in the A-C method alternates between the following two steps [1].

Critic update: The Critic updates its parameter ω to evaluate the current policy π_t, e.g., through m-step ($m \geq 1$) Bellman evaluation operator T^π to the current Q-function estimator (namely, m-step return), which leads to the following update rule at time step t,

$$Q_{\omega_{t+1}}(s,a) \leftarrow \mathbf{E}_{\pi_t}\left[(1-\gamma)\cdot\sum_{i=0}^{m-1}\gamma^i r(s_i,a_i) + \gamma^m \cdot Q_{\omega_t}(s_m,a_m) \mid s_0 = s, a_0 = a\right], \quad (7)$$

$$\omega_{t+1} \leftarrow \arg\min_\omega \mathbf{E}_{(s,a)\sim\rho^{\pi_t}}\left[Q_{\omega_{t+1}}(s,a) - \omega^\top\phi(s,a)\right]^2. \quad (8)$$

Actor update: The Actor is updated through a greedy step to maximize the learnt Q function, i.e.,

$$\pi_{t+1} \leftarrow \arg\max_\pi \mathbf{E}_{(s,a)\sim\rho^\pi}\left[Q_{\omega_{t+1}}(s,a)\right]. \quad (9)$$

3.1 Approximation Error in the Critic update

Solving the minimization problem in Eqn. (8) involves the expectation over the stationary state-action distribution ρ^{π_t} induced by the current policy π_t, which can be approximated by sample average in practice. Therefore, we consider the Critic update below based on two groups of samples, $\{(s_l,a_l)\}_{l=1}^N$ and $\{\tau_l\}_{l=1}^N$ where $\tau_l = \{s_{l,t}, a_{l,t}, r_{l,t}\}_{t=0}^m$, which are collected by following π_t:

$$\omega_{t+1} = \Gamma_R\Bigg\{\left(\textstyle\sum_{l=1}^N \phi(s_l,a_l)\,\phi(s_l,a_l)^\top\right)^{-1} \cdot \textstyle\sum_{l=1}^N\left((1-\gamma)\sum_{i=0}^{m-1}\gamma^i r_{l,i} + \gamma^m Q_{\omega_k}(s_{l,m},a_{l,m})\right)\phi(s_{l,m},a_{l,m})\Bigg\}, \quad (10)$$

where Γ is the projection operator onto the Critic parameter space Ω with radius R in $\mathbb{R}^d$. Since the samples in each trajectory τ_l are obtained via rolllouts, in general the samples in each trajectory follow a Markovian process (Dalal et al., 2018; Kumar et al., 2019).

In what follows, we use ω and $\widetilde{\omega}$ to distinguish the difference between the sample-based update and the solution from Eqn. (8), such that the approximation error in the Critic update can be quantified as $|Q_{\widetilde{\omega}_t} - Q_{\omega_t}|$. We first impose the following standard assumptions on the Bellman evaluation operator T^π, the base function ϕ and the MDP.

[1] We remark that our analysis framework and theoretical results are able to be applied to off-policy setting with the extra assumption on the behavior policy. We include the details in Appendix L.

Assumption 1. *For given Critic parameter ω and policy parameter θ, the following condition holds:*

$$\inf_{\bar{\omega}\in\Omega} \mathbf{E}_{\rho^{\pi_\theta}}\left[\left((T^{\pi_\theta})^m Q_\omega - \bar{\omega}^\top \phi\right)(s,a)\right] = 0,$$

where ρ^{π_θ} is the stationary state-action transition probability induced by policy π_θ.

Assumption 1 (Fu et al., 2020) indicates that the solution of the Critic update given in Eqn. (8) lies in the Critic parameter space Ω. We note that this assumption is used for ease of exposition, and our results can be modified by incorporating an additional constant term when this assumption does not hold. The proof sketch in this case can be found in Appendix C.

Assumption 2. *The base function ϕ in the Critic satisfies the following two conditions: (1) $\|\phi(s,a)\|_2 \leq 1$, $\forall\ (s,a) \in \mathcal{S}\times\mathcal{A}$; and (2) the smallest singular value for $\mathbf{E}_\rho[\phi(s,a)\phi(s,a)^\top]$ is lower bounded by a positive constant σ^* for any stationary state-action transition distribution ρ.*

Assumption 2 is widely used in the A-C method to guarantee that the minimization in Eqn. (8) can be attained by a unique minimizer (Fu et al., 2020; Bhandari et al., 2018).

Assumption 3. *The reward $r(s,a)$ satisfies the following two conditions: (1) The reward is upper bounded by a positive constant $r_{\max}$ for all $(s,a) \in \mathcal{S}\times\mathcal{A}$; and (2) the stationary state-action transition matrix $\boldsymbol{P}^\pi$ has non-zero spectral gap $1-\lambda_\pi > 0$ for all π.*

The first condition in Assumption 3 is often used for discounted MDPs to ensure a finite value function (Thrun & Schwartz, 1993; Fujimoto et al., 2018; Fu et al., 2020). Moreover, since the samples in the same trajectories are generally correlated, the second condition is adopted to guarantee the concentration properties of the Markov chain, which is generally true for the stationary Markov chain (see, e.g., (Jiang et al., 2018; Ortner, 2020; Amit et al., 2020)).

For any $\lambda \in [0,1)$, let $\alpha_1(\lambda) = (1+\lambda)/(1-\lambda)$, $\alpha_2(\lambda) = 5/(1-\lambda)$ where $\alpha_2(0) = 1/3$. Define

$$\tilde{r}_m = \frac{\sqrt{\alpha_2^2 r_{\max}^2 \left(\max\{\lambda_{\pi_t},0\}\right)^2 \ln^2 p - 2m\alpha_1\left(\max\{\lambda_{\pi_t},0\}\right)\ln p} - \alpha_2\left(\max\{\lambda_{\pi_t},0\}\right)\ln p}{m} + r_{\max}.$$

Then we can have the following main result on the approximation error in the Critic update step.

Theorem 1 (Approximation Error in Critic Update)**.** *Under Assumptions 1, 2, 3, the following inequality holds with probability at least $1-p$, for any $t>0$:*

$$\mathbf{E}[|Q_{\omega_t} - Q_{\tilde{\omega}_t}|] \leq \frac{4((1-\gamma)\tilde{r}_m + \gamma^m R)}{\sqrt{N}(\sigma^*)^2}\left(-\frac{2}{3N}\log\frac{p}{4d} + \sqrt{\frac{4}{9N^2}\log^2\frac{p}{4d} - \frac{2}{N}\log\frac{p}{4d}}\right),$$

where d is the dimension of the Critic parameter ω and R is the radius of Critic parameter space Ω as defined in Eqn. (10).

Theorem 1 establishes the upper bound for the approximation error in the Critic update, which encapsulates the impact of the finite sample size and the finite-step rollout with Bellman evaluation operator T^π. It can be seen from Theorem 1 that in order to obtain an accurate evaluation of the policy, we can either increase the sample size N in the update Eqn. (10) or have more steps of rollout with Bellman evaluation operator T^π. We remark that Theorem 1 considers the correlation across samples, and we appeal to the recent advances in Bernstein's Inequality for Markovian samples (Jiang et al., 2018)(Fan et al., 2021b) to tackle this challenge. The proof of Theorem 1 can be found in Appendix B and Appendix C.

3.2 Approximation Error in the Actor Update

In practice, the greedy search step for solving Eqn. (9) is generally approximated by multiple (e.g., N_a) steps of policy gradient. Based on the policy gradient theorem (Silver et al., 2014; Sutton et al., 1999), we can have the following update at gradient step $k \in [1, N_a]$ in the t-th Actor update:

$$\theta_{t,1} = \theta_t, \quad \theta_{t,N_a} = \theta_{t+1},$$
$$\theta_{t,k+1} = \theta_{t,k} + \alpha \mathbf{E}_{(s,a)\sim\rho^{\pi_{\theta_{t,k}}}}[Q_{\omega_{t+1}}(s,a)\nabla_\theta \pi_{\theta_{t,k}}(a|s)], \tag{11}$$

where α is the learning rate. For simplicity, we drop the subscript t in $\theta_{t,k}$ when no confusion will arise and denote $\rho^k := \rho^{\pi_{\theta_k}}$. As in the Critic update, we sample a trajectory with length l by following the current policy π_{θ_k}, i.e., $\{s_1, a_1, s_2, a_2, \cdots, s_l, a_l\}$, to approximate the expectation in Eqn. (11). Then we can have that

$$\theta_{k+1} = \theta_k + \alpha\tfrac{1}{l}\textstyle\sum_{i=1}^{l}[Q_{\omega_{t+1}}(s_i,a_i)\nabla_\theta \pi_{\theta_k}(a_i|s_i)] := \theta_k + \alpha(C_{k,t,1} + C_{k,t,2}) + \alpha f_{k,t}, \tag{12}$$

where $C_{k,t,1}$, $C_{k,t,2}$ and $f_{k,t}$ are defined as follows

$$C_{k,t,1} := 1/l \sum_{i=1}^{l} (Q_{\omega_{t+1}}(s_i, a_i) \nabla_\theta \pi_{\theta_k}(a_i|s_i) - Q_{\widetilde{\omega}_{t+1}}(s_i, a_i) \nabla_\theta \pi_{\theta_k}(a_i|s_i)),$$
$$C_{k,t,2} := 1/l \sum_{i=1}^{l} (Q_{\widetilde{\omega}_{t+1}}(s_i, a_i) \nabla_\theta \pi_{\theta_k}(a_i|s_i) - Q^{\pi_{\theta_t}}(s_i, a_i) \nabla_\theta \pi_{\theta_k}(a_i|s_i)),$$
$$f_{k,t} := 1/l \sum_{i=1}^{l} Q^{\pi_{\theta_t}}(s_i, a_i) \nabla_\theta \pi_{\theta_k}(a_i|s_i).$$

Here $C_{k,t,1}$ captures the error resulted from using samples to estimate expectation in the Critic update. Based on our result in Theorem 1, with high probability, this term will go to 0 when we have infinite samples or infinite rollout length m. Note that $(T^{\pi_{\theta_t}})^m Q_{\omega_t} = Q_{\widetilde{\omega}_{t+1}}$ (Critic update) and $\lim_{m\to\infty} (T^{\pi_{\theta_t}})^m Q_{\omega_t} = Q^{\pi_{\theta_t}}$. And $C_{k,t,2}$ implies the approximation error when applying the Bellman operator limited (m) times. This term will go to 0 when $m \to \infty$. $f_{k,t}$ is an unbiased estimation of the gradient of $\mathbf{E}_{(s,a)\sim\rho^k}[Q^{\pi_{\theta_t}}(s,a)]$, i.e., $\mathbf{E}[f_{k,t}] = \mathbf{E}_{(s,a)\sim\rho^k}[Q^{\pi_{\theta_t}}(s,a) \nabla_\theta \pi_{\theta_k}(a|s)]$.

Based on Eqn. (12), it is clear that the Actor update with the approximation error resulted from the Critic update can be viewed as a stochastic gradient update with some perturbation $C_{k,t} = C_{k,t,1} + C_{k,t,2}$. Denote θ^*_{t+1} as the solution of the Eqn. (9). For ease of exposition, we use $h(\omega, \theta)$ to denote the objective function in the Actor update:

$$h(\omega, \theta) = \mathbf{E}_{(s,a)\sim\rho^{\pi_\theta}} [Q_\omega(s,a)] = \mathbf{E}_{s\sim d^{\pi_\theta}} [v^{\pi_\omega}(s)]. \tag{13}$$

Note that h is a function of Actor parameter θ for a given Critic parameter ω. Next, we quantify the approximation error in the Actor update in terms of the gap $h(\omega_{t+1}, \theta^*_{t+1}) - h(\omega_{t+1}, \theta_{t+1})$.

Recall that Theorem 1 gives upper bound for the approximation error $\mathbf{E}_\omega[|Q_{\omega_t} - Q_{\widetilde{\omega}_t}|]$ in the Critic update, which has direct impact on $C_{k,t}$. Based on Theorem 1, we have the following two lemmas for the upper bounds on the bias term $b = \mathbf{E}[C_{k,t}]$ and the error term $\beta = f_{k,t} + C_{k,t} - \mathbf{E}[C_{k,t}]$ in the stochastic gradient update Eqn. (12), respectively. The proof of Lemmas 1 and 2 can be found in Appendix D and E, respectively.

Lemma 1 (σ^2-bounded noise)**.** *Suppose Assumptions 1, 2, 3 hold. Then there exist a constant $\sigma^2 \geq 0$ such that with probability at least $1-p$, $\mathbf{E}[\|\beta\|^2] \leq \|\nabla_\theta h(\omega,\theta) + b\|^2 + \sigma^2$, $\forall \theta$, where σ^2 depend on p.*

Lemma 2 (ζ^2-bounded bias)**.** *Suppose Assumptions 1, 2, 3 hold. Then there exists a constant $\zeta \geq 0$ such that with probability at least $1-p$, $\|b\|^2 \leq \zeta^2$, $\forall \theta$, where ζ^2 depends on p.*

Denote the score function $\psi_\theta(a|s) := \nabla_\theta \pi_\theta(a|s)$ and we impose the following standard assumptions.

Assumption 4. *For any $\theta, \theta' \in \mathbb{R}^d$ and any state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, there exist positive constants L_ψ, C_ψ and C_π such that the following holds: (1) $\|\psi_\theta - \psi_{\theta'}\| \leq L_\psi \|\theta - \theta'\|$; (2) $\|\psi_\theta\| \leq C_\psi$ and (3) $\|\pi_\theta(\cdot|s) - \pi_{\theta'}(\cdot|s)\|_{TV} \leq C_\pi \|\theta - \theta'\|$, where $\|\cdot\|_{TV}$ is the total-variation distance.*

We remark that the smoothness and bounded property of the score function as stated in the (1) and (2) in Assumption 4 are widely adopted in the literature (Xu et al., 2020a;b; Zou et al., 2019; Agarwal et al., 2020; Kumar et al., 2019), and it has been shown (Xu et al., 2020a) that (3) in Assumption 4 can be satisfied for any smooth policy with bounded action space.

For the sake of tractability, we next give the following two lemmas about the smoothness and Polyak-Lojasiewicz Condition on the objective function $h(\cdot, \theta)$. The proof can be found in Appendix F.

Lemma 3 (L-smoothness)**.** *Suppose Assumption 4 hold. Then function $h(\cdot,\theta)$ is bounded from below by an infimum $h^{\inf} \in \mathbb{R}$, differentiable and ∇h is L-Lipschitz, i.e., $\|\nabla h(\omega,\theta) - \nabla h(\omega,\theta')\| \leq L\|\theta - \theta'\|$, $\forall\, \omega, \theta, \theta'$.*

Lemma 4 (μ-PL)**.** *If $\nabla h(\omega,\theta) \neq 0$, then we have $\frac{1}{2}\|\nabla h(\omega,\theta)\| \geq \mu(h(\omega,\theta^*) - h(\omega,\theta)) \geq 0, \forall\, \theta, \omega$.*

Let $\alpha \leq \frac{1}{2L}$. Next we present the upper bound of the approximation error in the Actor update.

Theorem 2 (Approximation Error in Actor Update)**.** *Given the updated Actor parameter θ_{t-1}, the following inequality holds with probability at least $1-p$:*

$$\mathbf{E}_\theta[\|h(\omega,\theta^*_t) - h(\omega,\theta_t)\|] \leq (1-\alpha\mu)^{N_a}(h(\omega,\theta^*_t) - h(\omega,\theta_{t-1})) + \frac{\zeta^2 + 2\alpha L \sigma^2}{2\mu}.$$

Theorem 2 reveals that due to the bias and noise induced by the Critic approximation error, running more gradient iterations do not necessarily guarantee the convergence to the optimal policy $\pi_{\theta^*_t}$. Note that Lemmas 1 and 2 in (Ajalloeian & Stich, 2020) are given in the form of assumptions, whereas in this work, we justify that both assumptions hold with high probability and prove Theorem 2, and the proof can be found in Appendix G.

4 IMPACT OF APPROXIMATION ERRORS ON FINITE-TIME LEARNING PERFORMANCE

We next quantify the impact of the approximations errors on the sub-optimality gap of the Warm-Start A-C method with inaccurate Actor/Critic updates. In Section 4.1 we first cast the A-C method as Newton's Method with perturbation, and then present both the finite-time upper bound and lower bound on the finite-time learning performance in Section 4.2.

4.1 ACTOR-CRITIC METHOD AS NEWTON'S METHOD WITH PERTURBATION

As mentioned earlier, the Critic update follows Eqn. (10) with finite samples and finite step rollout with Bellman evaluation operator T^{π} and the Actor update follows Eqn. (12). Given the policy π_t at time t, we denote the resulting policy of one A-C update as $\hat{\pi}_{t+1}$. Recall that we use $\widetilde{\pi}_{t+1}$ to denote the policy attained the $\max$ in $T(\boldsymbol{v}^{\pi_t})$ as illustrated in Figure 1. Furthermore, we define the following notations for ease of our discussion: (1) Denote $\mathcal{E}_{v,t} = \boldsymbol{v}^{\hat{\pi}_{t+1}} - \boldsymbol{v}^{\widetilde{\pi}_{t+1}}$ as the approximation error in the Actor update; (2) Denote $\mathcal{E}_{r,t} = \boldsymbol{r}_{\widetilde{\pi}_{t+1}} - \boldsymbol{r}_{\hat{\pi}_{t+1}}$ as the error in the reward vector, which is induced by the approximation error in the Actor update step; (3) Denote $\mathcal{E}_{P,t} = \boldsymbol{P}_{\widetilde{\pi}_{t+1}} - \boldsymbol{P}_{\hat{\pi}_{t+1}}$ as the error in the transition matrix $\boldsymbol{P}$; (4) Denote $\mathcal{E}_{\hat{J},t} = \boldsymbol{J}_{\widetilde{\boldsymbol{v}}_t}^{-1} - \boldsymbol{J}_{\hat{\boldsymbol{v}}_t}^{-1}$ where $\boldsymbol{J}_{\hat{\boldsymbol{v}}_t} = \boldsymbol{I} - \gamma \boldsymbol{P}_{\hat{\pi}_{t+1}}$ and $\boldsymbol{J}_{\widetilde{\boldsymbol{v}}_t} = \boldsymbol{I} - \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}}$.

Following the same line as in Section 2.2, we treat the A-C algorithm as Newton's method with perturbation $\mathcal{E}_t$, i.e.,

$$\boldsymbol{v}^{\hat{\pi}_{t+1}} = \boldsymbol{v}^{\hat{\pi}_t} - (\boldsymbol{J}_{\hat{\boldsymbol{v}}_t}^{-1}(\boldsymbol{v}^{\hat{\pi}_t} - T(\boldsymbol{v}^{\hat{\pi}_t})) - \mathcal{E}_t) := \boldsymbol{v}^{\hat{\pi}_t} - \hat{\mathcal{L}}(t), \tag{14}$$

where $\hat{\mathcal{L}}(t)$ is the stochastic estimator of Newton's update $\mathcal{L}(t) = \boldsymbol{J}_{\hat{\boldsymbol{v}}^{\pi_t}}^{-1}\left(\boldsymbol{v}^{\hat{\pi}_t} - T\left(\boldsymbol{v}^{\hat{\pi}_t}\right)\right)$, and

$$\mathcal{E}_t = \mathcal{E}_{v,t} + \mathcal{E}_{\hat{J},t}(\boldsymbol{v}^{\hat{\pi}_{t+1}} - (\boldsymbol{r}_{\widetilde{\pi}_{t+1}} + \gamma \boldsymbol{P}_{\widetilde{\pi}_{t+1}} \boldsymbol{v}^{\hat{\pi}_{t+1}})) - \boldsymbol{J}_{\hat{\boldsymbol{v}}_t}^{-1}(\mathcal{E}_{r,t} + \gamma \mathcal{E}_{P,t} \boldsymbol{v}^{\hat{\pi}_t}),$$

which can be further decomposed into bias and Martingale difference noise as follows:

$$\mathcal{B}(t) \triangleq \mathbf{E}[\hat{\mathcal{L}}(t)] - \mathcal{L}(t) = \mathbf{E}[\mathcal{E}_t],$$
$$\mathcal{N}(t) \triangleq \hat{\mathcal{L}}(t) - \mathbf{E}[\hat{\mathcal{L}}(t)] = \mathcal{E}_t - \mathbf{E}[\mathcal{E}_t].$$

We have a few observations in order. It can be seen that the perturbation $\mathcal{E}_t$ results from both Actor approximation error (e.g., $\mathcal{E}_{r,t}$, $\mathcal{E}_{P,t}$) and Critic approximation error (e.g., $\mathcal{E}_{v,t}$). More importantly, the learnt Q function in the Critic update Eqn. (10) is biased in general due to finite rollout steps m, which further leads to the biased gradients in the Actor update Eqn. (11) (Kumar et al., 2019). Clearly, the estimation bias plays an important role in affecting the learning performance, especially when deep neural networks are used as function approximations, which has been extensively investigated using empirical studies (Fujimoto et al., 2018; Elfwing et al., 2018; Van Hasselt et al., 2016).

Next, we examine the bias $\mathcal{B}(t)$ based on the approximation errors in the Actor/Critic updates. Combining the results in Theorem 1 and 2 on the approximation error in the Critic/Actor updates, we have the following result on the bias $\mathcal{B}(t)$. A full derivation is given in Appendix H.

Theorem 3 (Upper Bound on the Bias)**.** *Suppose Assumption 4 holds. Let $S_\epsilon(\cdot)$ be an open ball of radius ϵ. There exist positive constants L_b, and ϵ, such that when $\theta_{t+1} \in S_\epsilon(\theta^*_{t+1})$, the following holds for any $t > 0$,*

$$\|\mathcal{B}(t)\| \leq L_b \mathbf{E}_\theta[\|h(\cdot, \theta^*_{t+1}) - h(\cdot, \theta_{t+1})\|].$$

4.2 FINITE-TIME LEARNING PERFORMANCE AND ERROR PROPAGATION EFFECT

Lower Bound on Performance Gap. We first derive a lower bound to quantify the impact of the bias and the error propagation. By unrolling the recursion of the Newton update (with perturbation) Eqn. (14), we obtain the following theorem.

Theorem 4. *The lower bound of $\|\mathbf{E}[\boldsymbol{v}^* - \boldsymbol{v}^{\hat{\pi}_{t+1}}]\|$ satisfies that*

$$\|\mathbf{E}\left[\boldsymbol{v}^* - \boldsymbol{v}^{\hat{\pi}_{t+1}}\right]\| \geq \|\gamma^{t+1} \bar{\boldsymbol{P}}_{t+1}(\boldsymbol{v}^* - \boldsymbol{v}^{\pi_0}) + \textstyle\sum_{i=1}^{t} \gamma^i \bar{\boldsymbol{P}}_i \mathcal{B}(t-i) + \mathcal{B}(t)\|, \tag{15}$$

where $\bar{\boldsymbol{P}}_{t+1} = \mathbf{E}\left[\left(\prod_{i=0}^{t} \boldsymbol{P}_{\pi_{t+1-i}}\right)\right]$ and π_0 is the warm-start policy.

It can be seen form Theorem 4 that the bias terms $\{\mathcal{B}(t)\}$ add up over time, and the propagation effect of the bias terms is encapsulated by the last two terms on the right side of Eqn. (15). Clearly, the first

term on the right side, corresponding to the impact of the warm-start policy π_0, diminishes with A-C updates. To get a more concrete sense of Theorem 4, we consider the following special settings. (1) When the bias is always positive, i.e., $\mathcal{B}(t) > \mathbf{0}$ for all $t \geq 0$, the lower bound in Theorem 4 is always positive, i.e., $\|\mathbf{E}\left[\boldsymbol{v}^* - \boldsymbol{v}^{\hat{\pi}_{t+1}}\right]\| \geq \|\mathcal{B}(t)\| > 0$. In this case, the sub-optimal gap remains bounded away from zero. Similar conclusion can be made when the bias is always negative. (2) When the bias term can be either positive or negative, the lower bound is shown as Eqn. (15). In this case, the learning performance of the A-C algorithm largely depends on the behavior of the Bias term. It can be seen from Theorem 4 that even when the warm-start policy is near-optimal, it is still challenging to guarantee that online fine-tuning can improve the policy if the approximation error is not handled correctly. We note that this has also been observed empirically (Nair et al., 2020; Lee et al., 2022).

Upper Bound on Performance Gap. In order to derive the upper bound, we first introduce the following standard assumption on the Jacobian $\boldsymbol{J}_{\boldsymbol{v}}$ as in the analysis of policy iteration (Puterman & Brumelle, 1979; Grand-Clément, 2021).

Assumption 5. *For some $q > 0$ there exists a constant $0 < L_J < +\infty$ such that*

$$\|\boldsymbol{J}_{\boldsymbol{v}^\pi} - \boldsymbol{J}_{\boldsymbol{v}^*}\| \leq L_J \|\boldsymbol{v}^\pi - \boldsymbol{v}^*\|^q \ \ \forall\, \pi,$$

and there is a constant $0 < M < +\infty$ such that $\|\boldsymbol{J}_{\boldsymbol{v}^\pi}^{-1}\| \leq M, \ \ \forall\, \pi$.

Denote $H_t := \|\boldsymbol{J}_{\hat{\boldsymbol{v}}_t}^{-1}\left[\boldsymbol{J}_{\hat{\boldsymbol{v}}_t} - \boldsymbol{J}_{\boldsymbol{v}^*}\right]\|$. It is easy to see from Assumption 5 that the term H_t is upper bounded by $H_t \leq M L_J \|\boldsymbol{v}^{\hat{\pi}_t} - \boldsymbol{v}^*\|^q$. Then we have the finite-time upper bound on the sub-optimality gap in Theorem 5.

Theorem 5. *Suppose Assumption 5 holds. Then we have that for any $t > 0$,*

$$\|\mathbf{E}[\boldsymbol{v}^{\hat{\pi}_{t+1}} - \boldsymbol{v}^*]\| \leq \left(\prod_{i=0}^{t} H_{t-i}\right) \|\boldsymbol{v}^* - \boldsymbol{v}^{\pi_0}\| + \sum_{i=1}^{t} \left(\prod_{j=1}^{i} H_{t-j}\right) \|\mathcal{B}(t-i)\| + \|\mathcal{B}(t)\|.$$

The upper bound in Theorem 5 sheds light on the impact of warm-start policy π_0 (the first term) and the bias $\{\mathcal{B}(t)\}$ (the last two terms), thereby providing guidance on how to achieve desired finite-time learning performance. Specifically, consider the case when the approximation error is unbiased. Clearly, we have $\|\mathbf{E}[\boldsymbol{v}^{\hat{\pi}_{t+1}} - \boldsymbol{v}^*]\| \leq \left(\prod_{i=0}^{t} H_{t-i}\right) \|\boldsymbol{v}^* - \boldsymbol{v}^{\pi_0}\|$, which decreases quickly if the warm-start policy π_0 is close to the optimal policy π^*. This observation corroborates the most recent empirically finding, where the online RL is able to further improve the warm-start policy by few adaptation steps (Silver et al., 2017; Bertsekas, 2022a; Kalashnikov et al., 2018). More generally, when the bias $\mathcal{B}(t) \neq \mathbf{0}$, the upper bound hinges heavily on the biases in the approximation errors, even when the warm-start policy π_0 is close to the optimal policy. In this case, recall the result on the upper bound of the bias $\mathcal{B}(t)$ in Theorem 3, where we establish the connection between the bias and the approximation error. As expected, in order to reduce the performance gap, it is essential to decrease the bias in the approximation error, which can be achieved by increasing gradient steps and sample sizes. The proof of Theorem 4 and 5 are relegated to the Appendix I and J, respectively. The experiments results and analysis on the Gridworld benchmark can be found in Appendix K.

5 Conclusion

In this work, we take a finite-time analysis approach to quantify the impact of approximation errors on the learning performance of Warm-Start A-C method with a given prior policy. By delving into the intricate coupling between the updates of the Actor and the Critic, we first provide upper bounds on the approximation errors in both the Critic update and Actor update of online adaptation, respectively, where the recent advances on Bernstein's Inequality are leveraged to deal with the sample correlation therein. Based on these results, we next cast the Warm-Start A-C method as Newton's method with perturbation, which serves as the foundation for characterizing the impact of the approximation errors on the finite-time learning performance of Warm-Start A-C. In particular, we derive lower bounds on the sub-optimality gap under biased approximation errors, indicating that the performance gap can be bounded away from zero even for Warm-Start A-C with a good prior policy. And we also provide upper bounds on the sub-optimality gap, which provides guidance on the design of Warm-Start RL for achieving desired finite-time learning performance.

REFERENCES

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In Conference on Learning Theory, pp. 64–66. PMLR, 2020.

Ahmad Ajalloeian and Sebastian U Stich. On the convergence of SGD with biased gradients. arXiv preprint arXiv:2008.00051, 2020.

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In International conference on machine learning, pp. 269–278. PMLR, 2020.

Dimitri Bertsekas. Abstract Dynamic Programming. Athena Scientific, 2022a.

Dimitri Bertsekas. Lessons from Alphazero for Optimal, Model Predictive, and Adaptive Control. Athena Scientific, 2022b.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In Conference on learning theory, pp. 1691–1692. PMLR, 2018.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. Convex Optimization. Cambridge university press, 2004.

James J Choi, David Laibson, Brigitte C Madrian, and Andrew Metrick. Reinforcement learning and savings behavior. The Journal of finance, 64(6):2515–2534, 2009.

Gal Dalal, Gugan Thoppe, Balázs Szörényi, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In Conference On Learning Theory, pp. 1199–1233. PMLR, 2018.

Gal Dalal, Balazs Szorenyi, and Gugan Thoppe. A tale of two-timescale reinforcement learning with the tightest finite-time bound. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pp. 3701–3708, 2020.

Thinh Doan, Siva Maguluri, and Justin Romberg. Finite-time analysis of distributed TD (0) with linear function approximation on multi-agent reinforcement learning. In International Conference on Machine Learning, pp. 1626–1635. PMLR, 2019.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks, 107:3–11, 2018.

Jianqing Fan, Bai Jiang, and Qiang Sun. Hoeffding's inequality for general markov chains and its applications to statistical learning. Journal of Machine Learning Research, 22(139):1–35, 2021a. URL http://jmlr.org/papers/v22/19-479.html.

Jianqing Fan, Bai Jiang, and Qiang Sun. Hoeffding's inequality for general markov chains and its applications to statistical learning. Journal of Machine Learning Research, 22(139):1–35, 2021b.

Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Single-timescale actor-critic provably finds globally optimal policy. In International Conference on Learning Representations, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In International conference on machine learning, pp. 1587–1596. PMLR, 2018.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. Journal of Machine Learning Research, 16(1):1437–1480, 2015.

Matilde Gargiani, Andrea Zanelli, Dominic Liao-McPherson, TH Summers, and John Lygeros. Dynamic programming through the lens of semismooth newton-type methods. IEEE Control Systems Letters, 2022.

Sylvain Gelly and David Silver. Combining online and offline knowledge in UCT. In Proceedings of the 24th international conference on Machine learning, pp. 273–280, 2007.

Julien Grand-Clément. From convex optimization to MDPs: A review of first-order, second-order and quasi-newton methods for MDPs. arXiv preprint arXiv:2104.10677, 2021.

Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(6):1291–1307, 2012.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In Conference on Robot Learning, pp. 1025–1037. PMLR, 2020.

Auke Ijspeert, Jun Nakanishi, and Stefan Schaal. Learning attractor landscapes for learning motor primitives. Advances in neural information processing systems, 15, 2002.

Bai Jiang, Qiang Sun, and Jianqing Fan. Bernstein's inequality for general markov chains. arXiv preprint arXiv:1805.10721, 2018.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. Journal of artificial intelligence research, 4:237–285, 1996.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In Conference on Robot Learning, pp. 651–673. PMLR, 2018.

Beomjoon Kim, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Learning from limited demonstrations. Advances in Neural Information Processing Systems, 26, 2013.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. Advances in neural information processing systems, 12, 1999.

Aviral Kumar, Abhishek Gupta, and Sergey Levine. Discor: Corrective feedback in reinforcement learning via distribution correction. Advances in Neural Information Processing Systems, 33: 18560–18572, 2020.

Harshat Kumar, Alec Koppel, and Alejandro Ribeiro. On the sample complexity of actor-critic method for reinforcement learning with function approximation. arXiv preprint arXiv:1910.08412, 2019.

Seunghyun Lee, Younggyo Seo, Kimin Lee, Pieter Abbeel, and Jinwoo Shin. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In Conference on Robot Learning, pp. 1702–1712. PMLR, 2022.

David A Levin and Yuval Peres. Markov Chains and Mixing Times, volume 107. American Mathematical Soc., 2017.

Pascal Lezaud. Chernoff-type bound for finite markov chains. Annals of Applied Probability, pp. 849–867, 1998.

Yuxi Li. Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274, 2017.

Ashvin Nair, Murtaza Dalal, Abhishek Gupta, and Sergey Levine. Accelerating online reinforcement learning with offline datasets. arXiv preprint arXiv:2006.09359, 2020.

Yurii Nesterov. Introductory Lectures on Convex Optimization: A Basic Course, volume 87. Springer Science & Business Media, 2003.

Ronald Ortner. Regret bounds for reinforcement learning via markov chain concentration. Journal of Artificial Intelligence Research, 67:115–128, 2020.

Jan Peters and Stefan Schaal. Natural actor-critic. Neurocomputing, 71(7-9):1180–1190, 2008.

Martin L Puterman. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, 2014.

Martin L Puterman and Shelby L Brumelle. On the convergence of policy iteration in stationary dynamic programming. Mathematics of Operations Research, 4(1):60–69, 1979.

Manuel S Santos and John Rust. Convergence properties of policy iteration. SIAM Journal on Control and Optimization, 42(6):2094–2115, 2004.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295, 2016.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In International conference on machine learning, pp. 387–395. PMLR, 2014.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815, 2017.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science, 362(6419): 1140–1144, 2018.

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

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 12, 1999.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum, volume 6, 1993.

Ikechukwu Uchendu, Ted Xiao, Yao Lu, Banghua Zhu, Mengyuan Yan, Joséphine Simon, Matthew Bennice, Chuyuan Fu, Cong Ma, Jiantao Jiao, et al. Jump-start reinforcement learning. arXiv preprint arXiv:2204.02372, 2022.

Masatoshi Uehara, Masaaki Imaizumi, Nan Jiang, Nathan Kallus, Wen Sun, and Tengyang Xie. Finite sample analysis of minimax offline reinforcement learning: Completeness, fast rates and first-order efficiency. arXiv preprint arXiv:2102.02981, 2021.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In Proceedings of the AAAI conference on artificial intelligence, volume 30, 2016.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=BJgQfkSYDS.

Cathy Wu, Aboudy Kreidieh, Kanaad Parvate, Eugene Vinitsky, and Alexandre M Bayen. Flow: Architecture and benchmarking for reinforcement learning in traffic control. arXiv preprint arXiv:1710.05465, 10, 2017.

Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. Advances in neural information processing systems, 34, 2021.

Tengyu Xu, Zhe Wang, and Yingbin Liang. Improving sample complexity bounds for actor-critic algorithms. arXiv preprint arXiv:2004.12956, 2020a.

Tengyu Xu, Zhe Wang, and Yingbin Liang. Non-asymptotic convergence analysis of two time-scale (natural) actor-critic algorithms. arXiv preprint arXiv:2005.03557, 2020b.

Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. Advances in neural information processing systems, 32, 2019.

Shaofeng Zou, Tengyu Xu, and Yingbin Liang. Finite-sample analysis for SARSA with linear function approximation. Advances in neural information processing systems, 32, 2019.