Sample Complexity Bounds for Two Timescale Value-based Reinforcement Learning Algorithms

Tengyu Xu
The Ohio State University
xu.3260@osu.edu

Yingbin Liang
The Ohio State University
liang.889@osu.edu

Abstract

Two timescale stochastic approximation (SA) has been widely used in value-based reinforcement learning algorithms. In the policy evaluation setting, it can model the linear and nonlinear temporal difference learning with gradient correction (TDC) algorithms as linear SA and nonlinear SA, respectively. In the policy optimization setting, two timescale nonlinear SA can also model the greedy gradient-Q (Greedy-GQ) algorithm. In previous studies, the non-asymptotic analysis of linear TDC and Greedy-GQ has been studied in the Markovian setting, with diminishing or accuracy-dependent stepsize. For the nonlinear TDC algorithm, only the asymptotic convergence has been established. In this paper, we study the non-asymptotic convergence rate of two timescale linear and nonlinear TDC and Greedy-GQ under Markovian sampling and with accuracy-independent constant stepsize. For linear TDC, we provide a novel non-asymptotic analysis and show that it attains an ϵ-accurate solution with the optimal sample complexity of $\mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$ under a constant stepsize. For nonlinear TDC and Greedy-GQ, we show that both algorithms attain ϵ-accurate stationary solution with sample complexity $\mathcal{O}(\epsilon^{-2})$. It is the first non-asymptotic convergence result established for nonlinear TDC under Markovian sampling and our result for Greedy-GQ outperforms the previous result orderwisely by a factor of $\mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 Introduction

Two timescale stochastic approximation (SA) algorithms have wide applications in reinforcement learning (RL) [Sutton and Barto, 2018]. Typically, two timescale SA algorithms involve iterations of two types of variables updated at different speeds, i.e., the stepsizes for two iterates are chosen differently so that one iterate runs much faster than the other [Borkar, 1997, Borkar, 2009]. Such algorithms are widely used to solve both policy evaluation and policy optimization problems in RL, in which the goal of policy evaluation is to estimate the expected total reward (i.e. value function) of a target policy, and the goal of policy optimization is to search for a policy with the optimal expected total reward.

In the policy evaluation problem, temporal difference (TD) learning [Sutton, 1988] is one of the most widely used algorithms when a linear function class is utilized to approximate the value function. However, in the off-policy setting, in which the target policy to be evaluated is different from the behavior policy that generates samples, TD learning may diverge to infinity. To overcome such an issue, [Sutton et al., 2009] proposed the two timescale linear TD with gradient correction (TDC) algorithm, which has convergence guarantee in the off-policy setting. The two timescale linear TDC is a special case of two timescale linear SA, whose asymptotic convergence has been established in [Sutton et al., 2009, Borkar, 2009] and [Yu, 2017, Tadic, 2004, Yaji and Bhatnagar, 2016] for the i.i.d. and Markovian settings, respectively. The non-asymptotic convergence rate of two timescale linear TDC/SA has also been studied. In the i.i.d. setting, under diminishing stepsize, [Dalal et al., 2018b] established the sample complexity of $\mathcal{O}(\epsilon^{-1.5})$, and an improved complexity of $\mathcal{O}(\epsilon^{-1})$ was later established in [Dalal et al., 2019]. In the Markovian setting, [Xu et al., 2019] established the complexity of $\mathcal{O}(\epsilon^{-1.5}\log^2(1/\epsilon))$ under a diminishing stepsize, and [Gupta et al., 2019] established the complexity of $\mathcal{O}(\epsilon^{-1-\zeta}\log^2(1/\epsilon))$ under a ϵ-dependent stepsize, where

ζ can be an arbitrarily small positive constant. Recently, [Kaledin et al., 2020] provides a tighter complexity bound of $\mathcal{O}(\epsilon^{-1})$ for two timescale linear SA under a diminishing stepsize. Although having progressed significantly, existing convergence guarantee were established either under a *diminishing* stepsize or a drastically small ϵ-*level* stepsize, which yield very slow convergence and are rarely used in practice.

- *Thus, the first goal of this paper is to investigate the two timescale linear TDC* **under a constant stepsize (not ϵ-dependent)***, which is commonly adopted in practice, and to provide the finite-sample convergence guarantee for such a case. This necessarily requires a new approach differently from the existing ones.*

When a nonlinear function is utilized to approximate the value function, TD learning still suffers from the divergence issue [Tsitsiklis and Van Roy, 1997]. To address that, [Bhatnagar et al., 2009] proposed the two timescale nonlinear TDC, which can be modeled as a two timescale nonlinear SA. The asymptotic convergence of two timescale nonlinear SA has been well established in [Borkar, 1997, Tadic, 2004, Karmakar and Bhatnagar, 2018]. However, the non-asymptotic convergence of two timescale nonlinear SA has only been established in the i.i.d. setting under some restrict assumptions such as global (local) stability and local linearizion [Borkar and Pattathil, 2018, Mokkadem and Pelletier, 2006]. So far, the non-asymptotic convergence performance of two timescale nonlinear TDC has not been studied under the general Markovian sampling.

- *The second goal of this paper is to provide the first non-asymptotic convergence analysis for two timescale nonlinear TDC with a constant stepsize, under* **Markovian sampling***, and without restricted assumptions.*

Moreover, in the policy optimization problem, Q-learning [Watkins and Dayan, 1992] has been widely used and has achieved significant success in practice. However, in the function approximation setting, Q-learning does not have convergence guarantee [Baird, 1995] unless under some restricted regularity assumptions [Melo et al., 2008, Zou et al., 2019, Cai et al., 2019]. In corresponding to this, [Maei and Sutton, 2010] proposed the Greedy-GQ algorithm in the linear function approximation setting, in which the algorithm is guaranteed to converge to a locally optimal policy without restricted assumptions. Similarly to nonlinear TDC algorithms, Greedy-GQ also adopts a two timescale update scheme, and is a special case of two timescale nonlinear SA. Under single-sample update and Markovian sampling, [Wang and Zou, 2020] provided the non-asymptotic convergence rate of Greedy-GQ with diminishing stepsize, which achieves the complexity of $\mathcal{O}(\epsilon^{-3}\log(\epsilon^{-1}))$. However, such a rate does not attain the typical complexity order of nonconvex optimization, and can be potentially improved with a larger stepsize.

- *The last focus of this paper is to provide an improved non-asymptotic convergence rate for two timescale Greedy-GQ under a constant stepsize.*

1.1 Our Contributions

For two timescale linear TDC, we show that it achieves the sample complexity of $\mathcal{O}(\epsilon^{-1}\log(\epsilon^{-1}))$, which has the optimal dependence on ϵ due to the lower bound given in [Dalal et al., 2019]. Such a rate has been established in [Kaledin et al., 2020], but only under a *diminishing* stepsize, which is rarely used in practice due to the slow empirical performance. In contrast, our guarantee is established under a **constant (not ϵ-dependent) stepsize**, which is commonly used in practice. Our analysis approach leverages the mini-batch sampling for each iteration to control the convergence error, which is significantly different from that in [Kaledin et al., 2020], and can be of independent interest.

For two timescale nonlinear TDC, we establish the first non-asymptotic convergence rate under **Markovian sampling**. We show that the mini-batch two timescale nonlinear TDC algorithm achieves the sample complexity of $\mathcal{O}(\epsilon^{-2})$.

For two timescale Greedy-GQ, we show that mini-batch two timescale Greedy-GQ with a constant stepsize and under Markovian sampling achieves the sample complexity of $\mathcal{O}(\epsilon^{-2})$. Our result orderwisely outperforms the previous result of Greedy-GQ with diminishing stepsize in [Wang and Zou, 2020] by a factor of $\mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$.

1.2 Related Work

Due to the vast amount of studies on SA and value-based RL algorithms, we include here only the studies that are highly related to our work.

Policy evaluation with *linear* function approximation. In the on-policy setting, TD learning [Sutton, 1988] has been proposed to solve the policy evaluation problem in the linear function approximation setting. The non-asymptotic convergence rate of TD learning has been established in [Dalal et al., 2018a] for the i.i.d. setting and in [Bhandari et al., 2018, R. Srikant, 2019, Hu and Syed, 2019] for the Markovian setting. [Cai et al., 2019] explored the linearizable structure

of neural networks in the overparameterized regime, and studied the non-asymptotic convergence rate of TD learning with neural network approximation. [Zou et al., 2019] studied the convergence rate of SARSA with linear function approximation in the Markovian setting, which can been viewed as a policy evaluation with dynamic changing transition kernel.

In the off-policy setting, GTD, GTD2 and TDC have been proposed to solve the divergence issue of TD learning [Sutton et al., 2008, Sutton et al., 2009, Maei, 2011]. The convergence rate of one timescale GTD and GTD2 algorithms has been established in [Liu et al., 2015] by converting the objective into a convex-concave saddle problem in the i.i.d. setting, and was further generalized to the Markovian setting in [Wang et al., 2017]. For two timescale linear TDC, in the i.i.d. setting, the non-asymptotic analysis was provided in [Dalal et al., 2018b, Dalal et al., 2019]. In the Markovian setting, the non-asymptotic convergence rate was first established in [Xu et al., 2019] under diminishing stepsize and in [Gupta et al., 2019] under constant stepsize. The result in [Xu et al., 2019] was later improved by [Kaledin et al., 2020] to achieve the optimal convergence rate.

Policy evaluation with *nonlinear* function approximation. Two timescale nonlinear TDC is proposed by [Bhatnagar et al., 2009], in which a smooth nonlinear function is utilized to approximate the value function. Nonlinear TDC with i.i.d. samples is a special case of two time-scale nonlinear SA with martingale noise, whose asymptotic convergence has been established in [Bhatnagar et al., 2009, Maei, 2011] by using asymptotic convergence results in nonlinear SA [Borkar, 1997, Borkar, 2009, Tadic, 2004]. Under the global/local asymptotic stability assumptions or local linearizion assumption, the non-asymptotic convergence of two timescale nonlinear SA with martingale noise has been studied in [Borkar and Pattathil, 2018]. Under certain stability assumptions, the asymptotic convergence of two timescale nonlinear SA with Markov noise was established in [Karmakar et al., 2016, Karmakar and Bhatnagar, 2018]. A concurrent study [Qiu et al., 2020] also investigated nonlinear TDC and obtained the same sample complexity of $\mathcal{O}(\epsilon^{-2})$ as our result. However, [Qiu et al., 2019] only considered the i.i.d. setting, whereas we considered the more general Markovian setting.

Policy optimization with *linear* function approximation. Q-learning [Watkins and Dayan, 1992] is one of the most widely used value-based policy optimization algorithms. The asymptotic and non-asymptotic convergence have been established for Q-learning with linear function approximation in [Melo et al., 2008] and [Zou et al., 2019], respectively, under certain regularity assumption. Under a similar regularity assumption, [Cai et al., 2019] established the convergence rate of Q-Learning in the neural network approximation setting. However, without regularity assumptions, Q-Learning does not have convergence guarantee in the function approximation setting. [Maei et al., 2010] proposed two timescale Greedy-GQ to solve the divergence issue of Q-Learning with linear function approximation, and the asymptotic convergence of Greedy-GQ was also established therein. Recently, [Wang and Zou, 2020] studied the non-asymptotic convergence rate of Greedy-GQ under diminishing stepsize in the Markovian setting. In this paper, we provide an orderwisely better convergence rate than that in [Wang and Zou, 2020].

2 Markov Decision Process

Consider a Markov decision process (MDP) denoted $(\mathcal{S}, \mathcal{A}, \mathsf{P}, r, \gamma)$. Here, $\mathcal{S} \subset \mathbb{R}^d$ is a state space, $\mathcal{A}$ is an action set, $\mathsf{P} = \mathsf{P}(s'|s,a)$ is the transition kernel, $r(s,a,s')$ 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 time-step t, suppose the process is in some state $s_t \in \mathcal{S}$. Then an action $a_t \in \mathcal{A}$ is taken based on the distribution $\pi(\cdot|s_t)$, the system transitions to a next state $s_{t+1} \in \mathcal{S}$ governed by the transition kernel $\mathsf{P}(\cdot|s_t, a_t)$, and a reward $r_t = r(s_t, a_t, s_{t+1})$ is received. We assume the associated Markov chain $p(s'|s) = \sum_{a\in\mathcal{A}} p(s'|s,a)\pi(a|s)$ is ergodic, and let μ_π be the induced stationary distribution of this MDP, i.e., $\sum_s p(s'|s)\mu_\pi(s) = \mu_\pi(s')$. The state value function for policy π is defined as: $V^\pi(s) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t, s_{t+1})|s_0 = s, \pi]$, and the state-action value function is defined as: $Q^\pi(s,a) = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r(s_t, a_t, s_{t+1})|s_0 = s, a_0 = a, \pi]$. It is known that $V^\pi(s)$ is the unique fixed point of the Bellman operator T^π, i.e., $V^\pi(s) = T^\pi V^\pi(s) \coloneqq r^\pi(s) + \gamma \mathbb{E}_{s'|s} V^\pi(s')$, where $r^\pi(s) = \mathbb{E}_{a,s'|s} r(s,a,s')$ is the expected reward of the Markov chain induced by the policy π. We take the following standard assumption for the MDP in this paper, which has also been adopted in previous works [Bhandari et al., 2018, Zou et al., 2019, R. Srikant, 2019, Xu et al., 2019, Xu et al., 2020b].

Assumption 1 (Geometric ergodicity)**.** *There exist constants $\kappa > 0$ and $\rho \in (0,1)$ such that*

$$\sup_{s\in\mathcal{S}} d_{TV}(\mathbb{P}(s_t|s_0 = s), \mu_{\pi_b}(s_t)) \le \kappa\rho^t, \forall t \ge 0,$$

where $\mathbb{P}(s_t|s_0 = s)$ is the distribution of s_t conditioned on $s_0 = s$ and $d_{TV}(P,Q)$ denotes the total-variation distance between the probability measures P and Q.

Assumption 1 holds for any time-homogeneous Markov chain with finite state space and any

uniformly ergodic Markov chain with general state space [Bhandari et al., 2018, Zou et al., 2019, Xu et al., 2019].

3 Two Timescale TDC with Linear Function Approximation

In this section we first introduce the two timescale linear TDC algorithm to solve the policy evaluation problem, and then present our convergence rate result.

3.1 Algorithm

When $\mathcal{S}$ is large or infinite, a linear function $\hat{v}(s,\theta) = \phi(s)^\top\theta$ is often used to approximate the value function $V^\pi(s)$, where $\phi(s) \in \mathbb{R}^d$ is a fixed feature vector for state s and $\theta \in \mathbb{R}^d$ is a parameter vector. We can also write the linear approximation in the vector form as $\hat{v}(\theta) = \Phi\theta$, where Φ is the $|\mathcal{S}| \times d$ feature matrix. Without loss of generality, we assume that the feature vector $\|\phi(s)\|_2 \leq 1$ for all $s \in \mathcal{S}$ and the columns of the feature matrix Φ are linearly independent. Here we consider policy evaluation problem in the off-policy setting. Namely, a sample path $\{(s_t, a_t, s_{t+1})\}_{t\geq 0}$ is generated by the Markov chain according to a behavior policy π_b, but our goal is to obtain the value function of a target policy π, which is different from π_b.

To find a parameter $\theta^* \in \mathbb{R}^d$ with $\mathbb{E}_{\mu_{\pi_b}}\hat{v}(s,\theta^*) = \mathbb{E}_{\mu_{\pi_b}}T^\pi\hat{v}(s,\theta^*)$. The linear TDC algorithm [Sutton et al., 2009] updates the parameter by minimizing the mean-square projected Bellman error (MSPBE) objective, defined as

$$J(\theta) = \mathbb{E}_{\mu_{\pi_b}}[\hat{v}(s,\theta) - \Pi T^\pi\hat{v}(s,\theta)]^2,$$

where Π is the orthogonal projection operation onto the function space $\hat{\mathcal{V}} = \{\hat{v}(\theta) \mid \theta \in \mathbb{R}^d \text{ and } \hat{v}(\cdot,\theta) = \phi(\cdot)^\top\theta\}$. When the columns of the feature matrix Φ are linearly independent, [Sutton et al., 2009] shows that $J(\theta)$ is strongly convex and has $\theta^* = -A^{-1}b$ as its global minimum, i.e., $J(\theta^*) = 0$, where $A = \mathbb{E}_{\mu_{\pi_b}}[(\gamma\mathbb{E}_\pi[\phi(s')|s] - \phi(s))\phi(s)]$ and $b = \mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_\pi[r(s,a,s')|s]\phi(s)]$. A convenient way to find θ^* is to minimize the MSPBE objective function $J(\theta)$ using the gradient descent method: $\theta_{t+1} = \theta_t - \frac{\alpha}{2}\nabla J(\theta_t)$, where $\alpha > 0$ is the stepsize and the gradient $\nabla J(\theta)$ was derived by [Bhatnagar et al., 2009] as follows:

$$\begin{aligned}&-\frac{1}{2}\nabla J(\theta)\\ &= \mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_\pi[\delta(\theta)|s]\phi(s)] - \gamma\mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_\pi[\phi(s')|s]\phi(s)^\top]w(\theta),\end{aligned} \tag{1}$$

where $\delta(\theta) = r(s,a,s') + \gamma\hat{v}(s',\theta) - \hat{v}(s,\theta)$ is the temporal difference error, $w(\theta) := \mathbb{E}_{\mu_{\pi_b}}[\phi(s)\phi(s)^\top]^{-1}\mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_\pi[\delta(\theta)|s]\phi(s)]$. In practice, stochastic gradient descent (SGD) method is usually adopted to perform the update in eq. (1) approximately. However, directly sampling is not applicable to $w(\theta)$. To solve such an issue, an auxiliary parameter w_t can be introduced to estimate the vector $w(\theta_t)$, i.e., $w_t \approx w(\theta_t)$, by solving a linear SA with the following corresponding ODE:

$$\dot{w} = -\mathbb{E}_{\mu_{\pi_b}}[\phi(s)\phi(s)^\top]w + \mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_\pi[\delta(\theta)|s]\phi(s)].$$

Given w_t, the parameter θ_t can then be updated with a stochastic approximation of $\nabla J(\theta_t)$ obtained via directly sampling:

$$\theta_{t+1} = \theta_t + \alpha\frac{1}{|\mathcal{B}_t|}\sum_{j\in\mathcal{B}_t} g(\theta_t, w_t, x_j), \tag{2}$$

where $\mathcal{B}_t$ is the mini-batch sampled from the MDP, $g(\theta_t, w_t, x_j) = \rho(s_j,a_j)(\delta_j(\theta_t)\phi(s_j) - \gamma\phi(s_{j+1})\phi(s_j)^\top w_t)$, $\rho(s,a) = \pi(a|s)/\pi_b(a|s)$ is the importance weighting factor with $\rho_{\max}$ being its maximum value, and x_j denotes the sample (s_j, a_j, s_{j+1}).

Algorithm 1 is an **online** algorithm based on a **single sample path**. Algorithm 1 adopts a two timescale update scheme, in which parameters θ_t and w_t are updated simultaneously but with different stepsizes. Specifically, the main parameter θ_t iterates at a slow timescale with a smaller stepsize, and the auxiliary parameter w_t iterates at a fast timescale with a larger stepsize. By doing so, w_t can be close to $w(\theta_t)$ asymptotically, so that θ_t is updated approximately in the direction of $-\nabla J(\theta)$. Algorithm 1 utilizes an accuracy-independent constant stepsize, i.e., $\alpha, \beta = \mathcal{O}(1)$ for both the updates of θ_t and w_t, and a mini-batch of samples $\{(s_j, a_j, s_{j+1})\}_{i_t\leq j\leq i_t+M-1}$ are taken sequentially from the trajectory at each iteration to perform the update. As we will show later, linear TDC in this setting is guaranteed to converge to the global optimal with an arbitrary accuracy level.

Algorithm 1 Two Timescale Linear TDC

1: **Input:** batch size M, learning rate α and β
2: **Sampling:** A trajectory $\{s_j, a_j\}_{j\geq 0}$ is sampled by following the behaviour policy π_b
3: **Initialization:** θ_0 and w_0
4: **for** $t = 0, \cdots, T-1$ **do**
5: $\quad i_t = tM$
6: $\quad w_{t+1} = w_t + \beta\frac{1}{M}\sum_{j=i_t}^{i_t+M-1}(-\phi(s_j)\phi(s_j)^\top w_t + \rho(s_j,a_j)\delta_j(\theta_t)\phi(s_j))$
7: $\quad \theta_{t+1} = \theta_t + \alpha\frac{1}{M}\sum_{j=i_t}^{i_t+M-1}\rho(s_j,a_j)(\delta_j(\theta_t)\phi(s_j) - \gamma\phi(s_{j+1})\phi(s_j)^\top w_t)$
8: **end for**
9: **Output:** θ_T

3.2 Convergence Analysis

We define matrix $C = -\mathbb{E}_{\mu_{\pi_b}}[\phi(s)\phi(s)^\top]$. Let $\lambda_1 = \left|\lambda_{\max}(A^\top C^{-1} A)\right|$, $\lambda_2 = |\lambda_{\max}(C)|$ and $R_\theta = \|\theta^*\|_2$. The following theorem provides the convergence rate and sample complexity of Algorithm 1.

Theorem 1. *Suppose Assumption 1 hold. Consider Algorithm 1 of two timescale linear TDC update. Let the stepsize* $\alpha \leq \min\left\{\frac{1}{8\lambda_1}, \frac{\lambda_1\lambda_2}{12}, \frac{\sqrt{\lambda_2\beta}}{4\sqrt{6}\rho_{\max}}\frac{\lambda_2\sqrt{\lambda_2\beta}}{16\rho_{\max}^2}, \frac{\lambda_1\lambda_2\beta}{64\rho_{\max}^2}, \frac{\lambda_1\lambda_2^2\beta}{768}\right\}$, $\beta \leq \min\left\{\frac{1}{8\lambda_2}, \frac{\lambda_2}{4}\right\}$ *and the batch size* $M \geq 128\left(\rho_{\max}^2 + \frac{1}{\lambda_2^2}\right)\frac{1+(\kappa-1)\rho}{1-\rho}\max\left\{1, \frac{8\beta+8\lambda_2\beta^2}{\lambda_1\lambda_2\alpha}, \frac{8+12\lambda_1\alpha}{\lambda_1}\right\}$. *Then we have*

$$\mathbb{E}[\|\theta_T - \theta^*\|_2^2] \leq \left(1 - \frac{\min\{\lambda_1\alpha, \lambda_2\beta\}}{8}\right)^T \Delta_0 + \frac{A_1}{M}, \quad (3)$$

where $\Delta_0 = \|w_0 - w^*(\theta_0)\|_2^2 + \|\theta_0 - \theta^*\|_2^2$, *where* A_1 *is a constant defined in eq.* (30) *in Appendix A. Furthermore, let* $M \geq \frac{2A_1}{\epsilon}$ *and* $T \geq \frac{8}{\min\{\lambda_1\alpha, \lambda_2\beta\}}\ln\left(\frac{2\Delta_0}{\epsilon}\right)$. *The total sample complexity for Algorithm 1 to achieve an* ϵ*-accurate optimal solution* θ^*, *i.e.,* $\mathbb{E}[\|\theta_T - \theta^*\|_2^2] \leq \epsilon$, *is given by*

$$TM = \Theta\left(\frac{1}{\epsilon}\log\left(\frac{1}{\epsilon}\right)\right).$$

Theorem 1 shows that the convergence error of Algorithm 1 consists of two terms: the first term is the transient error decreasing at an exponential rate, and the second term is the variance error that diminishes as the batch size M increases. This is in contrast to the single-sample TDC under constant stepsizes, which suffers from the variance and bias errors with order $\mathcal{O}(\beta^2/\alpha)$ [Gupta et al., 2019]. Thus, ϵ-level small stepsizes α and β are required in single-sample TDC to reduce the variance error to achieve the required ϵ-accurate optimal solution, which can slow down the practical convergence speed significantly. In contrast, mini-batch TDC can attain high accuracy with a large constant (not ϵ-level) stepsize. Our result of $\mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$ achieves the optimal complexity order due to the lower bound given in [Dalal et al., 2019]. In contrast to the same sample complexity established in [Kaledin et al., 2020], which is applicable only under a diminishing stepsize, our result given in Theorem 1 is applicable under the constant stepsize, which is practically preferred due to the much better performance.

We next provide a sketch of the proof for Theorem 1.

Proof Sketch of Theorem 1. The proof of Theorem 1 consists of the following three steps. At t-th step, we call $\|\theta_t - \theta^*\|_2^2$ as the training error and $\|w_t - w(\theta_t)\|_2^2$ as the tracking error.

Step 1: We establish the following induction relationships for the tracking error:

$$\begin{aligned}&\mathbb{E}\left[\|w_{t+1} - w(\theta_{t+1})\|_2^2\right]\\ &\leq (1 - \Theta(\lambda_2\beta) + \Theta(\alpha^2/\beta))\mathbb{E}\left[\|w_t - w(\theta_t)\|_2^2\right]\\ &\quad + \Theta(\alpha^2/\beta + \lambda_1\alpha)\mathbb{E}[\|\theta_t - \theta^*\|_2^2] + \Theta(1/M). \quad (4)\end{aligned}$$

Step 2: We then establish the induction relationships for the training error:

$$\begin{aligned}&\mathbb{E}\left[\|\theta_{t+1} - \theta^*\|_2^2\right]\\ &\leq (1 - \Theta(\lambda_1\alpha) + \Theta(\alpha^2))\mathbb{E}\left[\|\theta_t - \theta^*\|_2^2\right]\\ &\quad + \Theta(\alpha + \alpha^2)\mathbb{E}\left[\|w_t - w(\theta_t)\|_2^2\right] + \Theta(1/M). \quad (5)\end{aligned}$$

Step 3: Combing eq. (4) and eq. (5) and letting the stepsize α and β and batch size M satisfy the requirement specified in Theorem 1, we establish the induction relationship of $\Delta_t = \mathbb{E}[\|\theta_t - \theta^*\|_2^2] + \mathbb{E}[\|w_t - w(\theta_t)\|_2^2]$ as follows:

$$\Delta_{t+1} \leq (1 - \Theta(\min\{\lambda_1\alpha, \lambda_2\beta\}))\,\Delta_t + \Theta(1/M). \quad (6)$$

Applying eq. (6) recursively from $t = T-1$ to 0 yields the desired convergence result. □

4 Two Timescale TDC with Nonlinear Function Approximation

In this section we first introduce the nonlinear two timescale TDC algorithm to solve the policy evaluation problem, then we provide our non-asymptotic convergence rate result.

4.1 Algorithm

In this section we consider policy evaluation problem with nonlinear function approximation, in which a parameterized smooth *nonlinear* function $\hat{v}(s,\theta)$ is used to approximate the value function $V^\pi(s)$. [Bhatnagar et al., 2009] proposed an algorithm to find a parameter for the approximator $\hat{v}(s,\theta)$, named nonlinear TDC. The nonlinear TDC updates the parameter by minimizing the following mean-square projected Bellman error objective defined as:

$$J(\theta) = \mathbb{E}_{\mu_\pi}[\hat{v}(s,\theta) - \Pi_\theta T^\pi \hat{v}(s,\theta)]^2, \quad (7)$$

where Π_θ is the orthogonal projection operation into the function space $\bar{\mathcal{V}} = \{\bar{V}(s,\zeta) \mid \zeta \in \mathbb{R}^d$ and $\bar{v}(s,\zeta) = \phi_\theta(s)^\top\zeta$ with $(\phi_\theta(s))_i = \nabla_{\theta_i}\hat{v}(s,\theta)\}$. In general, since $J(\theta)$ defined in eq. (7) is nonconvex with respect to the parameter θ, finding the global minimum of $J(\theta)$

is NP-hard. However, we can still apply gradient descent method to find a local optimum (i.e., first-order stationary point) of $J(\theta)$, via updating the parameter θ iteratively as $\theta_{t+1} = \theta_t - \frac{\alpha_t}{2}\nabla J(\theta_t)$, where $\alpha_t > 0$ is the stepsize and the gradient $\nabla J(\theta)$ was derived by [Bhatnagar et al., 2009] as follows:

$$-\frac{1}{2}\nabla J(\theta) = \mathbb{E}[\delta(\theta)\phi_\theta(s)] - \gamma\mathbb{E}[\phi_\theta(s')\phi_\theta(s)^\top]w(\theta) - h(\theta, w(\theta)), \tag{8}$$

where $\delta(\theta) = r(s,a,s') + \gamma\hat{v}(s',\theta) - \hat{v}(s,\theta)$ is the temporal difference and

$$w(\theta) \coloneqq \mathbb{E}[\phi_\theta(s)\phi_\theta(s)^\top]^{-1}\mathbb{E}[\delta(\theta)\phi_\theta(s)],$$
$$h(\theta, u) \coloneqq \mathbb{E}[(\delta(\theta) - \phi_\theta(s)^\top u)\nabla^2_\theta V_\theta(s)u].$$

Similarly to linear TDC studied in Section 3, in order to estimate the gradient in eq. (8), an auxiliary parameter w_t can be used to estimate the vector $w(\theta_t)$, i.e., $w_t \approx w(\theta_t)$, by solving a linear SA with the following corresponding ODE:

$$\dot{w} = -\mathbb{E}[\phi_\theta(s)\phi_\theta(s)^\top]w + \mathbb{E}[\delta(\theta)\phi_\theta(s)]. \tag{9}$$

Given w_t, the parameter θ_t can then be updated with a stochastic approximation of $\nabla J(\theta_t)$ obtained via directly sampling:

$$\theta_{t+1} = \theta_t + \alpha_t \frac{1}{|\mathcal{B}_t|}\sum_{j\in\mathcal{B}_t} g(\theta_t, w_t, x_j), \tag{10}$$

where $\mathcal{B}_t$ is the minibatch sampled from the MDP, x_j denotes the sample (s_j, a_j, s_{j+1}) and we define $g(\theta_t, w_t, x_j) = \delta_j(\theta_t)\phi_{\theta_t}(s_j) - \gamma\phi_{\theta_t}(s_{j+1})\phi_{\theta_t}(s_j)^\top w_t - h_j(\theta_t, w_t)$, where $h_j(\theta_t, w_t) = (\delta_j(\theta_t) - \phi_{\theta_t}(s_j)^\top w_t)\nabla^2_\theta V_{\theta_t}(s_j)w_t$. The nonlinear TDC algorithm is shown in Algorithm 2. Similarly to Algorithm 1, here we also use a mini-batch of samples for each update.

Algorithm 2 Two Time-scale Nonlinear TDC

1: **Input** batch size M, learning rate α and β
2: **Sampling:** A trajectory $\{s_j, a_j\}_{j\geq 0}$ is sampled by following the policy π
3: **Initialization:** θ_0 and w_0
4: **for** $t = 0, 1, ..., T-1$ **do**
5: $i_t = tM$
6: $w_{t+1} = w_t + \beta\frac{1}{M}\sum_{j=i_t}^{i_t+M-1}(-\phi_{\theta_t}(s_j)\phi_{\theta_t}(s_j)^\top w_t + \delta_j(\theta_t)\phi_{\theta_t}(s_j))$
7: $\theta_{t+1} = \theta_t + \alpha\frac{1}{M}\sum_{j=i_t}^{i_t+M-1}(\delta_j(\theta_t)\phi_{\theta_t}(s_j) - \gamma\phi_{\theta_t}(s_{j+1})\phi_{\theta_t}(s_j)^\top w_t - h_j(\theta_t, w_t))$
8: **end for**
Output: $\tilde{\theta}_{\hat{T}}$ with $\hat{T}$ chosen uniformly from $\{1, \cdots, T\}$

4.2 Convergence Analysis

Our analysis of Algorithm 2 will be based on the following assumptions.

Assumption 2 (Bounded feature). *For any state $s \in \mathcal{S}$ and any vector $\theta \in \mathbb{R}^d$, we have $\|\phi_\theta(s)\|_2 \leq C_\phi$, $|V(s,\theta)| \leq C_v$ and $\left\|\nabla^2_\theta V(s,\theta)\right\|_F \leq D_v$, where C_ϕ, C_v and D_v are positive constants.*

Assumption 3 (Smoothness). *For any state $s \in \mathcal{S}$ and any vector $\theta, \theta' \in \mathbb{R}^d$, we have $|V(s,\theta) - V(s,\theta')| \leq L_v\|\theta - \theta'\|_2$, $\|\phi_\theta(s) - \phi_{\theta'}(s)\|_2 \leq L_\phi\|\theta - \theta'\|_2$, and $\left\|\nabla^2_\theta V(s,\theta) - \nabla^2_\theta V(s,\theta')\right\|_2 \leq L_h\|\theta - \theta'\|_2$, where L_v, L_ϕ, and L_h are positive constants.*

Assumption 4 (Non-singularity). *For any vector $\theta \in \mathbb{R}^d$, we have $eig\{\mathbb{E}[\phi_\theta(s)\phi_\theta(s)^\top]\} \geq \lambda_v$, where λ_v is a positive constant.*

Assumption 5 (Lipschitz gradient). *For any vector θ, θ' and $w, w' \in \mathbb{R}^d$, and any sample x, we have $\|\nabla J(\theta) - \nabla J(\theta')\|_2 \leq L_J\|\theta - \theta'\|_2$, and $\|g(\theta, w, x) - g(\theta, w', x)\|_2 \leq L_e\|w - w'\|_2$ where L_J and L_e are positive constants.*

Assumptions 2-5 are equivalent to the assumptions adopted in the original nonlinear TDC analysis [Bhatnagar et al., 2009], and can be satisfied by appropriately choosing the approximation function class $\hat{v}(s,\theta)$. The following theorem characterizes the converge rate and sample complexity of Algorithm 2.

Theorem 2. *Consider the two timescale nonlinear TDC algorithm in Algorithm 2. Suppose Assumptions 1-5 hold. Let the stepsize $\beta \leq \min\{\frac{\lambda_v}{8C_\phi^4}, \frac{8}{\lambda_v}\}$ and $\alpha \leq \min\{\frac{1}{2L_J}, \frac{\lambda_v\beta}{8\sqrt{2}L_wL_e}, \frac{L_J\lambda_v^2\beta^2}{384L_w^2L_e^2}\}$. We have*

$$\mathbb{E}\left[\left\|\nabla J(\theta_{\hat{T}})\right\|_2^2\right] \leq \frac{8(J(\theta_0) - \mathbb{E}[J(\theta_T)])}{\alpha T} + \frac{B_1\|w_0 - w(\theta_0)\|_2^2}{T} + \frac{B_2}{M},$$

where B_1 and B_2 are constants defined in Appendix B in eq. (48). Furthermore, let $M \geq \frac{2B_2}{\epsilon}$ and $T \geq \frac{2}{\epsilon}\left[\frac{8J(\theta_0)}{\alpha} + B_1\|w_0 - w(\theta_0)\|_2^2\right]$. The total sample complexity for Algorithm 2 to achieve an ϵ-accurate stationary point, i.e., $\mathbb{E}\left[\left\|\nabla J(\theta_{\hat{T}})\right\|_2^2\right] \leq \epsilon$, is given by

$$TM = \Theta\left(\frac{1}{\epsilon^2}\right).$$

Theorem 2 shows that the convergence error of Algorithm 2 consists of three terms: the first two terms are the transient error decreasing at a sublinear rate as T increases, and the third term contains the variance and bias errors that diminish as the batch size M increases. We next provide a sketch of the proof for Theorem 2.

Proof Sketch of Theorem 2. The proof of Theorem 2 consists of the following four steps.

Step 1: We first provide Lemma 4 to show that $w(\theta)$ is L_w-Lipschitz:

$$\|w(\theta) - w(\theta')\|_2 \leq L_w \|\theta - \theta'\|_2, \quad \text{for all } \theta, \theta' \in \mathbb{R}^d.$$

This property is crucial for the convergence analysis of two time-scale nonlinear TDC. It indicates that if θ_t changes slowly, then $w(\theta_t)$ also changes slowly. This allows our finite time analysis to be over a slowly changing linear SA with corresponding ODE defined in eq. (9), guaranteeing that $\|w_t - w(\theta_t)\|_2^2$ is small in an amortized sense.

Step 2: We then establish the induction relationships for the tracking error $\|w_t - w(\theta_t)\|_2^2$:

$$\begin{aligned}
&\mathbb{E}\left[\|w_{t+1} - w(\theta_{t+1})\|_2^2\right] \\
&\leq (1 - \Theta(\lambda_v \beta))\, \mathbb{E}\left[\|w_t - w(\theta_t)\|_2^2\right] \\
&\quad + \Theta(\alpha^2/\beta)\mathbb{E}\left[\|\nabla J(\theta_t)\|_2^2\right] + \Theta(1/M). \qquad (11)
\end{aligned}$$

Step 3: We then establish the induction relationships for the gradient norm $\|\nabla J(\theta_t)\|_2^2$:

$$\begin{aligned}
&\big(\Theta(\alpha) - \Theta(\alpha^2)\big)\, \mathbb{E}\left[\|\nabla J(\theta_t)\|_2^2\right] \\
&\leq \mathbb{E}[J(\theta_t)] - \mathbb{E}[J(\theta_{t+1})] \\
&\quad + \Theta(\alpha + \alpha^2)\mathbb{E}\left[\|w_t - w(\theta_t)\|_2^2\right] + \Theta(1/M). \qquad (12)
\end{aligned}$$

Step 4: Applying eq. (11) and eq. (12) recursively from $t = T-1$ to 0 and combing those two results together yield

$$\begin{aligned}
&\big(\Theta(\alpha) - \Theta(\alpha^2) - \Theta(\alpha^3)\big) \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla J(\theta_t)\|_2^2\right] \\
&\leq J(\theta_0) - \mathbb{E}[J(\theta_T)] \\
&\quad + \Theta((\alpha + \alpha^2)/\beta)\, \|w_0 - w(\theta_0)\|_2^2 + \Theta(1/M).
\end{aligned}$$

Letting the stepsize α and β and the batch size M satisfies the requirement specified in Theorem 2, we can then obtain the desired convergence result. □

5 Policy Optimization: Greedy-GQ Algorithm

In this section, we will provide the non-asymptotic convergence result of Greedy-GQ [Maei et al., 2010], which is also a two timescale nonlinear SA algorithm.

Greedy-GQ was proposed in [Maei et al., 2010] to solve the divergence issue of Q-Learning in the linear function approximation setting. In Greedy-GQ, the goal of the agent is to learn an optimal policy for the MDP with respect to the total expected discounted reward. In the linear function approximation setting, a linear function $\hat{Q}(s, a, \theta) = \phi(s, a)^\top \theta$ is used to approximate the state-action value function $Q(s, a)$, where $\phi(s, a) \in \mathbb{R}^d$ is a fixed feature vector for state-action pair (s, a) and $\theta \in \mathbb{R}^d$ is a parameter vector. Without loss of generality, we assume that the feature vector $\|\phi(s, a)\|_2 \leq 1$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and the columns of the feature matrix Φ are linearly independent. In this setting, we hope to find a solution θ that satisfies

$$\Pi T^{\pi_\theta} \hat{Q}(s, a, \theta) = \hat{Q}(s, a, \theta), \quad \text{for all } (s, a) \in \mathcal{S} \times \mathcal{A}, \qquad (13)$$

where π_θ is the soft-max greedy policy with respect to the state-action value function $\hat{Q}(s, a, \theta)$, i.e., $\pi_\theta(a|s) = \frac{\exp(\tau \hat{Q}(s,a,\theta))}{\sum_{a' \in \mathcal{A}} \exp(\tau \hat{Q}(s,a',\theta))}$, where $\tau > 0$ is the temperature parameter, and T^{π_θ} denotes the Bellman operator with policy π_θ. Similarly to the TDC algorithms, Greedy-GQ searches a parameter that satisfies eq. (13) by minimizing a projected Bellman error objective function defined as:

$$J(\theta) = \mathbb{E}_{\mu_{\pi_b}}[\hat{Q}(s, a, \theta) - \Pi T^{\pi_\theta} \hat{Q}(s, a, \theta)]^2, \qquad (14)$$

where $\delta(\theta) = r(s, a, s') + \gamma \hat{Q}(s', b, \theta) - \hat{Q}(s, a, \theta)$ is the temporal difference error, with $a \sim \pi_\theta(\cdot|s)$ and $b \sim \pi_\theta(\cdot|s')$. Since $J(\theta)$ is nonconvex and smooth everywhere, we can apply gradient descent method to find a local optimal (stationary point) of the objective $J(\theta)$ via applying the update $\theta_{t+1} = \theta_t - \frac{\alpha}{2} \nabla J(\theta_t)$ iteratively, in which

$$\begin{aligned}
&\frac{1}{2} \nabla J(\theta_t) \\
&= -\mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_{\pi_\theta}[\delta(\theta)|s, a]\phi(s, a)] \\
&\quad + \gamma \mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_{\pi_\theta}[\phi(s', b)|s, a]\phi(s, a)^\top] w(\theta)
\end{aligned}$$

where

$$w(\theta) = \mathbb{E}_{\mu_{\pi_b}}[\phi(s, a)\phi(s, a)^\top]^{-1} \mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_{\pi_\theta}[\delta(\theta)|s, a]\phi(s, a)].$$

Similarly to the nonlinear TDC algorithms in section 4, here an auxiliary parameter w_t is adopted to estimate the vector $w(\theta_t)$ by solving a linear SA with the following corresponding ODE:

$$\dot{w} = -\mathbb{E}_{\mu_{\pi_b}}[\phi(s, a)\phi(s, a)^\top] w + \mathbb{E}_{\mu_{\pi_b}}[\mathbb{E}_{\pi_\theta}[\delta(\theta)|s, a]\phi(s, a)].$$

Then, θ_t can be updated via direct sampling

$$\theta_{t+1} = \theta_t + \alpha_t \frac{1}{|\mathcal{B}_t|} \sum_{j \in \mathcal{B}_t} g(\theta_t, w_t, x_j),$$

where $\rho_\theta(s, a) = \pi_\theta(a|s)/\pi_b(a|s)$ is the importance weighting factor bounded by $\rho_{\max}$ and we define $g(\theta_t, w_t, x_j) = \rho_{\theta_t}(s_{j+1}, a_{j+1})(\delta_j(\theta_t)\phi(s_j, a_j) -$

$\gamma\phi(s_{j+1},a_{j+1})\phi(s_j,a_j)^\top w_t)$. The two timescale Greedy-GQ algorithm is shown below.

Algorithm 3 Two Timescale Greedy-GQ

Input: batch size M, learning rate α and β
Sampling: A trajectory $\{s_j,a_j\}_{j\geq 0}$ is sampled by following the behaviour policy π_b
Initialization: θ_0 and w_0
for $t=0,\cdots,T-1$ **do**
  $i_t = tM$
  $w_{t+1} = w_t + \beta\frac{1}{M}\sum_{j=i_t}^{i_t+M-1}(-\phi(s_j,a_j)\phi(s_j,a_j)^\top w_t + \rho_{\theta_j}(s_j,a_j)\delta_j(\theta_t)\phi(s_j,a_j))$
  $\theta_{t+1} = \theta_t+\alpha\frac{1}{M}\sum_{j=i_t}^{i_t+M-1}\rho_{\theta_t}(s_{j+1},a_{j+1})(\delta_j(\theta_t)\phi(s_j)-\gamma\phi(s_{j+1},a_{j+1})\phi(s_j,a_j)^\top w_t)$
end for
Output: $\tilde{\theta}_{\hat{T}}$ with $\hat{T}$ chosen uniformly from $\{1,\cdots,T\}$

By slightly abusing notations in Section 4, we make the follow standard assumptions.

Assumption 6 (Non-singularity). *We have* $(\max_{\theta\in\mathbb{R}^d}\left|\lambda_{\max}\{A_\theta^\top C^{-1}A_\theta\}\right|)^{-1} = \lambda_1$ *and* $|\lambda_{\max}\{C\}| = \lambda_2$, *where* $A_\theta = \mathbb{E}_{\mu_{\pi_b}}[(\gamma\mathbb{E}_{\pi_\theta}[\phi(s')|s] - \phi(s))\phi(s)^\top]$ *and* $C = -\mathbb{E}_{\mu_{\pi_b}}[\phi(s)\phi(s)^\top]$ *and* λ_1 *and* λ_2 *are positive constants.*

Assumption 7 (Bounded importance factor). *For any state-action pair* $(s,a)\in\mathcal{S}\times\mathcal{A}$ *and any* $\theta\in\mathbb{R}^d$, *we have* $\rho_\theta(s,a)\leq\rho_{\max}$, *where* $\rho_{\max}$ *is a positive constant.*

Note that Assumption 7 can be satisfied when the behaviour policy is non-degenerated for all states. Moreover, we make the following Lipschitz property of the gradient $\nabla J(\theta)$, which has been verified in [Wang and Zou, 2020] in a similar setting.

Lemma 1. *Suppose Assumption 1 and Assumption 6 hold, for any* $\theta,\theta'\in\mathbb{R}^d$, *we have* $\|\nabla J(\theta)-\nabla J(\theta')\|_2\leq L_J\|\theta-\theta'\|_2$, *where* L_J *is a positive constant.*

Note that the Greedy-GQ algorithm in Algorithm 3 and nonlinear TDC algorithm in Algorithm 2 share similar structures. Both objectives are nonconvex and both algorithms adopt a two timescale update scheme, in which the fast timescale iteration corresponds to a linear SA and the slow time-scale iteration corresponds to a nonlinear SA. Thus, the analysis of two time-scale nonlinear TDC in Section 4 can be extended to study the convergence rate of Greedy-GQ algorithm. The following theorem characterizes the convergence rate and sample complexity of Algorithm 3.

Theorem 3. *Consider the two timescale Greedy-GQ algorithm in Algorithm 3. Suppose Assumptions 1, 6 and 7 hold. Let the stepsize* $\beta\leq\min\{\frac{\lambda_2}{4},\frac{8}{\lambda_2}\}$ *and* $\alpha\leq\min\{\frac{1}{8L_J},\frac{\lambda_2\sqrt{\lambda_2}}{8\sqrt{2}\rho_{\max}}\beta,\frac{L_J\lambda_2^3}{5312\rho_{\max}^2\lambda_1^2}\beta^2\}$, *and batch size* $M\geq\frac{1+(\kappa-1)\rho}{1-\rho}\max\{128\left(\rho_{\max}^2+\frac{1}{\lambda_2^2}\right)[1+\frac{\lambda_2^2\beta}{4\alpha^2}(\frac{2\beta}{\lambda_2}+2\beta^2)],\frac{\beta^2\lambda_2^3(\rho_{\max}+1)^4}{\rho_{\max}^2\alpha^2}\}$. *We have*

$$\mathbb{E}\left[\left\|\nabla J(\theta_{\hat{T}})\right\|_2^2\right] \leq \frac{8(J(\theta_0)-\mathbb{E}[J(\theta_T)])}{\alpha T}+\frac{192\rho_{\max}^2}{\lambda_2\beta}\frac{\|w_0-w^*(\theta_0)\|_2^2}{T} + \frac{32C_1[1+(\kappa-1)\rho]}{M(1-\rho)},$$

where C_1 *is a positive constant defined in eq.* (60) *in Appendix C. Furthermore, let* $M\geq\frac{64C_2[1+(\kappa-1)\rho]}{(1-\rho)\epsilon}$ *and* $T\geq\frac{2}{\epsilon}\left[\frac{8J(\theta_0)}{\alpha}+\frac{192\rho_{\max}^2\|w_0-w(\theta_0)\|_2^2}{\lambda_2\beta}\right]$. *The total sample complexity for Algorithm 2 to achieve an* ϵ*-accurate stationary point, i.e.,* $\mathbb{E}\left[\left\|\nabla J(\theta_{\hat{T}})\right\|_2^2\right]\leq\epsilon$, *is given by*

$$TM=\Theta\left(\frac{1}{\epsilon^2}\right).$$

Similarly to Theorem 2, in Theorem 3 we show that Algorithm 3 converges to an ϵ-accurate stationary point with sample complexity $\mathcal{O}(\epsilon^{-2})$. Note that [Wang and Zou, 2020] studied the convergence rate of two timescale Greedy-GQ with diminishing stepsize, which achieves the complexity of $\mathcal{O}(\epsilon^{-3}\log(1/\epsilon))$. Theorem 3 for two timescale Greedy-GQ with constant stepsize outperforms the result in [Wang and Zou, 2020] by a factor of $\mathcal{O}(\epsilon^{-1}\log(1/\epsilon))$, indicating that the constant stepsize can significantly improve the convergence rate of two timescale Greedy-GQ algorithm.

6 Conclusion

In this paper, we study the convergence rate for two timescale linear and nonlinear TDC and Greedy-GQ under Markovian sampling and constant stepsize. Specifically, we show that the complexity result of linear TDC orderwisely achieves the optimal convergence rate under a constant stepsize. Our result for nonlinear TDC is the first under Markovian sampling. Moreover, our sample complexity result of Greedy-GQ outperforms the previous result orderwisely. For future work, it is interesting to apply more advance optimization techniques, e.g., acceleration, variance reduction, to further improve the convergence performance of the value-based RL algorithms studied in this paper.

7 Acknowledgement

The work was supported partially by the U.S. National Science Foundation under the grants CCF-1801855 and CCF-1900145.

References

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

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

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

[Borkar, 1997] Borkar, V. S. (1997). Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294.

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

[Borkar and Pattathil, 2018] Borkar, V. S. and Pattathil, S. (2018). Concentration bounds for two time scale stochastic approximation. In *Proc. Conference on Communication, Control, and Computing (Allerton)*, pages 504–511.

[Cai et al., 2019] Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. (2019). Neural temporal-difference and Q-learning provably converge to global optima. *arXiv preprint arXiv:1905.10027*.

[Dalal et al., 2019] Dalal, G., Szorenyi, B., and Thoppe, G. (2019). A tale of two-timescale reinforcement learning with the tightest finite-time bound. *arXiv preprint arXiv:1911.09157*.

[Dalal et al., 2018a] Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. (2018a). Finite sample analyses for TD (0) with function approximation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*.

[Dalal et al., 2018b] Dalal, G., Szorenyi, B., Thoppe, G., and Mannor, S. (2018b). Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In *Proc. Conference on Learning Theory (COLT)*.

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

[Hu and Syed, 2019] Hu, B. and Syed, U. (2019). Characterizing the exact behaviors of temporal difference learning algorithms using Markov jump linear system theory. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 8477–8488.

[Kaledin et al., 2020] Kaledin, M., Moulines, E., Naumov, A., Tadic, V., and Wai, H.-T. (2020). Finite time analysis of linear two-timescale stochastic approximation with Markovian noise. *arXiv preprint arXiv:2002.01268*.

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

[Karmakar et al., 2016] Karmakar, P., Ramaswamy, A., Bhatnagar, S., and Borkar, V. S. (2016). Asymptotic and non-asymptotic convergence properties of stochastic approximation with controlled markov noise without ensuring stability.

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

[Maei, 2011] Maei, H. R. (2011). *Gradient temporal-difference learning algorithms*. PhD thesis, University of Alberta.

[Maei and Sutton, 2010] Maei, H. R. and Sutton, R. S. (2010). GQ (lambda): A general gradient algorithm for temporal-difference prediction learning with eligibility traces. In *Proc. Artificial General Intelligence (AGI)*. Atlantis Press.

[Maei et al., 2010] Maei, H. R., Szepesvári, C., Bhatnagar, S., and Sutton, R. S. (2010). Toward off-policy learning control with function approximation. In *Proc. International Conference of Machine Learning (ICML)*.

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

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

[Qiu et al., 2020] Qiu, S., Yang, Z., Wei, X., Ye, J., and Wang, Z. (2020). Single-timescale stochastic nonconvex-concave optimization for smooth nonlinear TD learning. *arXiv preprint arXiv:2008.10103*.

[Qiu et al., 2019] Qiu, S., Yang, Z., Ye, J., and Wang, Z. (2019). On the finite-time convergence of actor-critic algorithm. In *Optimization Foundations for Reinforcement Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS)*.

[R. Srikant, 2019] R. Srikant, L. Y. (2019). Finite-time error bounds for linear stochastic approximation and TD learning. *arXiv preprint arXiv:1902.00923*.

[Sutton, 1988] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44.

[Sutton and Barto, 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

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

[Sutton et al., 2008] Sutton, R. S., Szepesvári, C., and Maei, H. R. (2008). A convergent o(n) algorithm for off-policy temporal-difference learning with linear function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 1609–1616.

[Tadic, 2004] Tadic, V. B. (2004). Almost sure convergence of two time-scale stochastic approximation algorithms. In *Proc. American Control Conference*, volume 4, pages 3802–3807.

[Tsitsiklis and Van Roy, 1997] Tsitsiklis, J. N. and Van Roy, B. (1997). Analysis of temporal-diffference learning with function approximation. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 1075–1081.

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

[Wang and Zou, 2020] Wang, Y. and Zou, S. (2020). Finite-sample analysis of greedy-gq with linear function approximation under markovian noise. *arXiv preprint arXiv:2005.10175*.

[Watkins and Dayan, 1992] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3-4):279–292.

[Xu et al., 2020a] Xu, T., Wang, Z., and Liang, Y. (2020a). Improving sample complexity bounds for actor-critic algorithms. *arXiv preprint arXiv:2004.12956*.

[Xu et al., 2020b] Xu, T., Wang, Z., Zhou, Y., and Liang, Y. (2020b). Reanalysis of variance reduced temporal difference learning. In *Proc. International Conference on Learning Representations (ICLR)*.

[Xu et al., 2019] Xu, T., Zou, S., and Liang, Y. (2019). 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.

[Yaji and Bhatnagar, 2016] Yaji, V. and Bhatnagar, S. (2016). Stochastic recursive inclusions in two timescales with non-additive iterate dependent Markov noise. *arXiv preprint arXiv:1611.05961*.

[Yu, 2017] Yu, H. (2017). On convergence of some gradient-based temporal-differences algorithms for off-policy learning. *arXiv preprint arXiv:1712.09652*.

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