Non-Asymptotic Analysis for Two Time-scale TDC with General Smooth Function Approximation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Temporal-difference learning with gradient correction (TDC) is a two time-scale algorithm for policy evaluation in reinforcement learning. This algorithm was initially proposed with linear function approximation, and was later extended to the one with general smooth function approximation. The asymptotic convergence for the on-policy setting with general smooth function approximation was established in [5], however, the non-asymptotic convergence analysis remains unsolved due to challenges in the non-linear and two-time-scale update structure, non-convex objective function and the projection onto a time-varying tangent plane. In this paper, we develop novel techniques to address the above challenges and explicitly characterize the non-asymptotic error bound for the general off-policy setting with i.i.d. or Markovian samples, and show that it converges as fast as $\mathcal{O}(1/\sqrt{T})$ (up to a factor of $\mathcal{O}(\log T)$). Our approach can be applied to a wide range of value-based reinforcement learning algorithms with general smooth function approximation.

1 Introduction

In reinforcement learning (RL), an agent interacts with a stochastic environment in order to maximize the total reward [30]. Towards this goal, it is often needed to evaluate how good a policy performs, and more specifically, to learn its value function. Temporal difference (TD) learning algorithm is one of the most popular policy evaluation approaches. However, when applied with function approximation approach and/or under the off-policy setting, the TD learning algorithm may diverge [2, 33]. To address this issue, a family of gradient-based TD (GTD) algorithms, e.g., GTD, GTD2, temporal-difference learning with gradient correction (TDC) and Greedy-GQ, were developed for the case with linear function approximation [21, 31, 22, 32, 31]. These algorithms were later extended to the case with general smooth function approximation in [5], where asymptotic convergence guarantee was established for the on-policy setting with i.i.d. samples.

Despite the success of the GTD methods in practice, previous theoretical studies only showed that these algorithms converge asymptotically, and did not suggest how fast these algorithms converge and how the accuracy of the solution depends on various parameters of the algorithms. Not until recently have the non-asymptotic error bounds for these algorithms been investigated, e.g., [8, 15, 35, 38, 14, 9, 34], which mainly focus on the case with linear function approximation. These results thus cannot be directly applied to more practical applications with general smooth function approximation, e.g., neural networks, which have greater representation power, do not need to construct feature mapping, and are widely used in practice.

In this paper, we develop a non-asymptotic analysis for the TDC algorithm with general smooth function approximation (which we refer to as non-linear TDC) for both i.i.d. and Markovian samples. Technically, the analysis in this paper is not a straightforward extension of previous studies on those GTD algorithms with linear function approximation. First of all, different from existing studies with

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

linear function approximation whose objective functions are *convex* and the updates are *linear*, the objective function of the non-linear TDC algorithm is *non-convex*, and the two time-scale updates are *non-linear* functions of the parameters. Second, the objective function of the non-linear TDC algorithm, the mean-square projected Bellman error (MSPBE), involves a projection onto a *time-varying* tangent plane which depends on the sample trajectory, whereas for GTD algorithms with linear function approximation, this projection is time-invariant. Third, due to the two time-scale structure of the algorithm and the Markovian noise, novel techniques to deal with the stochastic bias and the tracking error need to be developed.

1.1 Challenges and Contributions

In this section, we summarize the technical challenges and our contributions.

Analysis for two time-scale non-linear updates and non-convex objective. Unlike many existing results on two time-scale stochastic approximation, e.g., [16, 13, 14] and the studies of linear GTD algorithms in [38, 19, 34, 8], the objective function of the non-linear TDC is non-convex, and its two time-scale updates are non-linear. Therefore, existing studies on linear two time-scale algorithms cannot be directly applied. Moreover, the convergence to global optimum cannot be guaranteed for the non-linear TDC algorithm, and therefore, we study the convergence to stationary points. In this paper, we develop a novel non-asymptotic analysis of the non-linear TDC algorithm, which solves RL problems from a non-convex optimization perspective. We note that our analysis is not a straightforward extension of analyses of non-convex optimization, as the update rule here is two time-scale and the noise is Markovian. The framework we develop in this paper can be applied to analyze a wide range of value-based RL algorithms with general smooth function approximation.

Time-varying projection. For the MSPBE, a projection of the Bellman error onto the parameterized function class is involved. However, unlike linear function approximation, the projection onto a general smooth class of functions usually does not have a closed-form solution. Thus, a projection onto the tangent plane at the current parameter is used instead, which incurs a time-varying projection that depends on the current parameter and thus the sample trajectory. This brings in additional challenges in the bias and variance analysis due to such dependency. We develop a novel approach to decouple such a dependency and characterize the bias by exploiting the uniform ergodicity of the underlying MDP and the smoothness of the parameterized function. The new challenges posed by the time-varying projection and the dependence between the projection and the sample trajectory are not special to the non-linear TDC investigated in this paper, and they exist in a wide range of value-based algorithms with general smooth function approximation, where our techniques can be applied.

A tight tracking error analysis. Due to the two time-scale structure of the update rule, the tracking error, which measures how fast the fast time-scale tracks its own limit, needs to be explicitly bounded. Unlike the studies on two time-scale linear stochastic approximation [8, 14, 16], where a linear transformation can asymptotically decouple the dependence between the fast and slow time-scale updates, it is non-trivial to construct such a transformation for non-linear updates. To develop a tight bound on the tracking error, we develop a novel technique that bounds the tracking error as a function of the gradient of the MSPBE. This leads to a bound on the tracking error in terms of the slow time-scale parameter. Although we do not decouple the fast and slow time-scale updates, we still obtain a desired convergence rate of $\mathcal{O}(1/\sqrt{T})$ (up to a factor of $\log T$), which matches with the complexity of stochastic gradient descent for non-convex problems [12].

1.2 Related Work

TD, Q-learning and SARSA. The asymptotic convergence of TD with linear function approximation was shown in [33], and the non-asymptotic analysis of TD was developed in [28, 17, 4, 9, 29]. Moreover, [7] further studied the non-asymptotic error bound of TD learning with neural function approximation. Q-learning and SARSA are usually used for solving the optimal control problem and were shown to converge asymptotically under some conditions in [23, 27]. Their non-asymptotic error bounds were also studied in [39]. The non-asymptotic analysis of Q-learning under the neural function approximation was developed in [7, 36]. Note that all these algorithms are one time-scale, while the TDC algorithm we study is a two time-scale algorithm.

GTD methods with linear function approximation. A class of GTD algorithms were proposed to address the divergence issue for off-policy training [2] and arbitrary smooth function approximation

[33], e.g., GTD, GTD2 and TDC [21, 31, 22, 32, 31]. Recent studies established their non-asymptotic convergence rate, e.g., [9, 34, 18, 13, 38, 8, 14, 19, 35, 20] under i.i.d. and Markovian settings. These studies focus on the case with linear function approximation, and thus the objective functions are convex, and the updates are linear. In this paper, we focus on the non-linear TDC algorithm with general smooth function approximation, where the two time-scale update rule is non-linear, the objective is non-convex, and the projection is time-varying, and thus new techniques are required to develop the non-asymptotic analysis.

Non-linear two time-scale stochastic approximation. There are also studies on asymptotic convergence rate and non-asymptotic analysis for non-linear two time-scale stochastic approximation, e.g., [25, 10]. Although the non-linear update rule is investigated, it is assumed that the algorithm converges to the global optimum. In this paper, we do not make such an assumption on the global convergence, which may not necessarily hold for the non-linear TDC algorithm, and instead, we study the convergence to stationary points, which is a widely used convergence criterion for non-convex optimization problems. We also note that there is a resent work studying the batch-based non-linear TDC in [37], where at each update, a batch of samples is used. To achieve a sample complexity of $\mathcal{O}(\epsilon^{-2})$, a batch size of $\mathcal{O}(\epsilon^{-1})$ is required in [37] to control the bias and variance. We note that by setting the batch size being one in [37], the desired sample complexity cannot be obtained, and their error bound will be a constant. In this paper, we focus on the non-linear TDC algorithm without using the batch method, where the parameters update in an online and incremental fashion and at each update only one sample is used. Our error analysis is novel and more refined as it does not require a large batch size of $\mathcal{O}(\epsilon^{-1})$ while still achieving the same sample complexity.

2 Preliminaries

2.1 Markov Decision Process

A Markov decision process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, \mathsf{P}, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $\mathsf{P} = \mathsf{P}(s'|s,a)$ is the transition kernel, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^+$ is the reward function bounded by $r_{\max}$, and $\gamma \in [0,1]$ is the discount factor. A stationary policy π maps a state $s \in \mathcal{S}$ to a probability distribution $\pi(\cdot|s)$ over the action space $\mathcal{A}$. At each time-step t, suppose the process is at some state $s_t \in \mathcal{S}$, and an action $a_t \in \mathcal{A}$ is taken. Then the system transits to the next state s_{t+1} following the transition kernel $\mathsf{P}(\cdot|s_t, a_t)$, and the agent receives a reward $r(s_t, a_t, s_{t+1})$.

For a given policy π and any initial state $s \in \mathcal{S}$, we define its value function as $V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t, S_{t+1}) | S_0 = s, \pi\right]$. The goal of policy evaluation is to use the samples generated from the MDP to estimate the value function. The value function satisfies the Bellman equation: $V^\pi(s) = T^\pi V^\pi(s)$ for any $s \in \mathcal{S}$, where the Bellman operator T^π is defined as

$$T^\pi V(s) = \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} \mathsf{P}(s'|s,a)\pi(a|s) r(s,a,s') + \gamma \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} \mathsf{P}(s'|s,a)\pi(a|s) V(s'). \tag{1}$$

Hence the value function V^π is the fixed point of the Bellman operator T^π [3].

2.2 Function Approximation

In practice, the state space $\mathcal{S}$ usually contains a large number of states or is even continuous, which will induce a heavy computational overhead. A popular approach is to approximate the value function using a parameterized class of functions. Consider a parameterized family of functions $\left\{V_\theta : \mathcal{S} \to \mathbb{R} | \theta \in \mathbb{R}^N\right\}$, e.g., neural networks. The goal is to find a V_θ with a compact representation in θ to approximate the value function V^π. In this paper, we focus on a general family of smooth functions, which may not be linear in θ.

3 TDC with Non-Linear Function Approximation

In this section, we introduce the TDC algorithm with general smooth function approximation in [5] for the off-policy setting with both i.i.d. samples and Markovian samples, and further characterize the non-asymptotic error bounds.

Consider the the following mean-square projected Bellman error (MSPBE):

$$J(\theta) = \mathbb{E}_{\mu^\pi}\left[\|V_\theta(s) - \mathbf{\Pi}_\theta T^\pi V_\theta(s)\|^2\right], \tag{2}$$

where μ^π is the stationary distribution induced by the policy π, and $\mathbf{\Pi}_\theta$ is the orthogonal projection onto the tangent plane of V_θ at θ: $\left\{\hat{V}_\zeta(s)|\zeta \in \mathbb{R}^N \text{ and } \hat{V}_\zeta(s) = \phi_\theta(s)^\top\zeta\right\}$ and $\phi_\theta(s) = \nabla V_\theta(s)$. Note that the projection is onto the tangent plane instead of $\left\{V_\theta : \theta \in \mathbb{R}^N\right\}$ since the projection onto the latter one may not be computationally tractable if V_θ is non-linear.

In [5], the authors proposed a two time-scale TDC algorithm to minimize the MSPBE $J(\theta)$. Specifically, a stochastic gradient descent approach is used with the weight doubling trick (for the double sampling problem) [32], which yield a two time-scale update rule. We note that the algorithm developed in [5] was for the on-policy setting with i.i.d. samples from the stationary distribution, and the asymptotic convergence of the algorithm to stationary points was established.

In the off-policy setting, the goal is to estimate the value function V^π of the target policy π using the samples from a different behavior policy π_b. In this case, the MSPBE can be written as

$$J(\theta) = \mathbb{E}_{\mu^{\pi_b}}[\|V_\theta(s) - \mathbf{\Pi}_\theta T^\pi V_\theta(s)\|^2], \tag{3}$$

and we use the approach of importance sampling. Following steps similar to those in [21], $J(\theta)$ can be further written as

$$J(\theta) = \mathbb{E}_{\mu^{\pi_b}}[\rho(S,A)\delta_{S,A,S'}(\theta)\phi_\theta(S)]^\top A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}[\rho(S,A)\delta_{S,A,S'}(\theta)\phi_\theta(S)], \tag{4}$$

where $\delta_{s,a,s'}(\theta) = r(s,a,s') + \gamma V_\theta(s') - V_\theta(s)$ is the TD error, $\phi_\theta(s) = \nabla V_\theta(s)$ is the character vector, $\rho(s,a) = \frac{\pi(a|s)}{\pi_b(a|s)}$ is the importance sampling ratio for a given sample $O = (s,a,r,s')$ and $A_\theta = \mathbb{E}_{\mu^{\pi_b}}[\phi_\theta(S)\phi_\theta(S)^\top]$.

To compute $\nabla J(\theta)$, we consider its i-th entry, i.e., the partial derivative w.r.t. the i-th entry of θ:

$$\begin{aligned}
&-\frac{1}{2}\frac{\partial J(\theta)}{\partial \theta^i} \\
&= \underbrace{-\mathbb{E}_{\mu^{\pi_b}}\left[\frac{\partial}{\partial\theta^i}(\rho\delta\phi)\right]^\top A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta\phi\right]}_{(a)} + \frac{1}{2}\underbrace{(A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta\phi\right])^\top\mathbb{E}_{\mu^{\pi_b}}\left[\frac{\partial}{\partial\theta^i}(\phi\phi^\top)\right](A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta\phi\right])}_{(b)},
\end{aligned} \tag{5}$$

where to simplify notations, we omit the dependence on θ, S, A and S'. To get an unbiased estimate of the terms in (5), several independent samples are needed, but this is not applicable when there is only one sample trajectory. Hence we employ the weight doubling trick [32]. Define $\omega(\theta) = A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho(S,A)\delta_{S,A,S'}(\theta)\phi_\theta(S)\right]$, then term (a) can be written as follows:

$$\begin{aligned}
&-\mathbb{E}_{\mu^{\pi_b}}\left[\frac{\partial}{\partial\theta^i}(\rho\delta\phi)\right]^\top A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta\phi\right] \\
&= -\mathbb{E}_{\mu^{\pi_b}}\left[\rho(\gamma(\phi_\theta(S'))_i - (\phi_\theta(S))_i)\phi_\theta(S)\right]^\top\omega(\theta) - \mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta(\nabla^2 V)_i\right]^\top\omega(\theta);
\end{aligned} \tag{6}$$

and term (b) can be written as follows:

$$(A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta\phi\right])^\top\mathbb{E}_{\mu^{\pi_b}}\left[\frac{\partial}{\partial\theta^i}(\phi\phi^\top)\right](A_\theta^{-1}\mathbb{E}_{\mu^{\pi_b}}\left[\rho\delta\phi\right]) = 2\mathbb{E}_{\mu^{\pi_b}}\left[\phi^\top\omega(\theta)(\frac{\partial}{\partial\theta^i}\phi^\top)\omega(\theta)\right]. \tag{7}$$

Hence the gradient can be re-written as

$$-\frac{\nabla J(\theta)}{2} = \mathbb{E}_{\mu^{\pi_b}}\left[\rho(S,A)\delta_{S,A,S'}(\theta)\phi_\theta(S)\right] - h(\theta,\omega(\theta)) - \gamma\mathbb{E}_{\mu^{\pi_b}}\left[\rho(S,A)\phi_\theta(S')\phi_\theta(S)^\top\right]\omega(\theta), \tag{8}$$

where $h(\theta,\omega) = \mathbb{E}_{\mu^{\pi_b}}[\left(\rho(S,A)\delta_{S,A,S'}(\theta) - \phi_\theta(S)^\top\omega\right)\nabla^2 V_\theta(S)\omega]$. Thus with this weight doubling trick [32], a two time-scale stochastic gradient descent algorithm can be constructed. In Algorithm 1, we present the algorithm for the Markovian setting. The algorithm under the i.i.d. setting is slightly different, hence we refer the readers to Algorithm 2 in Appendix B.

Algorithm 1 Non-Linear Off-Policy TDC under the Markovian Setting

Input: T, α, β, π, π_b, $\{V_\theta | \theta \in \mathbb{R}^N\}$
Initialization: θ_0,w_0

Choose $W \sim \text{Uniform}(0, 1, ..., T-1)$
for $t = 0, 1, ..., W-1$ **do**
  Sample $O_t = (s_t, a_t, r_t, s_{t+1})$ following π_b
  $\delta_t(\theta_t) = r(s_t, a_t, s_{t+1}) + \gamma V_{\theta_t}(s_{t+1}) - V_{\theta_t}(s_t)$
  $h_t(\theta_t, \omega_t) = \big(\rho_t \delta_t(\theta_t) - \phi_{\theta_t}(s_t)^\top \omega_t\big) \nabla^2 V_{\theta_t}(s_t)\omega_t$
  $\omega_{t+1} = \mathbf{\Pi}_{R_\omega}\big(\omega_t + \beta\big(-\phi_{\theta_t}(s_t)\phi_{\theta_t}(s_t)^\top \omega_t + \rho_t \delta_t(\theta_t)\phi_{\theta_t}(s_t)\big)\big)$
  $\theta_{t+1} = \theta_t + \alpha\big(\rho_t \delta_t(\theta_t)\phi_{\theta_t}(s_t) - \gamma \rho_t \phi_{\theta_t}(s_{t+1})\phi_{\theta_t}(s_t)^\top \omega_t - h_t(\theta_t, \omega_t)\big)$
end for

Output: θ_W

In Algorithm 1, $\mathbf{\Pi}_{R_\omega}(v) = \arg\min_{\|w\| \le R_\omega} \|v - w\|$ denotes the projection operator, where $R_\omega = \frac{\rho_{\max} C_\phi}{\lambda_v}(r_{\max} + (1+\gamma)C_v)$ (the constants are defined in Section 3.1). As we will show in (44) in the appendix that for any $\theta \in \mathbb{R}^N$, $\omega(\theta)$ is always upper bounded by R_ω, i.e., $\|\omega(\theta)\| \le R_\omega$. The projection step in the algorithm is introduced mainly for the convenience of the analysis. Motivated by the randomized stochastic gradient method in [12], which is designed to analyze non-convex optimization problems, in this paper, we also consider a randomized version of the non-linear TDC algorithm. Specifically, let W be an independent random variable with a uniform distribution over $\{0, 1, ..., T-1\}$. We then run the non-linear TDC algorithm for W steps and output θ_W.

3.1 Non-asymptotic Error Bounds

In this section, we present our main results of the non-asymptotic error bounds on the convergence of the off-policy non-linear TDC algorithm. Our results will be based on the following assumptions.

Assumption 1 (Boundedness and Smoothness)**.** *For any $s \in \mathcal{S}$ and any $\theta, \theta' \in \mathbb{R}^N$,*

$$|V_\theta(s)| \le C_v, \qquad \|\phi_\theta(s)\| \le C_\phi,$$
$$\|\nabla^2 V_\theta(s)\| \le D_v, \qquad \|\nabla^2 V_\theta(s) - \nabla^2 V_{\theta'}(s)\| \le L_V \|\theta - \theta'\|,$$

where C_ϕ, C_v, D_v and L_V are some positive constants.

From Assumption 1, it follows that for any $\theta, \theta' \in \mathbb{R}^N$, $|V_\theta(s) - V_{\theta'}(s)| \le C_\phi \|\theta - \theta'\|$, and $\|\phi_\theta(s) - \phi_{\theta'}(s)\| \le D_v \|\theta - \theta'\|$. We note that these assumptions are equivalent to the assumptions adopted in the original non-linear TDC asymptotic convergence analysis in [5], and can be easily satisfied by appropriately choosing the function class $\{V_\theta : \theta \in \mathbb{R}^N\}$. For example, in neural networks, these assumptions can be satisfied if the activation function is Lipschitz and smooth [11, 26, 24].

Assumption 2 (Non-singularity)**.** *For any $\theta \in \mathbb{R}^N$, $\lambda_L(A_\theta) \ge \lambda_v > 0$, where $\lambda_L(A)$ denotes the minimal eigenvalue of the matrix A and λ_v is a positive constant.*

Assumption 3 (Bounded Importance Sampling Ratio)**.** *For any $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\rho(s, a) = \frac{\pi(a|s)}{\pi_b(a|s)} \le \rho_{\max}$, for some positive constant $\rho_{\max}$.*

The following assumption is only needed for the analysis under the Markovian setting, and is widely used for analyzing the Markovian noise, e.g., [35, 14, 37, 39, 28, 4].

Assumption 4. *(Geometric uniform ergodicity) There exist some constants $m > 0$ and $\kappa \in (0, 1)$ such that $\sup_{s \in \mathcal{S}} d_{TV}(\mathbb{P}(s_t = \cdot | s_0 = s, \pi), \mu^\pi) \le m\kappa^t$, for any $t > 0$, where d_{TV} denotes the total-variation distance between the probability measures.*

We then present the bounds on the convergence of the TDC algorithm with general smooth function approximation in the following theorem.

Theorem 1. *Consider the following step-sizes: $\alpha = \mathcal{O}\left(\frac{1}{T^a}\right)$, and $\beta = \mathcal{O}\left(\frac{1}{T^b}\right)$, where $\frac{1}{2} \le a \le 1$ and $0 < b \le a$. Then, (1) under the i.i.d. setting, $\|\nabla J(\theta_W)\|^2 = \mathcal{O}\left(\frac{1}{T^{1-a}} + \frac{1}{T^b} + \frac{1}{T^{1-b}}\right)$; and (2) under the Markovian setting, $\|\nabla J(\theta_W)\|^2 = \mathcal{O}\left(\frac{\log T}{T^{1-a}} + \frac{1}{T^{1-b}} + \frac{\log T}{T^b}\right)$.*

We only provide the order of the bounds in terms of T here, and the explicit bounds can be found in (86) and (130) in the appendix. It can be seen that the rate under the Markovian setting is slower than the one under the i.i.d. setting by a factor of $\log T$, which is essentially the mixing time introduced by the dependence of samples.

Theorem 1 characterizes the dependence between convergence rate and the step-sizes α and β. We also optimize over the step-sizes in the following corollary.

Corollary 1. *Let $a = b = \frac{1}{2}$, i.e., $\alpha, \beta = \mathcal{O}(1/\sqrt{T})$, then (1) under the i.i.d. setting, $\|\nabla J(\theta_W)\|^2 = \mathcal{O}(1/\sqrt{T})$; and (2) under the Markovian setting, $\|\nabla J(\theta_W)\|^2 = \mathcal{O}(\log T/\sqrt{T})$.*

Remark 1. Our result matches with the sample complexity for the batch-based algorithm in [37]. But their work requires a large batch size of $\mathcal{O}(\epsilon^{-1})$ to control the bias and variance, while ours only needs one sample in each step to update θ and ω and can still obtain the same convergence rate. We note that by setting the batch size being one in [37], their desired sample complexity cannot be obtained, and their error bound will be a *constant*. To obtain our non-asymptotic bound and sample complexity for the non-linear TDC algorithm, we develop a novel and more refined analysis on the tracking error, which will be discussed in the next section. Moreover, our result matches with the convergence rate of solving general non-convex optimization problems using stochastic gradient descent in [12]. Compared to their work, our analysis is more challenging due to the two time-scale structure and the gradient bias from the Markovian noise and the tracking error.

Remark 2. Some analyses on two time-scale stochastic approximation bound the tracking error in terms of $\frac{\alpha}{\beta}$, and require $\frac{\alpha}{\beta} \to 0$ in order to drive the tracking error to zero resulting in a convergence rate of $\mathcal{O}\left(\beta + \frac{\alpha}{\beta}\right)$ [6]. In this paper, we develop a much tighter bound on the tracking error in terms of the slow time-scale parameter $\nabla J(\theta)$. Therefore, the tracking error in our analysis is driven to zero by $\nabla J(\theta) \to 0$ not $\frac{\alpha}{\beta} \to 0$. Similar results that do not need $\frac{\alpha}{\beta} \to 0$ can also be found, e.g., in [16, 14]. We would like to point out that the techniques in [16, 14] cannot be applied in our analysis due to the non-linear two time-scale updates in this paper.

4 Proof Sketch

In this section, we provide an outline of the proof of Theorem 1 under the Markovian setting, and highlight our major technical contributions. For the complete proof of Theorem 1, we refer the readers to Appendices B.2 and C.2.

Let $O_t = (s_t, a_t, r_t, s_{t+1})$ be the sample observed at time t. Denote the tracking error by $z_t = \omega_t - \omega(\theta_t)$, which characterizes the error between the fast time-scale update and its limit if the slow time-scale update θ_t is kept fixed and only the fast time-scale is being updated. Denote by $G_{t+1}(\theta, \omega) \triangleq \rho_t \delta_t(\theta)\phi_\theta(s_t) - \gamma\rho_t\phi_\theta(s_{t+1})\phi_\theta(s_t)^\top\omega_t - h_t(\theta, \omega)$. Denote by τ_β the mixing time of the MDP, i.e., $\tau_\beta \triangleq \min\{t : m\kappa^t \leq \beta\}$.

Step 1. In this step, we decompose the error of gradient norm into two parts: the stochastic bias and the tracking error. We first show in Appendix A that $J(\theta)$ is L_J-smooth: for any $\theta_1, \theta_2 \in \mathbb{R}^N$,

$$\|\nabla J(\theta_1) - \nabla J(\theta_2)\| \leq L_J\|\theta_1 - \theta_2\|. \tag{9}$$

We note that the smoothness of $J(\theta)$ is also used in [37], which, however, is assumed instead of being proved as in this paper. It then follows that

$$\begin{aligned}\frac{\alpha}{2}\|\nabla J(\theta_t)\|^2 \leq & J(\theta_t) - J(\theta_{t+1}) + \underbrace{\alpha\langle\nabla J(\theta_t), -G_{t+1}(\theta_t, \omega(\theta_t)) + G_{t+1}(\theta_t, \omega_t)\rangle}_{(a)} \\ & + \underbrace{\alpha\left\langle\nabla J(\theta_t), \frac{\nabla J(\theta_t)}{2} + G_{t+1}(\theta_t, \omega(\theta_t))\right\rangle}_{(b)} + \frac{L_J}{2}\alpha^2\|G_{t+1}(\theta_t, \omega_t)\|^2. \end{aligned} \tag{10}$$

This implies that the error bound on the gradient norm is controlled by the tracking error (a) which is introduced by the two time-scale update rule, and the stochastic bias (b) which is due to the time-varying projection and the Markovian sampling.

Step 2. We first bound the tracking error. Re-write the update of ω_t in terms of z_t: $z_{t+1} = z_t + \beta\left(-A_{\theta_t}(s_t)z_t + b_t(\theta_t)\right) + \omega(\theta_t) - \omega(\theta_{t+1})$, where $A_{\theta_t}(s_t) = \phi_{\theta_t}(s_t)\phi_{\theta_t}(s_t)^\top$ and $b_t(\theta_t) = -A_{\theta_t}(s_t)\omega(\theta_t) + \rho_t\delta_t(\theta_t)\phi_{\theta_t}(s_t)$. From the Lipschitz continuity of $\omega(\theta)$, it follows that

$$\begin{aligned}\|z_{t+1}\| &\leq (1+\beta C_\phi^2)\|z_t\| + \beta(b_{\max} + L_\omega C_g),\\ \|z_{t+1} - z_t\| &\leq \beta C_\phi^2\|z_t\| + \beta(b_{\max} + L_\omega C_g),\end{aligned} \tag{11}$$

which further implies

$$\begin{aligned}&\mathbb{E}\left[\|z_{t+1}\|^2 - \|z_t\|^2\right]\\ &\leq \underbrace{\mathbb{E}[2z_t^\top(z_{t+1} - z_t + \beta A_{\theta_t}z_t)]}_{(c)} + \mathcal{O}\left(\beta^2\mathbb{E}[\|z_t\|^2] + \beta^2\right) + \beta\mathbb{E}\left[2z_t^\top(-A_{\theta_t})z_t\right],\end{aligned} \tag{12}$$

where the last term in (12) can be further upper bounded by $-2\beta\lambda_v\mathbb{E}[\|z_t\|^2]$.

One challenging part in our analysis is to bound term (c). Equivalently, we decompose the following term into three parts:

$$\begin{aligned}&\mathbb{E}\left[z_t^\top\left(-A_{\theta_t}z_t - \frac{1}{\beta}(z_{t+1} - z_t)\right)\right]\\ &= \underbrace{\mathbb{E}[z_t^\top(-A_{\theta_t} + A_{\theta_t}(s_t))z_t]}_{(d)} - \underbrace{\mathbb{E}[z_t^\top b_t(\theta_t)]}_{(e)} \underbrace{-\mathbb{E}\left[z_t^\top\frac{\omega(\theta_t) - \omega(\theta_{t+1})}{\beta}\right]}_{(f)}.\end{aligned} \tag{13}$$

Consider term (d) in (13). Unlike the case with linear function approximation, where the character function $\nabla V_\theta(s) = \phi(s)$ is independent with θ, here the character function $\phi_\theta(s)$ depends on θ. We use the geometric uniform ergodicity property of the MDP and the Lipschitz continuity of A_θ and $A_\theta(s)$ to decouple the dependence. More specifically, for any fixed θ, $\mathbb{E}[A_\theta(s_t)]$ converges to A_θ as t increases. Let $t = \tau_\beta$, then we have that

$$\begin{aligned}&\mathbb{E}\left[z_{\tau_\beta}^\top(-A_{\theta_{\tau_\beta}} + A_{\theta_{\tau_\beta}}(s_{\tau_\beta}))z_{\tau_\beta}\right]\\ &= \mathbb{E}\left[z_0^\top(-A_{\theta_0} + A_{\theta_0}(s_{\tau_\beta}))z_0\right] + \mathbb{E}\left[z_0^\top(-A_{\theta_{\tau_\beta}} + A_{\theta_{\tau_\beta}}(s_{\tau_\beta}) + A_{\theta_0} - A_{\theta_0}(s_{\tau_\beta})))z_0\right]\\ &\quad + \mathbb{E}\left[(z_{\tau_\beta} - z_0)^\top(-A_{\theta_{\tau_\beta}} + A_{\theta_{\tau_\beta}}(s_{\tau_\beta}))(z_{\tau_\beta} - z_0)\right] + 2\mathbb{E}\left[(z_{\tau_\beta} - z_0)^\top(-A_{\theta_{\tau_\beta}} + A_{\theta_{\tau_\beta}}(s_{\tau_\beta}))z_0\right],\end{aligned} \tag{14}$$

which can be further bounded using the mixing time τ_β and the Lipschitz property of A_θ and $A_\theta(s_{\tau_\beta})$. We note that from the update of z_t, we can bound $\|z_{\tau_\beta} - z_0\|$ and $\|z_0\|$ by $\|z_{\tau_\beta}\|$, hence the bound in (14) can be bounded in terms of $\|z_{\tau_\beta}\|$.

Similarly, note that $\mathbb{E}[b_t(\theta)]$ converges to 0 as $t \to \infty$, then we can also bound term (e) in (13):

$$\mathbb{E}[z_{\tau_\beta}^\top b_{\tau_\beta}(\theta_{\tau_\beta})] = \mathbb{E}[(z_{\tau_\beta} - z_0)^\top b_{\tau_\beta}(\theta_{\tau_\beta})] + \mathbb{E}[z_0^\top b_{\tau_\beta}(\theta_0)] + \mathbb{E}[z_0^\top(b_{\tau_\beta}(\theta_{\tau_\beta}) - b_{\tau_\beta}(\theta_0))], \tag{15}$$

which can be similarly bounded in terms of $\|z_{\tau_\beta}\|$.

The challenge of bounding the third term (f) in (13) lies in bounding the difference between $\omega(\theta_t)$ and $\omega(\theta_{t+1})$. One simple approach is to use the Lipschitz continuity of $\omega(\theta)$ and bound $\|\theta_t - \theta_{t+1}\|$ by a constant of order $\mathcal{O}(\alpha)$, but this will lead to a loose bound because the update $G_{t+1}(\theta_t, \omega_t)$ is actually an estimator of the gradient, which will also converge to zero. The key idea in our analysis is to bound term (f) in terms of the gradient of the objective function $\nabla J(\theta)$. Specifically, we first rewrite term $\langle z_t, \omega(\theta_t) - \omega(\theta_{t+1})\rangle = -\langle z_t, \nabla\omega(\hat{\theta}_t)(\theta_{t+1} - \theta_t)\rangle = -\alpha\langle\nabla\omega(\hat{\theta}_t)G_{t+1}(\theta_t, \omega_t)\rangle$, where $\hat{\theta}_t = c\theta_t + (1-c)\theta_{t+1}$ for some $c \in [0, 1]$. It can be shown that

$$\begin{aligned}\mathbb{E}\left[z_{\tau_\beta}^\top\frac{\omega(\theta_{\tau_\beta}) - \omega(\theta_{\tau_\beta+1})}{\beta}\right] &= -\frac{\alpha}{\beta}\mathbb{E}[z_{\tau_\beta}^\top\nabla\omega(\hat{\theta}_{\tau_\beta})(G_{\tau_\beta+1}(\theta_{\tau_\beta}, \omega_{\tau_\beta}) - G_{\tau_\beta+1}(\theta_{\tau_\beta}, \omega(\theta_{\tau_\beta})))]\\ &\quad - \frac{\alpha}{\beta}\mathbb{E}\left[z_{\tau_\beta}^\top\nabla\omega(\hat{\theta}_{\tau_\beta})\left(G_{\tau_\beta+1}(\theta_{\tau_\beta}, \omega(\theta_{\tau_\beta})) + \frac{\nabla J(\theta_{\tau_\beta})}{2}\right)\right] + \frac{\alpha}{\beta}\mathbb{E}\left[z_{\tau_\beta}^\top\nabla\omega(\hat{\theta}_{\tau_\beta})\left(\frac{\nabla J(\theta_{\tau_\beta})}{2}\right)\right].\end{aligned} \tag{16}$$

The first term in (16) can be bounded in terms of $\|z_{\tau_\beta}\|^2$ using the Lipschitz property of $G_{\tau_\beta+1}(\theta,\omega)$ in ω. The second term can be bounded using the uniform ergodicity of the MDP and the Lipschitz property of $z_0^\top \nabla\omega(\theta)\left(G_{\tau_\beta+1}(\theta,\omega(\theta)) + \frac{\nabla J(\theta)}{2}\right)$ in θ. The third term can be bounded in terms of $\|z_{\tau_\beta}\|^2$ and $\|\nabla J(\theta_{\tau_\beta})\|^2$. Combining all bounds together, we have the bound on term (f) in (13):

$$\begin{aligned}&\left|\mathbb{E}\left[z_{\tau_\beta}^\top \frac{\omega(\theta_{\tau_\beta}) - \omega(\theta_{\tau_\beta+1})}{\beta}\right]\right| \\ &\leq \mathcal{O}\left(\frac{\alpha}{\beta}\right)\mathbb{E}\left[\left\|z_{\tau_\beta}\right\|^2\right] + \mathcal{O}\left(\alpha\tau_\beta\right)\mathbb{E}\left[\left\|z_{\tau_\beta}\right\|\right] + \mathcal{O}\left(\alpha\tau_\beta\right) + \mathcal{O}\left(\frac{\alpha}{8\beta}\right)\mathbb{E}\left[\left\|\nabla J(\theta_{\tau_\beta})\right\|^2\right]. \end{aligned} \tag{17}$$

We combine all the bounds on terms $(d), (e)$ and (f) and hence get the error bound on (13):

$$\mathbb{E}\left[z_t^\top\left(-A_{\theta_t}z_t - \frac{1}{\beta}(z_{t+1} - z_t)\right)\right] \leq \mathcal{O}\left(\frac{\alpha}{\beta}\right)\mathbb{E}[\|z_t\|^2] + \mathcal{O}(\beta\tau_\beta) + \mathcal{O}\left(\frac{\alpha}{\beta}\right)\mathbb{E}[\|\nabla J(\theta_t)\|^2]. \tag{18}$$

Plugging the above bound in (12), we have the following recursive bound on the tracking error:

$$\mathbb{E}\left[\|z_{t+1}\|^2\right] \leq \mathcal{O}(1-\beta)\mathbb{E}\left[\|z_t\|^2\right] + \mathcal{O}(\alpha)\mathbb{E}\left[\|\nabla J(\theta_t)\|^2\right] + \mathcal{O}(\beta^2\tau_\beta). \tag{19}$$

Then by recursively applying the inequality in (19) and summing up w.r.t. t from 0 to $T-1$, we obtain the bound on the tracking error $\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]/T$:

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|z_t\|^2]}{T} \leq \mathcal{O}\left(\frac{1}{T\beta} + \frac{\alpha}{\beta}\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} + \beta\tau_\beta\right).$$

Step 3. In this step we bound the stochastic bias term $\mathbb{E}\left[\left\langle \nabla J(\theta_t), \frac{\nabla J(\theta_t)}{2} + G_{t+1}(\theta_t,\omega(\theta_t))\right\rangle\right]$. Similarly, we add and subtract $\nabla J(\theta_0)$ and $G_{\tau_\beta+1}(\theta_0,\omega(\theta_0))$, and obtain that

$$\begin{aligned}&\left\langle \nabla J(\theta_{\tau_\beta}), \frac{\nabla J(\theta_{\tau_\beta})}{2} + G_{\tau_\beta+1}(\theta_{\tau_\beta},\omega(\theta_{\tau_\beta}))\right\rangle \\ &= \left\langle \nabla J(\theta_0), \frac{\nabla J(\theta_0)}{2} + G_{\tau_\beta+1}(\theta_0,\omega(\theta_0))\right\rangle + \left(\left\langle \nabla J(\theta_{\tau_\beta}), \frac{\nabla J(\theta_{\tau_\beta})}{2} + G_{\tau_\beta+1}(\theta_{\tau_\beta},\omega(\theta_{\tau_\beta}))\right\rangle\right. \\ &\quad \left. - \left\langle \nabla J(\theta_0), \frac{\nabla J(\theta_0)}{2} + G_{\tau_\beta+1}(\theta_0,\omega(\theta_0))\right\rangle\right), \end{aligned} \tag{20}$$

which again can be bounded using the geometry uniform ergodicity of the MDP and the Lipschitz continuity of $\left\langle \nabla J(\theta), \frac{\nabla J(\theta)}{2} + G_{\tau_\beta+1}(\theta,\omega(\theta))\right\rangle$.

Step 4. Plugging in the bounds on the tracking error and the stochastic bias and rearranging the terms, then it follows that $\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} \leq U\sqrt{\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}} + V$, where U and V are some constants depending on the step sizes, and the explicit definitions can be found in (129). By solving the inequality of $\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T}$, we obtain that

$$\frac{\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla J(\theta_t)\|^2]}{T} \leq \mathcal{O}\left(\beta\tau_\beta + \frac{1}{T\beta} + \alpha\tau_\beta + \frac{1}{T\alpha}\right).$$

5 Conclusion

In this paper, we extend the on-policy non-linear TDC algorithm to the off-policy setting, and characterize its non-asymptotic error bounds under both the i.i.d. and the Markovian settings. We show that the non-linear TDC algorithm converges as fast as $\mathcal{O}(1/\sqrt{T})$ (up to a factor of $\log T$). The techniques and tools developed in this paper can be used to analyze a wide range of value-based RL algorithms with general smooth function approximation. **Limitations:** It is not clear yet whether the stationary points that the TDC converges to are second-order stationary or potentially saddle points. **Negative social impacts:** This work is a theoretical investigation of some fundamental RL algorithms, and therefore, the authors do not foresee any negative societal impact.

References

[1] T. Archibald, K. McKinnon, and L. Thomas. On the generation of Markov decision processes. *Journal of the Operational Research Society*, 46(3):354–361, 1995.

[2] L. Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings*, pages 30–37. Elsevier, 1995.

[3] D. P. Bertsekas. Dynamic Programming and Optimal Control 3rd edition, volume II. *Belmont, MA: Athena Scientific*, 2011.

[4] J. Bhandari, D. Russo, and R. Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Proc. Annual Conference on Learning Theory (CoLT)*, pages 1691–1692. PMLR, 2018.

[5] S. Bhatnagar, D. Precup, D. Silver, R. S. Sutton, H. Maei, and C. Szepesvári. Convergent temporal-difference learning with arbitrary smooth function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 22, pages 1204–1212, 2009.

[6] V. S. Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[7] Q. Cai, Z. Yang, J. D. Lee, and Z. Wang. Neural temporal-difference learning converges to global optima. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 11312–11322, 2019.

[8] G. Dalal, B. Szorenyi, and G. Thoppe. A tale of two-timescale reinforcement learning with the tightest finite-time bound. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 3701–3708, 2020.

[9] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. *Proceedings of Machine Learning Research*, 75:1–35, 2018.

[10] T. T. Doan. Nonlinear two-time-scale stochastic approximation: Convergence and finite-time performance. *arXiv preprint arXiv:2011.01868*, 2020.

[11] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. In *Proc. International Conference on Machine Learning (ICML)*, pages 1675–1685. PMLR, 2019.

[12] S. Ghadimi and G. Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[13] H. Gupta, R. Srikant, and L. Ying. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 4706–4715, 2019.

[14] M. Kaledin, E. Moulines, A. Naumov, V. Tadic, and H.-T. Wai. Finite time analysis of linear two-timescale stochastic approximation with markovian noise. In *Proc. Annual Conference on Learning Theory (CoLT)*, pages 2144–2203. PMLR, 2020.

[15] P. Karmakar and S. Bhatnagar. Two time-scale stochastic approximation with controlled Markov noise and off-policy temporal-difference learning. *Mathematics of Operations Research*, 43(1):130–151, 2018.

[16] V. R. Konda, J. N. Tsitsiklis, et al. Convergence rate of linear two-time-scale stochastic approximation. *The Annals of Applied Probability*, 14(2):796–819, 2004.

[17] C. Lakshminarayanan and C. Szepesvari. Linear stochastic approximation: How far does constant step-size and iterate averaging go? In *Proc. International Conference on Artificial Intelligence and Statistics*, pages 1347–1355, 2018.

[18] B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient td algorithms. In *Proc. International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 504–513. Citeseer, 2015.

[19] S. Ma, Y. Zhou, and S. Zou. Variance-reduced off-policy TDC learning: Non-asymptotic convergence analysis. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 14796–14806, 2020.

[20] S. Ma, Y. Zhou, and S. Zou. Greedy-GQ with variance reduction: Finite-time analysis and improved complexity. In *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[21] H. R. Maei. Gradient temporal-difference learning algorithms. *Thesis, University of Alberta*, 2011.

[22] H. R. Maei, C. Szepesvári, S. Bhatnagar, and R. S. Sutton. Toward off-policy learning control with function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 719–726, 2010.

[23] F. S. Melo, S. P. Meyn, and M. I. Ribeiro. An analysis of reinforcement learning with function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 664–671. ACM, 2008.

[24] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[25] A. Mokkadem, M. Pelletier, et al. Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms. *The Annals of Applied Probability*, 16(3):1671–1702, 2006.

[26] B. Neyshabur. Implicit regularization in deep learning. *arXiv preprint arXiv:1709.01953*, 2017.

[27] T. J. Perkins and D. Precup. A convergent form of approximate policy iteration. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1627–1634, 2003.

[28] R. Srikant and L. Ying. Finite-time error bounds for linear stochastic approximation and TD learning. In *Proc. Annual Conference on Learning Theory (CoLT)*, pages 2803–2830, Phoenix, USA, Jun 2019.

[29] J. Sun, G. Wang, G. B. Giannakis, Q. Yang, and Z. Yang. Finite-sample analysis of decentralized temporal-difference learning with linear function approximation. *arXiv preprint arXiv:1911.00934*, 2019.

[30] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction, Second Edition*. The MIT Press, Cambridge, Massachusetts, 2018.

[31] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 993–1000, 2009.

[32] R. S. Sutton, H. R. Maei, and C. Szepesvári. A convergent $O(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1609–1616, 2009.

[33] J. N. Tsitsiklis and B. Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, May 1997.

[34] Y. Wang, W. Chen, Y. Liu, Z.-M. Ma, and T.-Y. Liu. Finite sample analysis of the gtd policy evaluation algorithms in markov setting. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 5504–5513, 2017.

[35] Y. Wang and S. Zou. Finite-sample analysis of greedy-gq with linear function approximation under markovian noise. In *Proc. International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 11–20. PMLR, 2020.

[36] P. Xu and Q. Gu. A finite-time analysis of q-learning with neural network function approximation. In *Proc. International Conference on Machine Learning (ICML)*, pages 10555–10565. PMLR, 2020.

[37] T. Xu and Y. Liang. Sample complexity bounds for two timescale value-based reinforcement learning algorithms. *arXiv preprint arXiv:2011.05053*, 2020.

[38] T. Xu, S. Zou, and Y. Liang. Two time-scale off-policy TD learning: Non-asymptotic analysis over Markovian samples. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 10633–10643, 2019.

[39] S. Zou, T. Xu, and Y. Liang. Finite-sample analysis for SARSA with linear function approximation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 8665–8675, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]