Decentralized Adaptive TD(λ) Learning With Linear Function Approximation: Nonasymptotic Analysis

Junlong Zhu, Tao Mao, Mingchuan Zhang, *Member, IEEE*, Quanbo Ge, Qingtao Wu, and Keqin Li, *Fellow, IEEE*

***Abstract*—In multiagent reinforcement learning, policy evaluation is a central problem. To solve this problem, decentralized temporal-difference (TD) learning is one of the most popular methods, which has been investigated in recent years. However, existing decentralized variants of TD learning often suffer from slow convergence due to the sensitive selection of learning rates. Inspired by the great success of adaptive gradient methods in the training of deep neural networks, this article proposes a decentralized adaptive TD(λ) learning algorithm for general λ with linear function approximation, referred to as D-AMSTD(λ), which can mitigate the selective sensitivity of learning rates. Furthermore, we establish the finite-time performance bounds of D-AMSTD(λ) under the Markovian observation model. The theoretical results show that D-AMSTD(λ) can linearly converge to an arbitrarily small size of neighborhood of the optimal weight. Finally, we verify the efficacy of D-AMSTD(λ) through a variety of experiments. The results show that D-AMSTD(λ) outperforms existing decentralized TD learning methods.**

***Index Terms*—Finite-time bounds, multiagent reinforcement learning (MARL), policy evaluation, temporal-difference (TD) learning.**

I. Introduction

IN MODERN machine learning, one of the most popular paradigms is reinforcement learning (RL), which learns an optimal policy via maximizing a cumulative reward [1]. Therefore, how to estimate the cumulative reward is a central problem in RL. To solve this problem, temporal-difference (TD) learning is one of the most dominant methods in RL [2], which estimates the cumulative reward through an iteration process under a given policy. The classical TD uses a tabular to represent the cumulative reward, where the estimates of entry-wise cumulative reward are stored state-by-state. Nevertheless, the "curse of dimensionality" can be caused by the tabular TD with large state spaces. To address this issue, various approximators can be used to approximate the cumulative reward [1], [3]. In general, approximators are divided into linear and nonlinear approximations. In particular, deep RL has been successfully applied in many domains [4], where deep neural networks are regarded as nonlinear approximators. For example, AlphaGo [5], AlphaGo Zero [6], games [7], etc.

Despite the success of RL with nonlinear approximators, however, the convergence of TD learning cannot be guaranteed through nonlinear methods [8]. To ensure the convergence of TD learning, linear function approximation methods are used in general. Furthermore, linear approximators are efficiently implemented on both computation and data in practice compared to nonlinear approximators [1]. For this reason, this article focuses on the design and analysis of TD learning, where the value function is parameterized by linearly combining the preselected bias functions. According to the update rule of the vanilla TD learning, the weight parameters can be estimated by an iteration process. Albeit the iteration process is simple, the rigorous analysis of convergence performance remains a challenging task [8]. Toward this direction, the asymptotic performance of TD learning with a linear approximation is established in [8], [9], [10], and [11]. Nevertheless, the nonasymptotic performance analysis has recently received increasing interest because massive data examples need to be handled in artificial intelligence, signal processing, and control tasks. Moreover, the statistical efficiency of algorithms can be better understood in terms of nonasymptotic analysis. Compared to the asymptotic analysis of TD learning, its nonasymptotic analysis faces more challenges in particular [12]. For example, the bias and correlation are introduced in the update of TD, which cannot correspond to stochastic gradient ascent. Despite these challenges, the nonasymptotic analysis of TD learning was also developed in recent years [13], [14], [15], [16], [17], [18].

Manuscript received 18 August 2023; revised 19 November 2023; accepted 19 March 2024. Date of publication 26 April 2024; date of current version 19 July 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61976243 and Grant 62033010; in part by the Science and Technology Development Plan of Henan Province of China under Grant 231100220600 and Grant 231111222600; in part by the Leading Talents of Science and Technology in the Central Plain of China under Grant 224200510004; in part by the Scientific and Technological Innovation Teams and Talents of Colleges and Universities in Henan Province of China under Grant 24IRTSTHN022 and Grant 22HASTIT014; in part by the Basic Research Projects in the University of Henan Province of China under Grant 23ZX003; and in part by the China Postdoctoral Science Foundation of China under Grant 2021M690914. This article was recommended by Associate Editor Q. Wei. *(Corresponding author: Mingchuan Zhang.)*

Junlong Zhu, Tao Mao, Mingchuan Zhang, and Qingtao Wu are with the School of Information Engineering, Henan University of Science and Technology, Luoyang 471023, China (e-mail: jlzhu@haust.edu.cn; maotao@stu.haust.edu.cn; zhang_mch@haust.edu.cn; wqt8921@haust.edu.cn).

Quanbo Ge is with the School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: geqb@nuist.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TSMC.2024.3382986.

Digital Object Identifier 10.1109/TSMC.2024.3382986

Albeit the nonasymptotic analysis of TD learning has made great progress, the aforementioned works were made for the centralized setting. When dealing with some tasks in the multiagent paradigm, however, this setting suffers from some limitations, including the communication bottleneck problem in wireless networks, the privacy and secrecy problem in sensitive applications, and the robust problem [19]. Moreover, the central coordinator may not even exist in many practical applications. For these reasons, decentralized TD learning methods were proposed recently in multiagent RL (MARL) environments [12], [20], [21], [22], [23], [24], [25], which suppose that each agent only knows its own local information and can communicate with its neighbors over a multiagent network. Moreover, the optimal weight parameter is sought in a cooperative manner without any central coordinator. Specifically, Mathkar and Borkar [20] analyzed the asymptotic performance of decentralized TD(0) learning by using the ordinary differential equation (ODE) approach. Under the independently and identically distributed (i.i.d.) assumption, Doan et al. [21] rigorously analyzed the nonasymptotic (i.e., finite-time) performance of decentralized TD(0) learning. Nevertheless, it is hard to satisfy the i.i.d. assumption in practice [14]. For this reason, the nonasymptotic analysis of decentralized TD(0) learning was established in [12] and [22] under the Markovian model, which is a more realistic scenario than i.i.d. Although TD(0) learning has a faster convergence rate, its approximation capability is weaker. Moreover, the more general TD learning, referred to as TD(λ) with $\lambda \in [0, 1]$, may learn more efficiently [1]. For this reason, the finite-time performance of TD(λ) learning was also established under the Markovian model in [24] and [25].

However, the above-mentioned decentralized TD learning may suffer from slow convergence because their convergence rates are sensitive to the selection of learning rates [16], [26]. In these decentralized TD learning with linear function approximation, the update rules are similar to stochastic gradient optimization, which leads to poor convergence performance because the gradient is scaled uniformly in all updated directions. To address this issue in SGD, various ADAM-type methods were recently proposed in stochastic optimization [27], [28], where the gradient is adaptively scaled by adjusting dynamically the learning rate. Due to their superior performance, ADAM-type methods have been usually used empirically in RL [29]. Motivated by the empirical success, TD learning incorporated ADAM-type updates and its convergence guarantee has been recently developed in [16] and [17]. Despite these theoretical efforts, the aforementioned ADAM-type TD learning algorithms were suitable for single-agent RL. Very recently, Zhu et al. [30] proposed a distributed adaptive TD(0) learning algorithm and proved its nonasymptotic performance. However, decentralized adaptive variants of TD(λ) learning with linear function approximation have rarely been investigated for MARL to our knowledge. Thereby, there still exists an open question:

Can provable decentralized adaptive TD(λ) methods for a general λ be developed to accelerate the decentralized vanilla TD(λ) learning as AMSGRAD *in [28]?*

To answer this question, we propose a decentralized adaptive TD(λ) learning algorithm by incorporating the adaptive gradient method into the decentralized TD(λ) learning. Nevertheless, the design and analysis of decentralized adaptive TD(λ) learning algorithm is highly nontrivial due to the following reasons: 1) the updates of TD(λ) learning do not follow the gradient ascent direction of stochastic optimization; 2) the Markovian model naturally gives rise to the biased "gradient" in the update process; 3) adaptive methods include complex update rules; and 4) the effort of the communication protocol and the adaptive learning rates is interactional in the update process. Despite these challenges, we also rigorously analyze the nonasymptotic performance of the proposed algorithm under the Markovian model. To obtain the finite-time convergence performance, we also choose a multistep Lyapunov function [18], [22], [31] to deal with the biased gradient introduced by the Markovian model. To the best of our knowledge, the proposed algorithm in this article is the first decentralized adaptive TD(λ) learning. In short, our contributions are summarized as follows.

1) We develop a decentralized adaptive TD learning algorithm with a general λ, referred to as **D-AMSTD(λ)**, where AMSGRAD is incorporated into the decentralized variant of TD(λ) learning.
2) We also establish the finite-time performance bound of **D-AMSTD(λ)** under the Markovian model, which is more real scenario than the i.i.d. model, i.e., **D-AMSTD(λ)** converges to a neighborhood of the optimal weight at a linear rate.
3) We verify the efficacy of **D-AMSTD(λ)** by a variety of experiments. The results show that the convergence performance of **D-AMSTD(λ)** outperforms existing decentralized TD learning algorithms with nonasymptotic convergence guarantees.

Organization: Section II briefly reviews some related works with respect to TD learning. Some requisite backgrounds are provided in Section III. In Section IV, we develop a decentralized adaptive variant of TD(λ) learning. Moreover, some standard assumptions are also made in Section IV for the performance analysis. In Section V, we present the main results of this article under the Markovian observation model. We provide the finite-time analysis of the proposed algorithm in Section VI. Meanwhile, we evaluate the effectiveness of the proposed algorithm by various experiments in Section VII. Finally, Section VIII concludes this article.

Notation: In this article, all vectors are column vectors. The d-dimensional real space is denoted by $\mathbb{R}^d$. The real matrix with size $m \times n$ is denoted by $\mathbb{R}^{m\times n}$. We use $\mathbb{1}_m$ to denote the m-dimensional vector with all ones. The transpose of a vector or matrix is represented by $(\cdot)^\top$. The ℓ_2-norm and ℓ_∞-norm of vectors are denoted by $\|\cdot\|$ and $\|\cdot\|_\infty$, respectively. The notation $\|\mathbf{M}\|_{1,1} \coloneqq \sum_{i,j=1}^{n} |m_{ij}|$ with a matrix $\mathbf{M} = [m_{ij}] \in \mathbb{R}^{n\times n}$. The element-wise product and division are designated as $x \odot y$ and x/y for any vectors x and y, respectively. Besides, the element-wise square root of a vector x is represented by $\sqrt{x}$. For any two vectors, we use $\min(\cdot,\cdot)$ and $\max(\cdot,\cdot)$ to represent the minimum and maximum, respectively.

II. Related Work

TD learning, which is a recursive method for estimating the value function, was originally proposed in [2]. Albeit the implementation of vanilla TD learning is simple and efficient, its performance analysis still requires sophisticated tools. Toward the theoretical direction, some performance analysis results of TD learning have been presented recently. Specifically, Jaakkola et al. [32] utilized stochastic approximation methods to establish the first convergence result of TD learning, which employs a tabular representation for the value function. When the state space is large or infinite, however, the tabular-based TD learning becomes intractable due to the problem of the curse of dimensionality. For this reason, Tsitsiklis and Roy [8] rigorously analyzed the asymptotic convergence of TD learning with linear function approximation. Although the asymptotic performance is revealed, the nonasymptotic performance, which is very important in practice, is barely known. Toward this direction, the first result is offered by [33]. Nonetheless, Lakshminarayanan and Szepesvári [34] pointed out that there are several serious errors in the theoretical analysis. Thereafter, Dalal et al. [14] established the nonasymptotic (i.e., finite-time) performance of TD learning, where they assumed that the observation model is i.i.d. However, this assumption cannot be satisfied in practice. In order to mitigate this assumption, Bhandari et al. [13] offered the finite-time convergence rate of TD learning under the Markovian observation model, in which the projection step is also introduced in this work. In practice, however, it is hard to implement the projection step. To eschew the projection step, the finite-time convergence rate was established by leveraging the Lyapunov theory in [15], [18], and [31]. Nonetheless, the original TD update, which is very sensitive to the selection of step-sizes, is used in the above-mentioned works. For this reason, Sun et al. [16] investigated the adaptive TD learning algorithms and established their finite-time performance, where AdaGrad [35] is incorporated into the original TD learning. Incorporating AMSGrad [28] into the vanilla TD learning was investigated in [17]. Moreover, the finite-time performance was also established for the different choices of step-sizes. However, all the aforementioned works were made for single-agent RL.

The multiagent paradigm, which includes multiple agents, is also widely applied within engineering, whereas single-agent RL may be unfit for solving MARL problems. For this reason, MARL has received significant interest in recent years. Specifically, Mathkar and Borkar [20] offered the first asymptotic analysis of the distributed TD(0) learning with gossip, in which the ODE-based method is utilized in their analysis. Thereafter, the finite-time convergence rate of distributed TD(0) learning was also established by [21] with i.i.d. observation models. To alleviate this strong assumption, Sun et al. [22] investigated the finite-time performance of decentralized TD(0) learning with the Markovian observation model, where a multistep Lyapunov method is used to control the bias of the gradient. Furthermore, Wang et al. [12] presented a decentralized TD(0) tracker to improve the nonasymptotic performance. Meanwhile, Lin and Ling [23] presented decentralized TD(0) learning methods by leveraging the gradient tracking technique, which is often used to accelerate the rate of convergence in distributed optimization [36], [37], [38]. This work does not establish the corresponding finite-time error bounds. Besides, Zhu et al. [30] designed and analyzed the adaptive variant of decentralized TD(0) learning. The aforementioned efforts are made for decentralized TD(0) learning, however, the more general TD(λ) learning may learn more efficiently in practice [1]. For this reason, Doan et al. [24] studied the decentralized variant of TD(λ) learning. Moreover, its finite-time convergence performance was also established for constant and diminishing step-sizes, respectively. Besides, a Byzantine-resilient decentralized TD(λ) learning and its finite-time performance were also investigated in [25]. The above-mentioned works utilize the original TD(λ) update, however, how to design and analyze the decentralized adaptive variants of TD(λ) learning remains an open problem. Indeed, the decentralized adaptive gradient algorithms have been recently proposed for the training of distributed machine learning models. More recently, decentralized optimization algorithms were proposed in [39] and [40]. Furthermore, the convergence performance was also analyzed rigorously for convex and submodular objective functions. For nonconvex objective functions, Chen et al. [41] first pointed out the convergence issue of DADAM [42] for nonconvex optimization. To address this issue, a general decentralized adaptive algorithmic framework was offered in this work. Moreover, the rate of convergence was also established rigorously. However, the decentralized adaptive variants of TD(λ) learning have barely been studied to the best of our knowledge. For this reason, we attempt to bridge this gap between ADAM-type updates and decentralized TD(λ) learning in MARL.

III. Preliminaries

This section provides some background for the Markov decision process (MDP). Thereafter, we review the policy evaluation problem. To solve the problem, we also review centralized TD learning, which is one of the most dominant algorithms for policy evaluation.

A. Markov Decision Process and Policy Evaluation

An MDP is characterized as a tuple $(\mathcal{S}, \mathcal{U}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a state space, $\mathcal{U}$ is an action space, $\mathcal{P}$ denotes a matrix of transition probability, and γ denotes a discount fact with $0 < \gamma < 1$. Thus, the probability of transitioning to state $s' \in \mathcal{S}$ under the state $s \in \mathcal{S}$ and the action $u \in \mathcal{U}$ is described by $\mathcal{P}(s', s, u) = \Pr(s'|s, u)$. Moreover, $\sum_{s' \in \mathcal{S}} \Pr(s'|s, u) = 1$. Meanwhile, $\mathcal{R}(s, u, s')$ stands for the transition reward. In this MDP, the value function (accumulative reward) $J_\omega : \mathcal{S} \mapsto \mathbb{R}$ under a policy $\omega : \mathcal{S} \mapsto \mathcal{U}$ is defined as

$$J_\omega(s) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}\left(s_t, u_t, s_t'\right) \middle| s_0 = s\right] \tag{1}$$

where s_0 denotes the initial state. Furthermore, the value function can be rewritten as [1]

$$J_\omega(s) = \sum_{u \in \mathcal{U}} \omega(u|s) \sum_{s' \in \mathcal{S}} \Pr\left(s'|s, u\right)\left[\mathcal{R}\left(s, u, s'\right) + \gamma J_\omega\left(s'\right)\right]. \tag{2}$$

When

$$\mathcal{R}(s) \coloneqq \sum_{s'\in\mathcal{S}}\sum_{u\in\mathcal{U}} \omega(u|s)\Pr(s'|s,u)\mathcal{R}(s,u,s')$$

and

$$\mathcal{P}(s,s') \coloneqq \sum_{u\in\mathcal{U}} \omega(u|s)\Pr(s'|s,u)$$

were known, the value function J_ω can be obtained by solving (2). For brevity, the subscript ω is neglected henceforth in the notations because it is fixed in this article.

B. Centralized Temporal-Difference Learning

The state space $\mathcal{S}$ is too large to compute the value function directly by solving (2) in practice. To mitigate the issue, the value function J is approximated by a low-dimensional function $\tilde{J}$. In particular, we consider linear function approximation, i.e.,

$$J(s) \approx \tilde{J}(s,\theta) \coloneqq \sum_{p=1}^{d} \theta_p \phi_p(s) \tag{3}$$

where $d \ll |\mathcal{S}|$ and $\phi_p(s) \in \mathbb{R}$ denotes a feature value and $\theta_p \in \mathbb{R}$ denotes a weight. For ease of exposition, define

$$\Phi \coloneqq \begin{pmatrix} | & & | \\ \phi_1 & \cdots & \phi_d \\ | & & | \end{pmatrix} = \begin{pmatrix} — & \phi(1)^\top & — \\ \vdots & \vdots & \vdots \\ — & \phi(|\mathcal{S}|)^\top & — \end{pmatrix} \in \mathbb{R}^{|\mathcal{S}|\times d}$$

where $\phi(s) = (\phi_1(s),\ldots,\phi_d(s))^\top$. Moreover, let

$$\tilde{J}(\theta) \coloneqq \left(\tilde{J}(1,\theta)\ldots,\tilde{J}(|\mathcal{S}|,\theta)\right)^\top$$

with $\theta \coloneqq (\theta_1,\ldots,\theta_d)^\top$, then (3) can be written as

$$\tilde{J}(\theta) = \Phi\theta. \tag{4}$$

To find the optimal approximation $\tilde{J}$ of J, it has necessitated the design of TD learning algorithms that can find effectively the optimal weight θ^*. In the centralized TD learning, the weight θ_t is updated as follows:

$$\theta_{t+1} = \theta_t + \alpha_t d_t \nabla_\theta \tilde{J}(s_t,\theta_t) \tag{5}$$

where $\alpha_t > 0$ is the learning rate, d_t denotes the TD at iteration t and is defined as

$$d_t \coloneqq r_t + \gamma\tilde{J}(s_{t+1},\theta_t) - \tilde{J}(s_t,\theta_t). \tag{6}$$

Here, $r_t = \mathcal{R}(s_{t+1},u_t,s_t)$. The gradient $\nabla_\theta\tilde{J}(s_t,\theta_t)$ can be computed by

$$\nabla_\theta\tilde{J}(s_t,\theta_t) = z_t \coloneqq \sum_{\tau=0}^{t}(\gamma\lambda)^{t-\tau}\phi(s_\tau) \tag{7}$$

where $\lambda \in [0,1]$ is a constant. For brevity, define

$$\begin{aligned} g(\theta_t,\zeta_t) &\coloneqq d_t\nabla_\theta\tilde{J}(s_t,\theta_t) \\ &= z_t(\gamma\phi(s_{t+1}) - \phi(s_t))^\top\theta_t + r_t z_t \end{aligned} \tag{8}$$

where ζ_t denotes the randomness of the tuple (s_t,r_t,s_{t+1}). Using (5) yields

$$\theta_{t+1} = \theta_t + \alpha_t g(\theta_t,\zeta_t). \tag{9}$$

For ease of exposition, let $Q_t \coloneqq z_t(\gamma\phi(s_{t+1}) - \phi(s_t))^\top$ and $b_t \coloneqq r_t z_t$, then (8) can be rewritten as

$$g(\theta_t,\zeta_t) = Q_t\theta_t + b_t. \tag{10}$$

Let $\pi \in \mathbb{R}^{|\mathcal{S}|}$ denote the unique stationary distribution of the Markov chain if it is ergodic. Moreover, let $\mathbf{D} \coloneqq \operatorname{diag}(\pi(1),\ldots,\pi(|\mathcal{S}|))$ and $r'(s) = \sum_{s'\in\mathcal{S}}\mathcal{P}(s,s')\mathcal{R}(s,s')$, where $\mathcal{R}(s,s') = \sum_{u\in\mathcal{U}}\omega(u|s)\Pr(s'|s,u)\mathcal{R}(s',u,s)$, then the following relation:

$$\bar{Q} \coloneqq \lim_{t\to\infty}\mathbb{E}[Q_t] = \Phi^\top\mathbf{D}(\mathbf{U}-\mathbf{I})\Phi \tag{11}$$

and

$$\bar{b} \coloneqq \lim_{t\to\infty}\mathbb{E}[b_t] = \Phi^\top\mathbf{D}\sum_{t=0}^{\infty}(\gamma\lambda\mathbf{P})^t r' \tag{12}$$

hold, where $\mathbf{P}$ stands for the transition matrix, $r' \coloneqq (r'(1),\ldots,r'(|\mathcal{S}|))^\top$, and

$$\mathbf{U} = (1-\lambda)\sum_{t=0}^{\infty}\lambda^t(\gamma\mathbf{P})^{t+1}.$$

Thus, we obtain

$$\bar{g}(\theta) \coloneqq \bar{Q}\theta + \bar{b}. \tag{13}$$

Further, Tsitsiklis and Roy [8] showed that $\bar{g}(\theta^*) = 0$.

IV. Multiagent Reinforcement Learning, Decentralized Adaptive TD(λ) Learning, and Assumptions

This section first introduces the policy evaluation problem in MARL. To solve this problem, we propose a decentralized adaptive TD(λ) learning algorithm over networks. Furthermore, we also make some standard assumptions for analyzing the nonasymptotic performance of the proposed algorithm.

A. Multiagent Reinforcement Learning

In this article, a network with N agents is denoted by $\mathcal{G} \coloneqq (\mathcal{V},\mathcal{E})$, where $\mathcal{V} \coloneqq \{1,\ldots,N\}$ is the set of agents and $\mathcal{E}$ is the set of edges. The set of neighbors of agent i is denoted by $\mathcal{N}_i \coloneqq \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\}$. Furthermore, MARL can be modeled by a multiagent MDP, which is characterized by a tuple $(\mathcal{S},\{\mathcal{U}_i\}_{i=1}^N,\mathcal{P},\{\mathcal{R}_i\}_{i=1}^N,\gamma,\mathcal{G})$. In the multiagent MDP, all agents can observe the state space $\mathcal{S}$, each agent $i \in \mathcal{V}$ only utilizes its own action space $\mathcal{U}_i$ and observes its own reward function $\mathcal{R}_i$, respectively.

Under a given policy ω_i of agent $i \in \mathcal{V}$, the action $u_{i,t} \in \mathcal{U}_i$ is selected by agent i, then the state s_t transits to s_{t+1} at time t. Meanwhile, agent i can reveal a reward $r_{i,t} \coloneqq \mathcal{R}_i(s_t,s_{t+1})$. The value function $J_{\mathcal{D}}(s)$ can be defined as

$$J_{\mathcal{D}}(s) \coloneqq \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{\infty}\gamma^t\mathcal{R}_i(s_t,u_{i,t},s_{t+1})\middle| s_0 = s\right]. \tag{14}$$

Furthermore, the value function $J_{\mathcal{D}}(s)$ is also the solution of the multiagent Bellman equation, i.e.,

$$J_{\mathcal{D}}(s) = \frac{1}{N}\sum_{i=1}^{N}\sum_{u_i\in\mathcal{U}_i}\omega(u_i|s) \times \sum_{s'\in\mathcal{S}}\Pr(s'|s,u_i)\big[\mathcal{R}_i(s,u_i,s') + \gamma J_{\mathcal{G}}(s')\big]. \quad (15)$$

Similar to the centralized RL, when $\mathcal{S}$ is very large, which leads to the problem of curse of dimensionality, we need to seek the approximation $\tilde{J}_{\mathcal{D}}(s)$ of the value function $J_{\mathcal{D}}(s)$. Toward this end, this article considers the case that linear approximation is used to approximate the value function $\tilde{J}_{\mathcal{D}}(s)$, i.e., this approximation is presented in (3) or (4). Define

$$\bar{b}_i := \Phi^\top \mathbf{D}\sum_{t=0}^{\infty}(\gamma\lambda\mathbf{P})^t r_i'$$

where $r_i' := (r_i'(1),\ldots,r_i'(|\mathcal{S}|))^\top$ with $r_i'(s) = \sum_{s'\in\mathcal{S}}\mathcal{P}(s,s')\mathcal{R}_i(s,s')$. Then, the optimal weight θ^* also satisfies the following equality [21]:

$$\bar{Q}\theta^* + \bar{b}_{\mathcal{D}} = 0 \quad (16)$$

where the matrix $\bar{Q}$ is negative definite and

$$\bar{b}_{\mathcal{D}} := \frac{1}{N}\sum_{i=1}^{N}\bar{b}_i.$$

In this article, we assume that each agent can communicate with its neighbors. Thus, the objective is to find the optimal weight θ^* in a cooperative way without any central agent.

B. Decentralized Adaptive TD(λ) Learning

In order to find cooperatively the optimal weight θ^* over the network $\mathcal{G}$, we propose a decentralized adaptive TD(λ) learning algorithm by integrating decentralized optimization algorithm and AMSGRAD into TD(λ) learning, which is called **D-AMSTD(λ)**. The details are summarized in Algorithm 1. More specifically, each agent $i\in\mathcal{V}$ computes local gradient g_i by observing (s_t, r_t, s_{t+1}), i.e.,

$$g_i(\theta_{i,t},\zeta_t) = z_t(\gamma\phi(s_{t+1}) - \phi(s_t))^\top\theta_{i,t} + r_{i,t}z_t. \quad (17)$$

The first and second moments are updated as follows:

$$m_{i,t} = \beta_1 m_{i,t-1} + (1-\beta_1)g_i(\theta_{i,t},\zeta_t) \quad (18)$$

and

$$v_{i,t} = \beta_2 v_{i,t-1} + (1-\beta_2)g_i(\theta_{i,t},\zeta_t)\odot g_i(\theta_{i,t},\zeta_t) \quad (19)$$

where $\beta_1,\beta_2\in[0,1)$ are constants. To ensure the convergence, we also define

$$\tilde{v}_{i,t} := \max(\tilde{v}_{i,t-1}, v_{i,t}). \quad (20)$$

Moreover, each agent $i\in\mathcal{V}$ performs a consensus step, where each agent exchanges its local estimate with its neighbors, i.e.,

$$\theta_{i,t+\frac{1}{2}} = \sum_{j\in\mathcal{N}_i} a_{ij}\theta_{j,t} \quad (21)$$

Algorithm 1 Decentralized Adaptive TD(λ) Learning Over Networks (**D-AMSTD(λ)**)

Input: The number of agents N; doubly stochastic matrix $A = [a_{ij}]\in\mathbb{R}^{N\times N}$; feature matrix Φ; learning rate α_t.
Output: $\{\theta_{i,t}\}$ for $i\in\mathcal{V}$ and $t\geq 1$.

Initialize: $\theta_{i,0}=\theta_c$, $m_{i,0}=0$, $w_{i,\frac{1}{2}} = v_{i,0} = \tilde{v}_{i,0} = \epsilon\mathbb{1}_d$ for all $i\in\mathcal{V}$.
for $t = 1, 2, \ldots$ **do**
  for $i = 1,\ldots,N$ **do**
    $g_i(\theta_{i,t},\zeta_t) = z_t(\gamma\phi(s_{t+1}) - \phi(s_t))^\top\theta_{i,t} + r_{i,t}z_t$;
    $m_{i,t} = \beta_1 m_{i,t-1} + (1-\beta_1)g_i(\theta_{i,t},\zeta_t)$;
    $v_{i,t} = \beta_2 v_{i,t-1} + (1-\beta_2)g_i(\theta_{i,t},\zeta_t)\odot g_i(\theta_{i,t},\zeta_t)$;
    $\tilde{v}_{i,t} = \max(\tilde{v}_{i,t-1}, v_{i,t})$;
    $\theta_{i,t+\frac{1}{2}} = \sum_{j\in\mathcal{N}_i} a_{ij}\theta_{j,t}$;
    $\widehat{w}_{i,t} = \sum_{j\in\mathcal{N}_i} a_{ij}\widehat{w}_{i,t-\frac{1}{2}}$;
    $w_{i,t} = \max(\widehat{w}_{i,t},\epsilon)$;
    $\theta_{i,t+1} = \theta_{i,t+\frac{1}{2}} + \alpha_t\frac{m_{i,t}}{\sqrt{w_{i,t}}}$;
    $z_{t+1} = \gamma\lambda z_t + \phi(s_{t+1})$;
    $\widehat{w}_{i,t+\frac{1}{2}} = \min(\widehat{w}_{i,t} + \tilde{v}_{i,t} - \tilde{v}_{i,t-1}, G_\infty^2)$.
  end for
end for

where $a_{ij}\geq 0$ denotes a weight. Then, each agent $i\in\mathcal{V}$ updates $\widehat{w}_{i,t}$ by performing a consensus step, i.e.,

$$\widehat{w}_{i,t} = \sum_{j\in\mathcal{N}_i} a_{ij}\widehat{w}_{i,t-\frac{1}{2}}. \quad (22)$$

Similar to vanilla ADAM for numerical stability, we also define

$$w_{i,t} := \max(\widehat{w}_{i,t},\epsilon)$$

where ϵ is a positive constant. Finally, the parameter $\theta_{i,t}$ is given by

$$\theta_{i,t+1} = \theta_{i,t+\frac{1}{2}} + \alpha_t\frac{m_{i,t}}{\sqrt{w_{i,t}}}. \quad (23)$$

Besides, the estimate of the second moment is updated as

$$\widehat{w}_{i,t+\frac{1}{2}} = \min\Big(\widehat{w}_{i,t} + \tilde{v}_{i,t} - \tilde{v}_{i,t-1}, G_\infty^2\Big) \quad (24)$$

where G_∞ is a constant. In this article, we assume that $\epsilon\leq G_\infty^2$ since $\epsilon>0$ can be arbitrarily small. For convenience, we also introduce the following matrices:

$$\Theta_t := \begin{bmatrix}\theta_{1,t}^\top\\ \theta_{2,t}^\top\\ \vdots\\ \theta_{N,t}^\top\end{bmatrix}\in\mathbb{R}^{N\times d},\quad M_t := \begin{bmatrix}m_{1,t}^\top\\ m_{2,t}^\top\\ \vdots\\ m_{N,t}^\top\end{bmatrix}\in\mathbb{R}^{N\times d}$$

$$\Xi(\Theta_t,\zeta_t) := \begin{bmatrix}g_1(\theta_{1,t},\zeta_t)^\top\\ g_2(\theta_{2,t},\zeta_t)^\top\\ \vdots\\ g_N(\theta_{N,t},\zeta_t)^\top\end{bmatrix}\in\mathbb{R}^{N\times d}$$

$$V_t := \begin{bmatrix}v_{1,t}^\top\\ v_{2,t}^\top\\ \vdots\\ v_{N,t}^\top\end{bmatrix}\in\mathbb{R}^{N\times d},\quad \tilde{V}_t := \begin{bmatrix}\tilde{v}_{1,t}^\top\\ \tilde{v}_{2,t}^\top\\ \vdots\\ \tilde{v}_{N,t}^\top\end{bmatrix}\in\mathbb{R}^{N\times d}$$

$$W_t \coloneqq \begin{bmatrix} w_{1,t}^\top \\ w_{2,t}^\top \\ \vdots \\ w_{N,t}^\top \end{bmatrix} \in \mathbb{R}^{N\times d}, \text{ and } \widehat{W}_t \coloneqq \begin{bmatrix} \widehat{w}_{1,t}^\top \\ \widehat{w}_{2,t}^\top \\ \vdots \\ \widehat{w}_{N,t}^\top \end{bmatrix} \in \mathbb{R}^{N\times d}.$$

Then, using (21) and (23) yields

$$\Theta_{t+1} = A\Theta_t + \alpha_t \frac{M_t}{\sqrt{W_t}} \tag{25}$$

where $A \coloneqq [a_{ij}] \in \mathbb{R}^{N\times N}$. In addition, we also define some average variables as follows:

$$\overline{\theta}_t \coloneqq \frac{1}{N}\sum_{i=1}^N \theta_{i,t}, \ \overline{w}_t \coloneqq \frac{1}{N}\sum_{i=1}^N w_{i,t}, \text{ and } \overline{\widehat{w}}_t \coloneqq \frac{1}{N}\sum_{i=1}^N \widehat{w}_{i,t}.$$

By the definition of Θ_t, we have

$$\overline{\theta}_t = \frac{1}{N}\Theta_t^\top \mathbb{1}_N. \tag{26}$$

C. Some Standard Assumptions

In order to ensure that **D-AMSTD(λ)** is convergent, we also make some standard assumptions, which can be found in the literature of optimization algorithms and TD learning. First, we present the following assumption that the domain is bounded, which is also made in [17], [27], and [28].

Assumption 1: There exists a positive constant D_∞ such that $\|\theta_{i,t} - \theta_{j,t}\| \le D_\infty$ for $i, j \in \mathcal{V}$ and $t \ge 0$.

We also need to make some assumptions with respect to the reward function $\mathcal{R}_i(s, s')$ and the feature matrix Φ, respectively. These assumptions can be also found in [13] and [16].

Assumption 2: For all $i \in \mathcal{V}$, suppose that $|\mathcal{R}_i(s, s')| \le r_{\max}$ with $r_{\max} > 0$ for all $s, s' \in \mathcal{S}$.

For simplicity, the uniform boundedness is adopted in this article. We can use nonuniform boundedness to replace Assumption 2, which was also made in [21].

Assumption 3: Suppose that Φ is full column rank. For all $s \in \mathcal{S}$, $\|\phi(s)\| \le 1$.

Assumption 3 means that the vectors $\{\phi(s)\}$ are linearly independent and normalized. Furthermore, because $d \ll |\mathcal{S}|$, we can also ensure that Assumption 3 holds. Furthermore, following Assumptions 2 and 3, we have:

$$\begin{aligned} \|Q_t\| &= \left\| z_t(\gamma\phi(s_{t+1}) - \phi(s_t))^\top \right\| \\ &\le (1+\gamma)\sum_{\tau=0}^t (\gamma\lambda)^{t-\tau} \\ &\le \frac{1+\gamma}{1-\gamma\lambda} \end{aligned} \tag{27}$$

and

$$\begin{aligned} \|b_{i,t}\| &= \|z_t r_{i,t}\| \\ &\le r_{\max}\sum_{\tau=0}^t (\gamma\lambda)^{t-\tau} \\ &\le \frac{r_{\max}}{1-\gamma\lambda}. \end{aligned} \tag{28}$$

Thus, by the definitions of $\bar{Q}$ and $\bar{b}_{\mathcal{D}}$, we can obtain $\|\bar{Q}\| \le (1+\gamma)/(1-\gamma\lambda)$ and $\|\bar{b}_{\mathcal{D}}\| \le r_{\max}/(1-\gamma\lambda)$, respectively.

Next, we make the standard assumption on the Markov chain, which is also provided in TD learning [8], [12], [22].

Assumption 4: Assume that the Markov chain with $\mathcal{P}$ is irreducible and aperiodic.

For two probability measures χ_1 and χ_2, we use $d_{TV}(\chi_1, \chi_2)$ to denote the total-variation norm, which is defined as

$$d_{TV}(\chi_1, \chi_2) \coloneqq \frac{1}{2}\sum_{s\in\mathcal{S}} |\chi_1(s) - \chi_2(s)|. \tag{29}$$

Then, using Assumption 4 yields [43]

$$\sup_{s\in\mathcal{S}} d_{\text{TV}}(\Pr(s_t \in \cdot | s_0 = s), \pi) \le \varrho_0 \varkappa^t \tag{30}$$

where $\varrho_0 > 0$ and $0 < \varkappa < 1$ are constants, and $s_t \in \cdot$ means that s_t is an element of the state space $\mathcal{S}$ or its subset.

Finally, we also present some standard assumptions on graphs, which are made in [44] and [45].

Assumption 5: The graph $\mathcal{G}$ is connected and undirected.

Assumption 5 ensures that the information can be exchanged among agents directly or indirectly.

Assumption 6: The matrix A is doubly stochastic. Moreover, if $(i, j) \in \mathcal{E}$, then $a_{ij} > 0$; otherwise $a_{ij} = 0$.

Assumption 6 implies that $\sum_{i=1}^N a_{ij} = \sum_{j=1}^N a_{ij} = 1$. Besides, the ith largest eigenvalue is denoted by σ_i. Furthermore, this article also defines $\sigma \coloneqq \max(|\sigma_2|, |\sigma_N|)$.

V. Main Results

This section establishes the convergence behaviors of **D-AMSTD(λ)** when $\alpha_t = \alpha$ under the Markovian model, which is a more realistic case than the i.i.d. model. Albeit we focus on constant learning rates, the nonasymptotic analytical methods can be also extended to diminishing learning rates. To obtain the nonasymptotic bounds, we first estimate the mean-squared error between the average weight $\overline{\theta}_t$ and the optimal weight θ^*. For this reason, we define some variables as

$$\begin{aligned} \Psi_1(\alpha, T) \coloneqq\ & \frac{8T^4\alpha^3(1+\gamma)^4(1-\beta_1)^4}{\epsilon^3(1-\gamma\lambda)^4}(1+\upsilon)^{2T-4} \\ & + 32T^2\frac{\alpha(1+\gamma)^2(1-\beta_1)^2}{(1-\gamma\lambda)^2} \\ & + \frac{2T^2\alpha(1+\gamma)^2(1-\beta_1)^2}{\epsilon^{3/2}(1-\gamma\lambda)^2}(1+\upsilon)^{T-2} \\ & + 4T(1-\beta_1)\Gamma(T;t) \\ \Psi_2(\alpha, T) \coloneqq\ & \frac{T^2\alpha(1+\gamma)^2(1-\beta_1)^2}{2\epsilon^{3/2}(1-\gamma\lambda)^2}(1+\upsilon)^{T-2}\cdot\|\theta^*\|^2 \\ & + \frac{8T^4\alpha^3(1+\gamma)^4(1-\beta_1)^4}{\epsilon^3(1-\gamma\lambda)^4}(1+\upsilon)^{2T-4}\cdot\|\theta^*\|^2 \\ & + 32T^2\frac{\alpha(1+\gamma)^2(1-\beta_1)^2}{(1-\gamma\lambda)^2}\cdot\|\theta^*\|^2 + \frac{\beta_1^2T^2G_\infty^2}{\mu'\epsilon} \\ & + \frac{T^2\alpha(1+\gamma)^2(1-\beta_1)^2}{2\epsilon^{3/2}(1-\gamma\lambda)^2}(1+\upsilon)^{T-2} \\ & \times \frac{1}{(1+\gamma)^2}\left(\frac{(1-\gamma\lambda)G_\infty}{1-\beta_1} + r_{\max}\right)^2 \\ & + \frac{(1-\beta_1)^2TG_\infty^2}{2\eta'\epsilon^2\sqrt{N}}\frac{\sigma}{1-\sigma}\sum_{k=0}^{t+T-2}\left\|\widetilde{V}_k - \widetilde{V}_{k-1}\right\|_{1,1} \end{aligned}$$

$$+\frac{1}{2}T(1-\beta_1)\Gamma(T;t)+16T^2\frac{\alpha r_{\max}^2(1-\beta_1)^2}{(1-\gamma\lambda)^2}+\frac{4T^4\alpha^3(1+\gamma)^2(1-\beta_1)^4r_{\max}^2}{\epsilon^3(1-\gamma\lambda)^4}(1+\upsilon)^{2T-4}+\frac{2dT\alpha\sqrt{N}G_\infty^4(1-\beta_1)^2}{\epsilon^2}\frac{\sigma}{1-\sigma}+\frac{2\alpha T^2\beta_1^2G_\infty^2}{\epsilon}$$

and

$$\Psi_2'(\alpha,T)\coloneqq\frac{T^2\alpha(1+\gamma)^2(1-\beta_1)^2}{2\epsilon^{3/2}(1-\gamma\lambda)^2}(1+\upsilon)^{T-2}\cdot\left\|\theta^*\right\|^2+\frac{8T^4\alpha^3(1+\gamma)^4(1-\beta_1)^4}{\epsilon^3(1-\gamma\lambda)^4}(1+\upsilon)^{2T-4}\cdot\left\|\theta^*\right\|^2+32T^2\frac{\alpha(1+\gamma)^2(1-\beta_1)^2}{(1-\gamma\lambda)^2}\cdot\left\|\theta^*\right\|^2+\frac{\beta_1^2T^2G_\infty^2}{\mu'\epsilon}+\frac{T^2\alpha(1+\gamma)^2(1-\beta_1)^2}{2\epsilon^{3/2}(1-\gamma\lambda)^2}(1+\upsilon)^{T-2}\times\frac{1}{(1+\gamma)^2}\left(\frac{(1-\gamma\lambda)G_\infty}{1-\beta_1}+r_{\max}\right)^2+\frac{(1-\beta_1)^2TG_\infty^2}{2\eta'\epsilon^2\sqrt{N}}\frac{\sigma}{1-\sigma}\sum_{k=0}^{t+T-2}\left\|\widetilde{V}_k-\widetilde{V}_{k-1}\right\|_{1,1}+\frac{1}{2}T(1-\beta_1)\alpha\Gamma(T)+16T^2\frac{\alpha r_{\max}^2(1-\beta_1)^2}{(1-\gamma\lambda)^2}+\frac{4T^4\alpha^3(1+\gamma)^2(1-\beta_1)^4r_{\max}^2}{\epsilon^3(1-\gamma\lambda)^4}(1+\upsilon)^{2T-4}+\frac{2dT\alpha\sqrt{N}G_\infty^4(1-\beta_1)^2}{\epsilon^2}\frac{\sigma}{1-\sigma}+\frac{2\alpha T^2\beta_1^2G_\infty^2}{\epsilon}$$

where

$$\upsilon\coloneqq\frac{\alpha(1+\gamma)(1-\beta_1)}{\epsilon(1-\gamma\lambda)}$$

$$\Gamma(T;t)\coloneqq\frac{2\varrho_0\varkappa^t(1+\gamma)}{T(1-\gamma\lambda)(1-\varkappa)}\cdot\max\{2\left\|\theta^*\right\|+r_{\max},1\}$$

and

$$\Gamma(T)\coloneqq\frac{2\varrho_0(1+\gamma)}{T(1-\gamma\lambda)(1-\varkappa)}\cdot\max\{2\left\|\theta^*\right\|+r_{\max},1\}.$$

Then, the formal statement of the convergence result about the average weight $\overline{\theta}_t$ is presented as follows.

Theorem 1: Under Assumptions 1–6, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t},\zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Let

$$\widehat{T}\coloneqq\min_T\left\{T\middle|\Gamma(T)<-\frac{(1-\beta_1)\kappa_{\max}^{\bar{Q}}+\frac{(\eta'+\mu')G_\infty}{T}}{4G_\infty(1-\beta_1)}\right\}$$

and

$$\widehat{\alpha}\coloneqq\min\left\{\alpha_0,\frac{1+\eta'+\mu'}{2\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}\right\}$$

where $\alpha_0>0$ is a constant. Moreover, define $t_\alpha\coloneqq\max\{t\geq1|\varkappa^t\geq\alpha\}$. Then, for $t\geq1$, we have

$$\mathbb{E}\left[\left\|\overline{\theta}_t-\theta^*\right\|^2\right]\leq\varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2-\frac{2\varrho_2\varrho_5'G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}+\min\{\varrho_4^{t-t_\alpha},1\}\cdot\left(\varrho_3\alpha^2-\frac{2\varrho_2\varrho_5'G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}\right)\tag{31}$$

where μ' and η' are some positive constants

$$\varrho_2\coloneqq\frac{\left(3+\frac{12\alpha^2(1-\beta_1)^2(1+\gamma)^2}{\epsilon(1-\gamma\lambda)^2}\right)^{\widehat{T}}-1}{2+\frac{12\alpha^2(1-\beta_1)^2(1+\gamma)^2}{\epsilon(1-\gamma\lambda)^2}}$$

$$\varrho_3\coloneqq\left[\frac{24(1-\beta_1)^2}{\epsilon(1-\gamma\lambda)^2}\left((1+\gamma)^2\left\|\theta^*\right\|^2+r_{\max}^2\right)+3\alpha^2\beta_1^2\frac{G_\infty^2}{\epsilon}+\frac{3d\sqrt{N}(1-\beta_1)^2G_\infty^4\sigma}{\epsilon^2(1-\sigma)}\right]\times\left[\frac{1-\widehat{T}}{2+\upsilon'}+\frac{(3+\upsilon')\left((3+\upsilon')^{\widehat{T}-1}-1\right)}{(2+\upsilon')^2}\right]$$

with

$$\upsilon'\coloneqq\frac{12\alpha^2(1-\beta_1)^2(1+\gamma)^2}{\epsilon(1-\gamma\lambda)^2}$$

$$\varrho_4\coloneqq1+\frac{\widehat{\alpha}\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{2\varrho_2G_\infty}\in(0,1)$$

$$\varrho_5\coloneqq\Psi_2(\widehat{\alpha},\widehat{T})-\frac{\varrho_3\alpha^2}{\varrho_2}\left[\eta'+\mu'+\frac{2\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{G_\infty}+\Psi_1(\widehat{\alpha},\widehat{T})\right]>0$$

and

$$\varrho_5'\coloneqq\Psi_2'(\widehat{\alpha},\widehat{T})-\frac{\varrho_3\alpha^2}{\varrho_2}\left[\eta'+\mu'+\frac{2\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{G_\infty}+\Psi_1(\widehat{\alpha},\widehat{T})\right].$$

The detailed proof of Theorem 1 can be found in Section VI. Under the Markovian model, the average weight $\overline{\theta}_t$ can linearly converge to the neighborhood of the optimal weight θ^*. Further, following from Lemma 2 and Theorem 1, we prove that $(1/N)\sum_{i=1}^N\mathbb{E}[\|\theta_{i,t}-\theta^*\|^2]$ is also linearly convergent under the Markovian model.

Proposition 1: Under Assumptions 1–6, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t},\zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Then, for $t\geq1$, we have

$$\frac{1}{N}\sum_{i=1}^N\mathbb{E}\left[\left\|\theta_{i,t}-\theta^*\right\|^2\right]\leq2\varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2+\frac{2\alpha^2}{1-\sigma^2}\frac{G_\infty^2}{\epsilon}+2\min\{\varrho_4^{t-t_\alpha},1\}\cdot\left(\varrho_3\alpha^2-\frac{2\varrho_2\varrho_5'G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}\right)-\frac{4\varrho_2\varrho_5'G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}.\tag{32}$$

The detailed proof of Proposition 1 can be found in Section VI. Furthermore, we also make the following remarks.

Remark 1: Proposition 1 implies that the proposed algorithm achieves the same convergence rate as the centralized TD learning [13], where a projection step is employed. However,

it is hard to implement the projection step in practice since it requires some prior knowledge with respect to the weight θ. Hence, the proposed algorithm has removed the projection step. In addition, the convergence rate of **D-AMSTD(λ)** also matches the centralized TD learning [15], where the projection step is removed. Nevertheless, only after a mixing-time, the bounds in [15] become available. However, the bounds obtained in this article are available for any $t \geq 1$.

Remark 2: **D-AMSTD(λ)** also achieves the same convergence rate as the distributed TD(λ) [24], where constant and diminishing learning rates are adopted. Similar to [15], these bounds in [24] can be applicable only after a mixing-time ($t \geq t_\alpha$), which denotes the second phase in our analysis. For any $t \geq 1$, however, our bounds in this article can be available. Further, the proposed algorithm combines the TD learning and the ADAM-type optimization algorithm, which has superior performance for training the models of deep learning. Furthermore, the proposed algorithm can mitigate the sensitive selection of learning rate because the adaptive learning rate is used in **D-AMSTD(λ)**.

VI. NONASYMPTOTIC ANALYSIS OF D-AMSTD(λ)

This section presents the proofs of the main results in detail. Toward this end, we present some lemmas, which are very important in our analysis. First, we bound uniformly the variables $g_i(\theta_{i,t}, \zeta_t)$, $m_{i,t}$, $v_{i,t}$, $\tilde{v}_{i,t}$, and $w_{i,t}$ below.

Lemma 1: Under Assumptions 1–3, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t}, \zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Then, for all $i \in \mathcal{V}$ and any $t \geq 0$, we have

$$\left\|g_i(\theta_{i,t}, \zeta_t)\right\|_\infty \leq \left\|g_i(\theta_{i,t}, \zeta_t)\right\| \leq \frac{(1+\gamma)D_\infty + r_{\max}}{1-\gamma\lambda}. \quad (33)$$

Moreover, we also have $\|m_{i,t}\|_\infty \leq \|m_{i,t}\| \leq G_\infty$, $\|v_{i,t}\|_\infty \leq G_\infty^2$, $\|\tilde{v}_{i,t}\|_\infty \leq G_\infty^2$, and $\|w_{i,t}\|_\infty \leq G_\infty^2$ for any $t \geq 0$ and all $i \in \mathcal{V}$, where $G_\infty := ([(1+\gamma)D_\infty + r_{\max}]/[1-\gamma\lambda])$.

Proof: See Appendix A in the supplementary material. ■

Next, we bound the consensus error $(1/N)\sum_{i=1}^N \|\theta_{i,t} - \overline{\theta}_t\|^2$, which plays a pivotal role in our proofs. The formal statement is described as follows.

Lemma 2: Under Assumptions 1–6, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t}, \zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Then, for any $t \geq 0$, we have

$$\frac{1}{N}\sum_{i=1}^N \left\|\theta_{i,t} - \overline{\theta}_t\right\|^2 \leq \frac{\alpha^2}{1-\sigma^2}\frac{dG_\infty^2}{\epsilon}. \quad (34)$$

Proof: See Appendix B in the supplementary material. ■

Next, we provide a nonasymptotic analysis of **D-AMSTD(λ)** under the Markovian model, where the observed data is collected from the trajectory of a single multiagent MDP. Compared to the i.i.d. model, the Markovian model will incur a biased estimate of $g_i(\theta_{i,t}, \zeta_t)$, which becomes a challenge in performance analysis. To address this issue, we first bound this bias by using Assumption 4. Moreover, this result is stated formally in the following lemma.

Lemma 3: Under Assumptions 1–6, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t}, \zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Then, we have

$$\left\|\frac{1}{TN}\sum_{\tau=t}^{t+T-1}\mathbb{E}\left[\Xi^\top(\Theta, \zeta_\tau)\mathbb{1}_N|\mathcal{H}_t\right] - \bar{g}(\overline{\theta})\right\| \leq \Gamma(T;t)\left(\left\|\overline{\theta} - \theta^*\right\| + 1\right) \quad (35)$$

for $\tau \geq 1$ and $\Theta \in \mathbb{R}^{N\times d}$, where

$$\Gamma(T;t) := \frac{2\varrho_0\varkappa^t(1+\gamma)}{T(1-\gamma\lambda)(1-\varkappa)} \cdot \max\{2\left\|\theta^*\right\| + r_{\max}, 1\}$$

and T is a positive integer.

Proof: See Appendix C in the supplementary material. ■

In the Markovian model, we cannot bound directly the mean-squared error since there exists a bias. To address this issue, we use Lemma 3 and a multistep Lyapunov function, which is also introduced in [18], [22], and [31]. Namely, define

$$\mathfrak{L}(t) := \sum_{\tau=t}^{t+T-1}\left\|\overline{\theta}_\tau - \theta^*\right\|^2. \quad (36)$$

To prove Theorem 1, we first bound the difference between $\mathfrak{L}(t+1)$ and $\mathfrak{L}(t)$. This result is stated formally as follows.

Lemma 4: Under Assumptions 1–6, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t}, \zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Moreover, there exist α and T such that

$$1 + \alpha\left(\eta' + \mu' + \frac{2T(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{G_\infty}\right) + \alpha\Psi_1(\alpha, T) \in (0,1).$$

Then, for $t \geq 0$, we have

$$\mathbb{E}[\mathfrak{L}(t+1) - \mathfrak{L}(t)|\mathcal{H}_t] \leq \alpha\Psi_2(\widehat{\alpha}, \widehat{T}) + \alpha\left[\eta' + \mu' + \frac{2\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{G_\infty} + \Psi_1(\widehat{\alpha}, \widehat{T})\right]\left\|\overline{\theta}_t - \theta^*\right\|^2 \quad (37)$$

where μ' and η' are positive constants

$$\widehat{T} = \min_T\left\{T\middle|\Gamma(T) < -\frac{(1-\beta_1)\kappa_{\max}^{\bar{Q}} + \frac{(\eta'+\mu')G_\infty}{T}}{4G_\infty(1-\beta_1)}\right\}$$

and

$$\widehat{\alpha} := \min\left\{\alpha_0, \frac{1+\eta'+\mu'}{2\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}\right\}.$$

Proof: See Appendix D in the supplementary material. ■

Further, the multistep Lyapunov function $\mathfrak{L}(t)$ is also upper bounded by the following result.

Lemma 5: Under Assumptions 1–6, the sequences $\{\theta_{i,t}\}$, $\{g_i(\theta_{i,t}, \zeta_t)\}$, $\{m_{i,t}\}$, $\{v_{i,t}\}$, $\{\tilde{v}_{i,t}\}$, and $\{w_{i,t}\}$ are generated by Algorithm 1. Then, for $t \geq 0$, we have

$$\mathfrak{L}(t) \leq \varrho_2\left\|\overline{\theta}_t - \theta^*\right\|^2 + \varrho_3\alpha^2 \quad (38)$$

where

$$\varrho_2 := \frac{\left(3 + \frac{12\alpha^2(1-\beta_1)^2(1+\gamma)^2}{\epsilon(1-\gamma\lambda)^2}\right)^{\widehat{T}} - 1}{2 + \frac{12\alpha^2(1-\beta_1)^2(1+\gamma)^2}{\epsilon(1-\gamma\lambda)^2}}$$

and

$$\varrho_3 \coloneqq \left[\frac{24(1-\beta_1)^2}{\epsilon(1-\gamma\lambda)^2}\left((1+\gamma)^2\left\|\theta^*\right\|^2 + r_{\max}^2\right) + 3\alpha^2\beta_1^2\frac{G_\infty^2}{\epsilon} + \frac{3d\sqrt{N}(1-\beta_1)^2G_\infty^4\sigma}{\epsilon^2(1-\sigma)}\right] \times \left[\frac{1-\widehat{T}}{2+\upsilon'} + \frac{(3+\upsilon')\left((3+\upsilon')^{\widehat{T}-1}-1\right)}{(2+\upsilon')^2}\right]$$

with

$$\upsilon' \coloneqq \frac{12\alpha^2(1-\beta_1)^2(1+\gamma)^2}{\epsilon(1-\gamma\lambda)^2}.$$

Proof: See Appendix E in the supplementary material. ∎

Due to space limitation, we only provide the proof roadmap of Theorem 1 as follows. The detailed proof can be found in Appendix F in the supplemental material.

Proof Roadmap of Theorem 1: The time t is divided into two phases: 1) $t \le t_\alpha$ and 2) $t > t_\alpha$, where $t_\alpha \coloneqq \max\{t \ge 1 | \varkappa^t \ge \alpha\}$.

1) The First Phase. By using Lemmas 4 and 5, we have

$$\begin{aligned}\mathbb{E}[\mathfrak{L}(t+1)|\mathcal{H}_t] &\le \left[1 + \frac{\widehat{\alpha}\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{2\varrho_2 G_\infty}\right]\mathbb{E}[\mathfrak{L}(t)|\mathcal{H}_t] \\ &\quad - \frac{\varrho_3\alpha^3}{\varrho_2}\left[\eta' + \mu' + \frac{2\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}{G_\infty} + \Psi_1(\widehat{\alpha},\widehat{T})\right] \\ &\quad + \alpha\Psi_2(\widehat{\alpha},\widehat{T}) \\ &= \varrho_4\mathbb{E}[\mathfrak{L}(t)|\mathcal{H}_t] + \varrho_5\alpha.\end{aligned}$$

Using some algebraic manipulations yields

$$\begin{aligned}\mathbb{E}[\mathfrak{L}(t)] &\le \varrho_4^t\mathfrak{L}(0) + \varrho_5\alpha\frac{\varrho_4^t-1}{\varrho_4-1} \\ &\le \varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 + \varrho_3\alpha^2\varrho_4^t + \varrho_5\alpha\frac{\varrho_4^t-1}{\varrho_4-1} \\ &\le \varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 + \varrho_3\alpha^2 + \frac{\varrho_5\alpha}{1-\varrho_4} \\ &\le \varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 + \varrho_3\alpha^2 - \frac{2\varrho_2\varrho_5 G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}.\end{aligned}$$

Following the definition of $\mathfrak{L}(t)$, we obtain:

$$\begin{aligned}\mathbb{E}\left[\left\|\overline{\theta}_t-\theta^*\right\|^2\right] &\le \mathbb{E}[\mathfrak{L}(t)] \\ &\le \varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 + \varrho_3\alpha^2 \\ &\quad - \frac{2\varrho_2\varrho_5 G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}.\end{aligned}$$

2) The Second Phase: For $t > t_\alpha$, we first have

$$\mathbb{E}[\mathfrak{L}(t+1)|\mathcal{H}_t] \le \varrho_4\mathbb{E}[\mathfrak{L}(t)|\mathcal{H}_t] + \varrho_5'\alpha.$$

Then, we have

$$\begin{aligned}\mathbb{E}[\mathfrak{L}(t)] &\le \varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 + \varrho_3\alpha^2\varrho_4^{t-t_\alpha} \\ &\quad - \left(\varrho_4^{t-t_\alpha}+1\right)\frac{2\varrho_2\varrho_5' G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}.\end{aligned}$$

Using some algebraic manipulations, we can obtain

$$\begin{aligned}\mathbb{E}\left[\left\|\overline{\theta}_t-\theta^*\right\|^2\right] &\le \varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 - \frac{2\varrho_2\varrho_5' G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}} \\ &\quad + \min\left\{\varrho_4^{t-t_\alpha}, 1\right\}\cdot\left(\varrho_3\alpha^2 - \frac{2\varrho_2\varrho_5' G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}\right).\end{aligned}$$

Thus, we obtain the result of Theorem 1.

Finally, we prove Proposition 1 by using Lemma 2 and Theorem 1. The more detailed proof is presented as follows.

Proof of Proposition 1: Following from the inequality $\|x+y\|^2 \le 2\|x\|^2 + 2\|y\|^2$, we obtain that:

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^N\mathbb{E}\left[\left\|\theta_{i,t}-\theta^*\right\|^2\right] &= \frac{1}{N}\sum_{i=1}^N\mathbb{E}\left[\left\|\theta_{i,t}-\overline{\theta}_t+\overline{\theta}_t-\theta^*\right\|^2\right] \\ &\le \frac{2}{N}\sum_{i=1}^N\mathbb{E}\left[\left\|\theta_{i,t}-\overline{\theta}_t\right\|^2\right] \\ &\quad + 2\mathbb{E}\left[\left\|\overline{\theta}_t-\theta^*\right\|^2\right] \\ &\le \frac{2\alpha^2}{1-\sigma^2}\frac{G_\infty^2}{\epsilon} + 2\varrho_2\varrho_4^t\left\|\overline{\theta}_0-\theta^*\right\|^2 \\ &\quad - \frac{4\varrho_2\varrho_5' G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}} \\ &\quad + 2\min\left\{\varrho_4^{t-t_\alpha}, 1\right\} \\ &\quad \cdot\left(\varrho_3\alpha^2 - \frac{2\varrho_2\varrho_5' G_\infty}{\widehat{T}(1-\beta_1)\kappa_{\max}^{\bar{Q}}}\right) \quad (39)\end{aligned}$$

where (39) is derived from Lemma 2 and Theorem 1. Then, we prove completely the result in Proposition 1. ∎

VII. Experiments

In this section, we conduct a variety of experiments to validate the theoretical analysis and evaluate the performance of **D-AMSTD(λ)**. To this end, we implement **D-AMSTD(λ)** to solve the *cooperative navigation* task [46], where all landmarks are occupied by agents in a cooperative manner.

To evaluate the performance of the proposed algorithm, the mean consensus error is used to measure the performance of **D-AMSTD(λ)**. In addition, we compare **D-AMSTD(λ)** with popular decentralized TD learning algorithms, including distributed TD(0) [22], distributed TD(0)+GT [23], distributed TD(0)+GC [23], MS-ADTD [30], and distributed TD(λ) [24]. For a fair comparison, we also implement the above-mentioned methods to solve the cooperative navigation task.

A. Experimental Settings

In our experiments, we set hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\gamma = 0.9$. Moreover, the learning rate α_t is selected as $\alpha_t = \alpha_0/\sqrt{t+1}$, where α_0 is a positive constant.

B. Experimental Results

First, we compare **D-AMSTD(λ)** with other decentralized TD learning algorithms with different numbers of agents under the cycle graph and star graph, respectively. In this experiment,

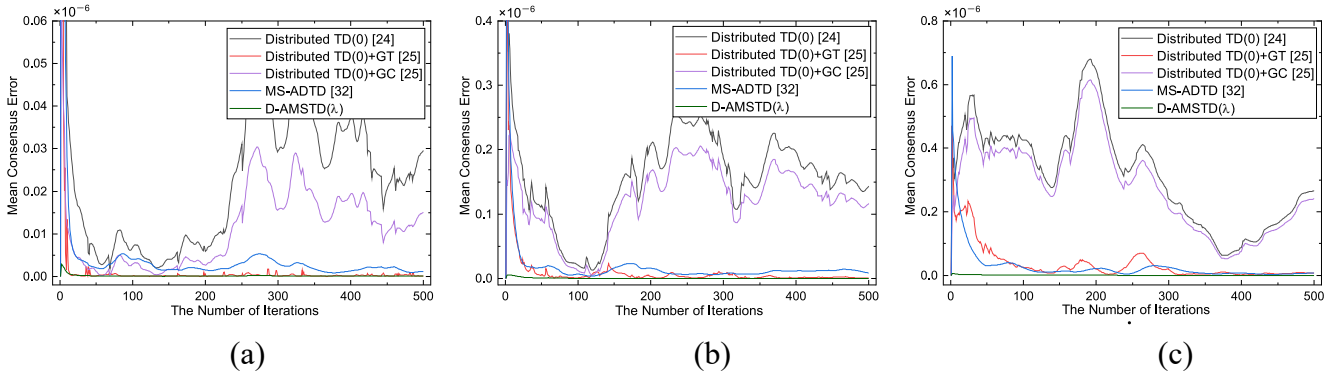

Fig. 1. Comparison of mean consensus error among distributed TD(0), distributed TD(0)+GT, distributed TD(0)+GC, MS-ADTD, and **D-AMSTD**(λ) on a cycle graph with different sizes. (a) 7 Agents. (b) 12 Agents. (c) 18 Agents.

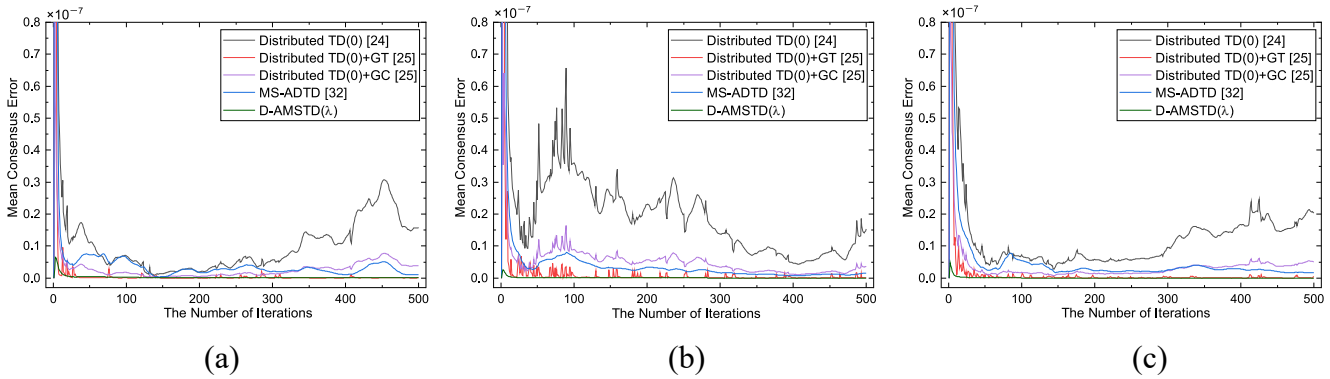

Fig. 2. Comparison of mean consensus error among distributed TD(0), distributed TD(0)+GT, distributed TD(0)+GC, MS-ADTD, and **D-AMSTD**(λ) on a star graph with different sizes. (a) 7 Agents. (b) 12 Agents. (c) 18 Agents.

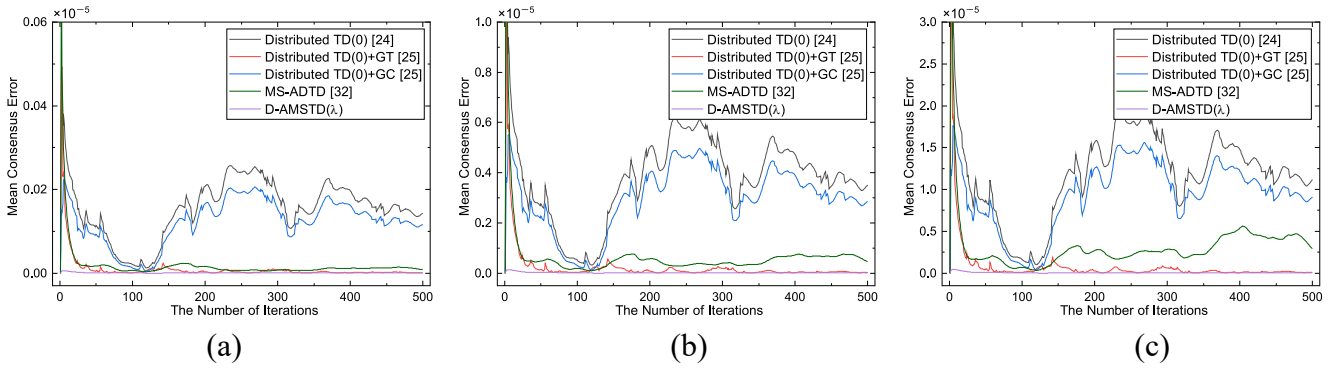

Fig. 3. Comparison of mean consensus error among distributed TD(0), distributed TD(0)+GT, distributed TD(0)+GC, MS-ADTD, and **D-AMSTD**(λ) on a cycle graph with fixed 12 agents and different α_0. (a) $\alpha_0 = 0.001$. (b) $\alpha_0 = 0.005$. (c) $\alpha_0 = 0.009$.

we set $\lambda = 0.9$. The experimental results are shown in Figs. 1 and 2, respectively. We can see that our algorithm **D-AMSTD**(λ) is overall faster than other decentralized TD learning methods for different numbers of agents under the cycle graph and star graph, respectively. Furthermore, the fluctuation of our algorithm is also smaller than other methods under different graphs with varying numbers of agents. In other words, **D-AMSTD**(λ) can more stably converge to the optimal point than other decentralized TD learning methods.

In the second experiment, we study how the learning rate affects the mean consensus error under the cycle graph with 12 agents. The results are shown in Fig. 3. More specifically, Fig. 3 shows that the proposed algorithm **D-AMSTD**(λ) is more robust to the selection of the hyperparameter α_0, which affects the learning rate α_t. In other words, **D-AMSTD**(λ) is less sensitive to the selection of learning rates compared with other distributed TD(0) learning methods.

In the third experiment, we test the mean consensus errors of **D-AMSTD**(λ) under the cycle graph with 12 agents, where $\alpha_0 = 0.5$. As shown in Fig. 4, we can observe that **D-AMSTD**(λ) outperforms distributed TD(λ) [24] when α_0 is larger. Furthermore, distributed TD(λ) is unstable with larger α_0 for different λ. In comparison, our algorithm **D-AMSTD**(λ) is still stable with larger α_0. According to Fig. 5, we also see

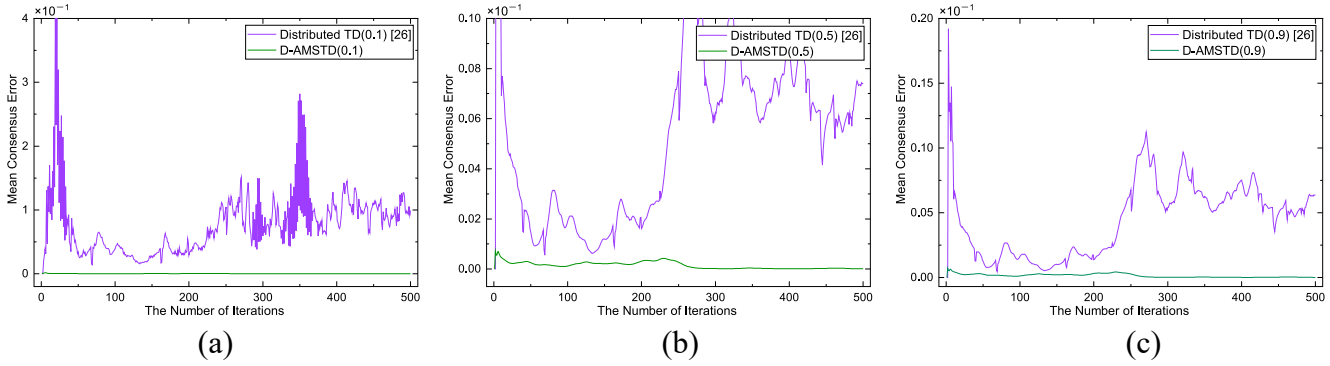

Fig. 4. Comparison of mean consensus error between distributed TD(λ) and **D-AMSTD**($\boldsymbol{\lambda}$) on a cycle graph with larger $\alpha_0 = 0.5$ and different λ. (a) $\lambda = 0.1$. (b) $\lambda = 0.5$. (c) $\lambda = 0.9$.

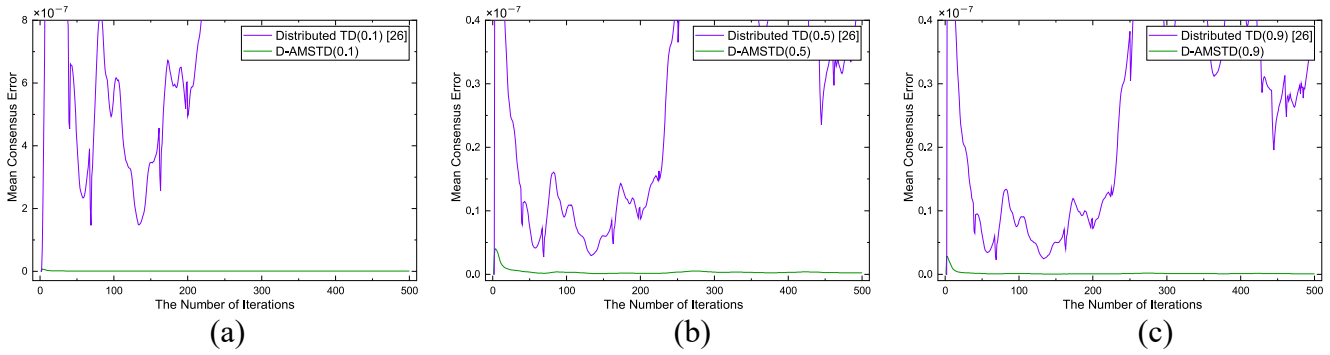

Fig. 5. Comparison of mean consensus error between distributed TD(λ) and **D-AMSTD**($\boldsymbol{\lambda}$) on a cycle graph with smaller $\alpha_0 = 0.001$ and different λ. (a) $\lambda = 0.1$. (b) $\lambda = 0.5$. (c) $\lambda = 0.9$.

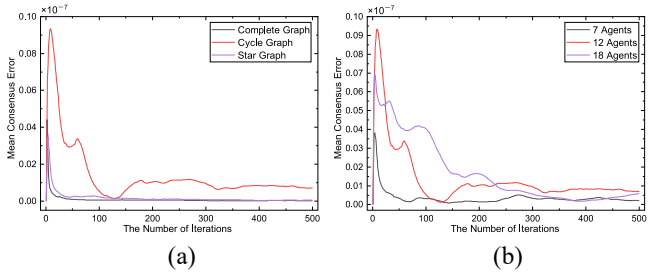

Fig. 6. (a) Comparison of **D-AMSTD**($\boldsymbol{\lambda}$) on graphs with different topology and (b) comparison of **D-AMSTD**($\boldsymbol{\lambda}$) on a cycle graph with different sizes.

that our algorithm **D-AMSTD**($\boldsymbol{\lambda}$) is more stable than distributed TD(λ) with different λ when $\alpha_0 = 0.001$. Moreover, when α_0 is smaller, distributed TD(λ) is still less stable under different λ. Thus, **D-AMSTD**($\boldsymbol{\lambda}$) is more robust to the selection of α_0 and λ. Moreover, we are easier to tune **D-AMSTD**($\boldsymbol{\lambda}$) compared to distributed TD(λ) in practice.

Finally, we investigate how the mean consensus error of **D-AMSTD**($\boldsymbol{\lambda}$) is affected by different graphs and the number of agents, where $\lambda = 0.9$. As shown in Fig. 6(a), the mean consensus error decreases more slowly on graphs with worse connectivity than on graphs with better connectivity. Meanwhile, Fig. 6(b) shows that **D-AMSTD**($\boldsymbol{\lambda}$) on smaller graphs is slightly faster than on larger graphs. Thereby, theoretical predictions are agreed with empirical results.

VIII. Conclusion

This article has proposed a decentralized adaptive TD(λ) learning algorithm termed as **D-AMSTD**($\boldsymbol{\lambda}$) for the decentralized policy evaluation problem. Moreover, we have also analyzed rigorously the finite-time performance of **D-AMSTD**($\boldsymbol{\lambda}$) under the Markovian model, i.e., **D-AMSTD**($\boldsymbol{\lambda}$) linearly converges to an arbitrarily small size of neighborhood of the optimal weight by appropriately choosing learning rates. Furthermore, the convergence rate can match the centralized variants of TD learning. Finally, we have conducted various experiments to verify the efficacy of **D-AMSTD**($\boldsymbol{\lambda}$).

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[2] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, 1988.
[3] T. Song, D. Li, Q. Jin, and K. Hirasawa, "Sparse proximal reinforcement learning via nested optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 11, pp. 4020–4032, Nov. 2020.
[4] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[5] D. Silver et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[6] D. Silver et al., "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
[7] O. Vinyals et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[8] J. N. Tsitsiklis and B. V. Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Automat. Control*, vol. 42, no. 5, pp. 674–690, May 1997.
[9] P. Dayan, "The convergence of TD(λ) for general λ," *Mach. Learn.*, vol. 8, pp. 341–362, May 1992.
[10] V. Tadić, "On the convergence of temporal-difference learning with linear function approximation," *Mach. Learn.*, vol. 42, no. 3, pp. 241–267, 2001.
[11] B. Hu and U. A. Syed, "Characterizing the exact behaviors of temporal difference learning algorithms using Markov jump linear system theory," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 8477–8488.
[12] G. Wang, S. Lu, G. B. Giannakis, G. Tesauro, and J. Sun, "Decentralized TD tracking with linear function approximation and its finite-time analysis," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 1–11.
[13] J. Bhandari, D. Russo, and R. Singal, "A finite time analysis of temporal difference learning with linear function approximation," in *Proc. 31st Annu. Conf. Learn. Theory, (COLT)*, 2018, pp. 1691–1692.
[14] G. Dalal, B. Szörényi, G. Thoppe, and S. Mannor, "Finite sample Analyses for TD(0) with function approximation," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 6144–6160.
[15] R. Srikant and L. Ying, "Finite-time error bounds for linear stochastic approximation and TD learning," in *Proc. 32nd Annu. Conf. Learn. Theory*, 2019, pp. 2803–2830.
[16] T. Sun, H. Shen, T. Chen, and D. Li, "Adaptive temporal difference learning with linear function approximation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 8812–8824, Dec. 2022.
[17] H. Xiong, T. Xu, Y. Liang, and W. Zhang, "Non-asymptotic convergence of Adam-type reinforcement learning algorithms under Markovian sampling," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 10460–10468.
[18] G. Wang, "Finite-time error bounds of biased stochastic approximation with application to TD-learning," *IEEE Trans. Signal Process.*, vol. 70, pp. 950–962, 2022.
[19] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.
[20] A. Mathkar and V. S. Borkar, "Distributed reinforcement learning via gossip," *IEEE Trans. Automat. Control*, vol. 62, no. 3, pp. 1465–1470, Mar. 2017.
[21] T. T. Doan, S. T. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD(0) with linear function approximation for multi-agent reinforcement learning," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 1626–1635.
[22] J. Sun, G. Wang, G. B. Giannakis, Q. Yang, and Z. Yang, "Finite-time analysis of decentralized temporal-difference learning with linear function approximation," in *Proc. 23rd Int. Conf. Artif. Intell. Statist., (AISTATS)*, 2020, pp. 4485–4495.
[23] Q. Lin and Q. Ling, "Decentralized TD(0) with gradient tracking," *IEEE Signal Process. Lett.*, vol. 28, pp. 723–727, Apr. 2021.
[24] T. T. Doan, S. T. Maguluri, and J. Romberg, "Finite-time performance of distributed temporal-difference learning with linear function approximation," *SIAM J. Math. Data Sci.*, vol. 3, no. 1, pp. 298–320, 2021.
[25] Z. Wu, H. Shen, T. Chen, and Q. Ling, "Byzantine-resilient decentralized policy evaluation with linear function approximation," *IEEE Trans. Signal Process.*, vol. 69, pp. 3839–3853, Jun. 2021.
[26] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *J. Mach. Learn. Res.*, vol. 5, pp. 1–25, Dec. 2003.
[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–15.
[28] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–23.
[29] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural Optimizer search with reinforcement learning," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 459–468.
[30] J. Zhu et al., "Provable distributed adaptive temporal-difference learning over time-varying networks," *Expert Syst. Appl.*, vol. 228, Oct. 2023, Art. no. 120406.
[31] G. Wang, B. Li, and G. B. Giannakis, "A multistep Lyapunov approach for finite-time analysis of biased stochastic approximation," 2020, *arXiv:1909.04299*.
[32] T. S. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, no. 6, pp. 1185–1201, 1994.
[33] N. Korda and P. La, "On TD(0) with function approximation: Concentration bounds and a centered variant with exponential convergence," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 626–634.
[34] C. Lakshminarayanan and C. Szepesvári, *Finite Time Bounds for Temporal Difference Learning with Function Approximation: Problems with Some 'State-of-the-Art' Results*, Univ. Alberta, Edmonton, AB, Canada, 2017.
[35] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Jul. 2011.
[36] G. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.
[37] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.
[38] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Math. Program.*, vol. 187, no. 1, pp. 409–457, 2021.
[39] M. Zhang, B. Hao, Q. Ge, J. Zhu, R. Zheng, and Q. Wu, "Distributed adaptive subgradient algorithms for online learning over time-varying networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 7, pp. 4518–4529, Jul. 2022.
[40] M. Zhang, Y. Zhou, Q. Ge, R. Zheng, and Q. Wu, "Decentralized randomized block-coordinate frank-Wolfe algorithms for submodular Maximization over networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 8, pp. 5081–5091, Aug. 2022.
[41] X. Chen, B. Karimi, W. Zhao, and P. Li, "On the convergence of decentralized adaptive gradient methods," 2021, *arXiv:2109.03194*.
[42] P. Nazari, D. A. Tarzanagh, and G. Michailidis, "DADAM: A consensus-based distributed adaptive gradient method for online optimization," 2019, *arXiv:1901.09109*.
[43] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times*, vol. 107, Providence, RI, USA: Am. Math. Soc., 2017.
[44] A. Nedić, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proc. IEEE*, vol. 106, no. 5, pp. 953–976, May 2018.
[45] J. Zhu, Q. Wu, M. Zhang, R. Zheng, and K. Li, "Projection-free decentralized online learning for submodular maximization over time-varying networks," *J. Mach. Learn. Res.*, vol. 22, no. 51, pp. 1–42, 2021.
[46] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6379–6390.