A Primal-dual Perspective for Distributed TD-learning

Han-Dong Lim , Donghwan Lee
Department of Electrical Engineering, KAIST
{limaries30, donghwan}@kaist.ac.kr

Abstract

The goal of this paper is to investigate distributed temporal difference (TD) learning for a networked multi-agent Markov decision process. The proposed approach is based on distributed optimization algorithms, which can be interpreted as primal-dual ordinary differential equation (ODE) dynamics subject to null-space constraints. Based on the exponential convergence behavior of the primal-dual ODE dynamics subject to null-space constraints, we examine the behavior of the final iterate in various distributed TD-learning scenarios, considering both constant and diminishing step-sizes and incorporating both i.i.d. and Markovian observation models. Unlike existing methods, the proposed algorithm does not require the assumption that the underlying communication network structure is characterized by a doubly stochastic matrix.

1 Introduction

Temporal-difference (TD) learning [Sutton, 1988] aims to solve the policy evaluation problem in Markov decision processes (MDPs), serving as the foundational pillar for many reinforcement learning (RL) algorithms [Mnih *et al.*, 2015]. Following the empirical success of RL in various fields [Wang *et al.*, 2022], theoretical exploration of TD-learning has become an active area of research. For instance, [Tsitsiklis and Van Roy, 1996] studied the asymptotic convergence of TD-learning, while non-asymptotic analysis has been examined in [Bhandari *et al.*, 2018; Srikant and Ying, 2019; Lee and Kim, 2022].

In contrast to the single-agent case, the theoretical understanding for TD-learning for networked multi-agent Markov decision processes (MAMDPs) has not been fully explored so far. In the networked MAMDPs, each agent follows its own policy and receives different local rewards while sharing their local learning parameters through communication networks. Under this scenario, several distributed TD-learning algorithms [Wang *et al.*, 2020; Doan *et al.*, 2019; Doan *et al.*, 2021; Sun *et al.*, 2020; Zeng *et al.*, 2022] have been developed based on distributed optimization frameworks [Nedic and Ozdaglar, 2009; Pu and Nedić, 2021].

The main goal of this paper is to provide finite-time analysis of a distributed TD-learning algorithm for networked MAMDPs from the perspectives of the primal-dual algorithms [Wang and Elia, 2011]. The proposed algorithms are inspired by the control system model for distributed optimization problems [Wang and Elia, 2011; Lee, 2023], and it can also be interpreted as the primal-dual gradient dynamics in [Qu and Li, 2018]. In this respect, we first study finite-time analysis of continuous-time primal-dual gradient dynamics in [Qu and Li, 2018] with special nullity structures on the system matrix. Based on the analysis of primal-dual gradient dynamics, we further provide a finite-time analysis of the proposed distributed TD-learning under both i.i.d. observation and Markov observation models. The main contributions are summarized as follows:

1. An improved or comparable to the state of art convergence rate for continuous-time primal-dual gradient dynamics [Qu and Li, 2018] with null-space constraints under specific conditions: the results can be applied to general classes of distributed optimization problems that can be reformulated as saddle-point problems [Wang and Elia, 2011];
2. Development of new distributed TD-learning algorithm inspired by [Wang and Elia, 2011; Lee, 2023], which does not require a double stochastic matrix. This offers a significant advantage in specific scenarios, such as wireless ad hoc networks or broadcast-based communication, where node degrees (number of neighbours) are often unknown due to factors like message loss during transmission [Hendrickx and Tsitsiklis, 2015]. This uncertainty makes it challenging to construct a doubly stochastic matrix, as most existing methods rely on precise knowledge of node degrees. In contrast, our algorithm does not require such additional information and thus remains effective in these environments;
3. New mean-squared error bounds of the distributed TD-learning under our consideration for both i.i.d. and Markovian observation models and under various conditions of the step-sizes: the distributed TD-learning is based on the control system model in [Wang and Elia, 2011; Lee, 2023] which does not require doubly stochastic matrix corresponding to its associated network graph. Note that the doubly stochastic assumption is required

in other distributed TD-learning algorithms based on the classical distributed optimization algorithms [Nedic and Ozdaglar, 2009; Pu and Nedić, 2021];

4. Empirical demonstrations of both the convergence and the rate of convergence of the algorithm are provided.

Related Works. Distributed optimization has been an active research field. In this context, [Nedic and Ozdaglar, 2009] investigated a distributed optimization algorithm over a communication network whose structure graph is represented by a doubly stochastic matrix. In this approach, each agent exchanges information with its neighbors, with the exchange being weighted by the corresponding element in the doubly stochastic matrix. Meanwhile, [Wang and Elia, 2011; Notarnicola *et al.*, 2023] provided control system approach to study distributed optimization problem.

The asymptotic convergence of distributed TD-learning has been studied in [Mathkar and Borkar, 2016; Stanković *et al.*, 2023]. [Doan *et al.*, 2019] provided finite-time analysis of distributed TD-learning based on the distributed optimization algorithm [Nedic and Ozdaglar, 2009] with i.i.d. observation model. Their analysis was extended to the Markovian observation model [Doan *et al.*, 2021]. [Sun *et al.*, 2020] studied distributed TD-learning based on [Nedic and Ozdaglar, 2009] with the Markovian observation model using multistep Lyapunov function [Wang *et al.*, 2019]. [Wang *et al.*, 2020] studied distributed TD-learning motivated by the gradient tracking method [Pu and Nedić, 2021]. [Zeng *et al.*, 2022] studied finite-time behavior of distributed stochastic approximation algorithms [Robbins and Monro, 1951] with general mapping including TD-learning and Q-learning, using Lyapunov-Razumikhin function [Zhou and Luo, 2018].

In the context of policy evaluation, [Macua *et al.*, 2014; Lee *et al.*, 2018; Wai *et al.*, 2018; Cassano *et al.*, 2020] studied distributed versions of gradient-TD [Sutton *et al.*, 2009]. The Gradient-TD method is reformulated as saddle-point problem [Macua *et al.*, 2014; Lee *et al.*, 2022], and the aforementioned works can be understood as distributed optimization over a saddle-point problem [Boyd and Vandenberghe, 2004].

2 Preliminaries

2.1 Markov Decision Process

Markov decision process (MDP) consists of five tuples $(\mathcal{S},\mathcal{A},\gamma,\mathcal{P},r)$, where $\mathcal{S} := \{1,2,\dots,|\mathcal{S}|\}$ is the collection of states, $\mathcal{A}$ is the collection of actions, $\gamma \in (0,1)$ is the discount factor, $\mathcal{P} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$ is the transition kernel, and $r : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ is the reward function. If action $a \in \mathcal{A}$ is chosen at state $s \in \mathcal{S}$, the transition to state $s' \in \mathcal{S}$ occurs with probability $\mathcal{P}(s,a,s')$, and incurs reward $r(s,a,s')$. Given a stochastic policy $\pi : \mathcal{S}\times\mathcal{A} \to [0,1]$, the quantity $\pi(a \mid s)$ denotes the probability of taking action $a \in \mathcal{A}$ at state $s \in \mathcal{S}$. We will denote $\mathcal{P}^\pi(s,s') := \sum_{a\in\mathcal{A}} \mathcal{P}(s,a,s')\pi(a \mid s)$, and $\mathcal{R}^\pi(s) := \sum_{a\in\mathcal{A}}\sum_{s'\in\mathcal{S}} \mathcal{P}(s,a,s')\pi(a \mid s)r(s,a,s')$, which is the transition probability from state $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ under policy π, and expected reward at state $s \in \mathcal{S}$, respectively. $d : \mathcal{S} \to [0,1]$ denotes the stationary distribution of the state $s \in \mathcal{S}$ under policy π. The policy evaluation problem aims to estimate the expected sum of discounted rewards following policy π, the so-called the value function, $v^\pi(s) = \mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k r(s_k,a_k,s_{k+1}) \middle| s_0 = s,\pi\right]$ for $s \in \mathcal{S}$.

Given a feature function $\boldsymbol{\phi} : \mathcal{S} \to \mathbb{R}^q$, our aim is to estimate the value function through learnable parameter $\boldsymbol{\theta}$, i.e., $v^\pi(s) \approx \boldsymbol{\phi}(s)^\top\boldsymbol{\theta}$, for $s \in \mathcal{S}$, which can be achieved through solving the optimization problem, $\min_{\boldsymbol{\theta}\in\mathbb{R}^q} \frac{1}{2}\left\|\boldsymbol{R}^\pi + \gamma\boldsymbol{P}^\pi\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{\Phi}\boldsymbol{\theta}\right\|^2_{\boldsymbol{D}^\pi}$, where $\boldsymbol{D}^\pi$ is a diagonal matrix whose elements are $d(1), d(2), \dots, d(|\mathcal{S}|)$, $\boldsymbol{P}^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ whose elements are $[\boldsymbol{P}^\pi]_{ij} := \mathcal{P}^\pi(i,j)$ for $i,j \in \mathcal{S}$, $\boldsymbol{R}^\pi \in \mathbb{R}^{|\mathcal{S}|}$, $[\boldsymbol{R}^\pi]_i := \mathbb{E}\left[r(s,a,s')|s = i\right]$ for $i \in \mathcal{S}$, and $\boldsymbol{\Phi} := \begin{bmatrix}\boldsymbol{\phi}(1) & \boldsymbol{\phi}(2) & \cdots & \boldsymbol{\phi}(|\mathcal{S}|)\end{bmatrix}^\top \in \mathbb{R}^{|\mathcal{S}|\times q}$. The solution of the optimization problem satisfies the so-called projected Bellman equation [Sutton *et al.*, 2009]:

$$\boldsymbol{\Phi}^\top\boldsymbol{D}^\pi\boldsymbol{\Phi}\boldsymbol{\theta} = \boldsymbol{\Phi}^\top\boldsymbol{D}^\pi\boldsymbol{R}^\pi + \gamma\boldsymbol{\Phi}^\top\boldsymbol{D}^\pi\boldsymbol{P}^\pi\boldsymbol{\Phi}\boldsymbol{\theta}.$$

Throughout the paper, we adopt the common assumption on the feature matrix, which is widely used in the literature [Bhandari *et al.*, 2018; Wang *et al.*, 2020].

Assumption 1. *$\left\|\boldsymbol{\phi}(s)\right\|_2 \le 1$ for all $s \in \mathcal{S}$ and $\boldsymbol{\Phi}$ is full-column rank matrix.*

2.2 Multi-Agent MDP

Multi-agent Markov decision process (MAMDP) considers a set of agents cooperatively computing the value function for a shared environment. Considering N agents, each agent can be denoted by $i \in \mathcal{V} := \{1,2,\dots,N\}$, and the agents communicate over networks that can be described by a connected and undirected simple graph $\mathcal{G} := (\mathcal{V},\mathcal{E})$, where $\mathcal{E} \subset \mathcal{V}\times\mathcal{V}$ is the set of edges. $\mathcal{N}_i \subset \mathcal{V}$ denotes the neighbour of agent $i \in \mathcal{V}$, i.e., $j \in \mathcal{N}_i$ if and only if $(i,j) \in \mathcal{E}$ for $i,j \in \mathcal{V}$. Each agent $i \in \mathcal{V}$ has its local policy $\pi^i : \mathcal{S}\times\mathcal{A}_i \to [0,1]$, where $\mathcal{A}_i$ is the action space of agent i, and receives reward following its local reward function $r^i : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ where $\mathcal{A} := \Pi_{i=1}^N\mathcal{A}_i$. MAMDP consists of five tuples $(\mathcal{S},\mathcal{A},\gamma,\mathcal{P},\{r^i\}_{i=1}^N)$, where $\mathcal{P} : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to [0,1]$ is the Markov transition kernel. The agents share the same state $s \in \mathcal{S}$, and when action $\boldsymbol{a} := (a_1,a_2,\dots,a_N) \in \mathcal{A}$ is taken, the state transits to $s' \in \mathcal{S}$ with probability $\mathcal{P}(s,\boldsymbol{a},s')$, and for $i \in \mathcal{V}$, agent i receives $r^i(s,\boldsymbol{a},s')$. The aim of the policy evaluation under MAMDP is to estimate the expected sum of discounted rewards averaged over N agents, i.e., $v^\pi(s) = \mathbb{E}\left[\sum_{k=0}^\infty\gamma^k\frac{1}{N}\sum_{i=1}^N r^i(s_k,\boldsymbol{a},s_{k+1})\right]$, for $s \in \mathcal{S}$. While learning, each agent $i \in \mathcal{V}$ can share its learning parameter over the communication network with its neighboring agents $j \in \mathcal{N}_i$. Following the spirit of single-agent MDP, the aim of each agent is now to compute the solution of the following equation:

$$\boldsymbol{\Phi}^\top\boldsymbol{D}^\pi\boldsymbol{\Phi}\boldsymbol{\theta} = \boldsymbol{\Phi}^\top\boldsymbol{D}^\pi\left(\frac{1}{N}\sum_{i=1}^N\boldsymbol{R}_i^\pi + \gamma\boldsymbol{P}^\pi\boldsymbol{\Phi}\boldsymbol{\theta}\right), \quad (1)$$

where $\boldsymbol{R}_i^\pi \in \mathbb{R}^{|\mathcal{S}|}$ for $i \in \mathcal{V}$, whose elements are $[\boldsymbol{R}_i^\pi]_j = \mathbb{E}\left[r^i(s,\boldsymbol{a},s') \mid s = j\right]$ for $j \in \mathcal{S}$. The equation (1) admits a unique solution $\boldsymbol{\theta}_c \in \mathbb{R}^q$, given by

$$\boldsymbol{\theta}_c = (\boldsymbol{\Phi}^\top \boldsymbol{D}^\pi(\boldsymbol{\Phi} - \gamma \boldsymbol{P}^\pi \boldsymbol{\Phi}))^{-1} \boldsymbol{\Phi}^\top \boldsymbol{D}^\pi \left(\frac{1}{N} \sum_{i=1}^{N} \boldsymbol{R}_i^\pi \right). \tag{2}$$

Note that the solution corresponds to the value function associated with the global reward $\sum_{k=0}^{\infty} \gamma^k \frac{1}{N} \sum_{i=1}^{N} r^i(s_k, \boldsymbol{a}_k, s_{k+1})$. Moreover, we will denote, for $1 \leq i \leq N$,

$$\boldsymbol{A} := \gamma \boldsymbol{\Phi}^\top \boldsymbol{D}^\pi \boldsymbol{\Phi} - \boldsymbol{\Phi}^\top \boldsymbol{D}^\pi \boldsymbol{P}^\pi \boldsymbol{\Phi}, \quad \boldsymbol{b}_i := \boldsymbol{\Phi}^\top \boldsymbol{D}^\pi \boldsymbol{R}_i^\pi, \tag{3}$$

and $w := \lambda_{\min}(\boldsymbol{\Phi}^\top \boldsymbol{D}^\pi \boldsymbol{\Phi})$. The bound on the reward will be denoted by a positive constant $R_{\max} \in \mathbb{R}$, i.e., $|r^i(s, \boldsymbol{a}, s')| \leq R_{\max}, \ 1 \leq i \leq N, \forall s, \boldsymbol{a}, s' \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$.

3 Analysis of Primal-Dual Gradient Dynamics

The so-called primal-dual gradient dynamics [Arrow *et al.*, 1958] will be the key tool for the analysis of the proposed distributed TD-learning. The analysis provided in this section will serve as the foundation for the subsequent analysis in Section 4. This section establishes exponential convergent behavior of the primal-dual gradient dynamics in terms of the Lyapunov method. To this end, let us consider the following constrained optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \quad f(\boldsymbol{\theta}) \quad \text{such that} \quad \boldsymbol{M}\boldsymbol{\theta} = \boldsymbol{0}_n, \tag{4}$$

where $\boldsymbol{\theta} \in \mathbb{R}^n$, $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ and $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable, smooth, and strongly convex function [Boyd and Vandenberghe, 2004]. One of the popular approaches for solving (4) is to formulate it into the saddle-point problem [Boyd and Vandenberghe, 2004], $L(\boldsymbol{\theta}, \boldsymbol{w}) = \min_{\boldsymbol{\theta} \in \mathbb{R}^n} \max_{\boldsymbol{w} \in \mathbb{R}^n} (f(\boldsymbol{\theta}) + \boldsymbol{w}^\top \boldsymbol{M} \boldsymbol{\theta})$, whose solution, $\boldsymbol{\theta}^*, \boldsymbol{w}^* \in \mathbb{R}^n$, exists and is unique when $\boldsymbol{M}$ has full-column rank [Qu and Li, 2018]. If $\boldsymbol{M}$ is rank-deficient, i.e., it is not full-column rank, there exists multiple $\boldsymbol{w}^*$ solving the saddle-point problem. It is known that its solution $\boldsymbol{\theta}^*, \boldsymbol{w}^*$ can be obtained by investigating the solution $\boldsymbol{\theta}_t, \boldsymbol{w}_t \in \mathbb{R}^n$ of the so-called primal-dual gradient dynamics [Qu and Li, 2018], with initial points $\boldsymbol{\theta}_0, \boldsymbol{w}_0 \in \mathbb{R}^n$,

$$\dot{\boldsymbol{\theta}}_t = -\nabla f(\boldsymbol{\theta}_t) - \boldsymbol{M}^\top \boldsymbol{w}_t, \quad \dot{\boldsymbol{w}}_t = \boldsymbol{M}\boldsymbol{\theta}_t.$$

[Qu and Li, 2018] studied exponential stability of the primal-dual gradient dynamics when $\boldsymbol{M}$ is full column-rank, using the classical Lyapunov approach [Sontag, 2013]. However, the proof relies on the invertibility of $\boldsymbol{M}$, and cannot be extended to the case when $\boldsymbol{M}$ is rank-deficient. As for such case, [Ozaslan and Jovanović, 2023; Cisneros-Velarde *et al.*, 2020; Gokhale *et al.*, 2023] proved exponential convergence to a particular solution $\boldsymbol{\theta}^*, \boldsymbol{w}^*$ using the tools based on singular value decomposition [Horn and Johnson, 2012]. In this paper, we will consider the following particular scenarios:

1. $\nabla f(\boldsymbol{\theta}_t) = \boldsymbol{U}\boldsymbol{\theta}_t$, where $\boldsymbol{U} \in \mathbb{R}^{n \times n}$, which is positive definite matrix, i.e., $\boldsymbol{U} + \boldsymbol{U}^\top \succ 0$;
2. $\boldsymbol{M}$ is symmetric and rank-deficient. Distributed algorithms are typical examples satisfying such condition and will be elaborated in subsequent sections.

We note that previous works considered general matrix $\boldsymbol{M}$, not necessarily a symmetric matrix. Moreover, note that the primal-dual gradient dynamics under such scenarios will appear in Section 4 as an ODE model of the proposed distributed TD-learning. The corresponding system can be rewritten as

$$\frac{d}{dt} \begin{bmatrix} \boldsymbol{\theta}_t \\ \boldsymbol{w}_t \end{bmatrix} = \begin{bmatrix} -\boldsymbol{U} & -\boldsymbol{M}^\top \\ \boldsymbol{M} & \boldsymbol{0}_{n \times n} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_t \\ \boldsymbol{w}_t \end{bmatrix}, \quad \boldsymbol{\theta}_0, \boldsymbol{w}_0 \in \mathbb{R}^n. \tag{5}$$

To study its exponential stability, let us introduce the Lyapunov function candidate $V(\boldsymbol{\theta}, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix}^\top \boldsymbol{S} \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix}$, where $\boldsymbol{S} \in \mathbb{R}^{2n \times 2n}$ is some symmetric positive definite matrix, and $\boldsymbol{\theta}, \boldsymbol{w} \in \mathbb{R}^n$. The candidate Lyapunov function considers projection of the iterate $\boldsymbol{w}_t$ to the range space of $\boldsymbol{M}$. As in previous works, the difficulty coming from singularity of $\boldsymbol{M}$ can be avoided by considering the range space and null space conditions of $\boldsymbol{M}$. In particular, [Ozaslan and Jovanović, 2023] employed a Lyapunov function that involves the gradient of the Lagrangian function, and considered the projected iterate $\boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w}_t$, where $\boldsymbol{M}\boldsymbol{M}^\dagger$ is the projection matrix onto range space of $\boldsymbol{M}$. [Cisneros-Velarde *et al.*, 2020] exploited a quadratic Lyapunov function in [Qu and Li, 2018] for the iterate $\boldsymbol{\theta}_t$ and $\boldsymbol{V}\boldsymbol{w}_t$, where $\boldsymbol{M} := \boldsymbol{T}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, which is the singular value decomposition of $\boldsymbol{M}$. [Gokhale *et al.*, 2023] considered a positive semi-definite matrix $\boldsymbol{S}$ and used semi-contraction theory [De Pasquale *et al.*, 2023] to prove exponential convergence of the primal-dual gradient dynamics.

We will adopt the quadratic Lyapunov function in [Qu and Li, 2018] with the projected iterate $\boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w}_t$, and leverage the symmetric property of $\boldsymbol{M}$ to show improved or comparable to the state of art convergence rate under the particular conditions newly imposed in this paper. When $\boldsymbol{M}$ is symmetric, the fact that the projection onto the column space of $\boldsymbol{M}$ and row space of $\boldsymbol{M}$ being identical simplifies the overall bounds. We first present the following Lyapunov inequality.

Lemma 2. *Let* $\boldsymbol{S} := \begin{bmatrix} \beta \boldsymbol{I}_n & \boldsymbol{M} \\ \boldsymbol{M} & \beta \boldsymbol{I}_n \end{bmatrix}$ *where* $\beta := \max\left\{ \frac{2\lambda_{\max}(\boldsymbol{M})^2 + 2 + \|\boldsymbol{U}\|_2^2}{\lambda_{\min}(\boldsymbol{U} + \boldsymbol{U}^\top)}, 4\lambda_{\max}(\boldsymbol{M}) \right\}$. *Then,* $\frac{\beta}{2} \boldsymbol{I}_{2n} \prec \boldsymbol{S} \prec 2\beta \boldsymbol{I}_{2n}$, *and we have, for any* $\boldsymbol{\theta}, \boldsymbol{w} \in \mathbb{R}^n$,

$$\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix}^\top \boldsymbol{S} \begin{bmatrix} -\boldsymbol{U} & -\boldsymbol{M} \\ \boldsymbol{M} & \boldsymbol{0}_{n \times n} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix} \leq -\min\{1, \lambda_{\min}^{+}(\boldsymbol{M})^2\} \left\| \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix} \right\|_2^2.$$

The proof is given in [1]Appendix Section C.1. Using the above Lemma 2, we can now prove the exponential stability of the ODE dynamics in (5).

Theorem 3. *Let* $V(\boldsymbol{\theta}, \boldsymbol{w}) = \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix}^\top \boldsymbol{S} \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w} \end{bmatrix}$.

[1] The Appendix can be found in https://arxiv.org/pdf/2310.00638

For $\boldsymbol{\theta}_0, \boldsymbol{w}_0 \in \mathbb{R}^n$ *and* $t \in \mathbb{R}^+$, *we have*

$$\begin{aligned} & V(\boldsymbol{\theta}_t, \boldsymbol{w}_t) \\ =& \mathcal{O}\left(\exp\left(\frac{-\min\{1, \lambda_{\min}^{+}(\boldsymbol{M})^2\}}{\max\left\{\frac{2\lambda_{\max}(\boldsymbol{M})^2+2+\|\boldsymbol{U}\|_2^2}{\lambda_{\min}(\boldsymbol{U}+\boldsymbol{U}^\top)}, 4\lambda_{\max}(\boldsymbol{M})\right\}} t\right)\right). \end{aligned}$$

The proof is given in Appendix Section C.2. We show that the above bound enjoys sharper or comparable to the state of the art convergence rate under particular conditions. With slight modifications, the Lyapunov function becomes identical to that of [Gokhale *et al.*, 2023]. However, we directly rely on classical Lyapunov theory [Khalil, 2015] rather than the result from semi-contraction theory [De Pasquale *et al.*, 2023] used in [Gokhale *et al.*, 2023]. The classical Lyapunov approach simplifies the proof steps compared to that of semi-contraction theory. The detailed comparative analysis is in Appendix Section D. The fact that $\boldsymbol{M}$ is symmetric and considering the projected iterate $\boldsymbol{M}\boldsymbol{M}^\dagger \boldsymbol{w}_t$, provides improved and comparable bound. Furthermore, as will be clear in Section 4, this enables us to extend the analysis to stochastic algorithms (TD-learning) without introducing involved analysis including (semi)-contraction theory or intricate Lyapunov function.

4 Distributed TD-Learning

In this section, we propose a new distributed TD-learning algorithm to solve (1) based on the result in [Wang and Elia, 2011]. In this scenario, each agent keeps its own parameter estimate $\boldsymbol{\theta}^i \in \mathbb{R}^q,\ 1 \le i \le N$, and the goal of each agent is to estimate the value function $v^\pi(s) \approx \boldsymbol{\phi}(s)^\top \boldsymbol{\theta}_c$ satisfying (1) (the value function associated with the global reward $\sum_{k=0}^{\infty} \gamma^k \frac{1}{N} \sum_{i=1}^{N} r^i$) under the assumption that each agent has access only to its local reward r^i. The parameter of each agent can be shared over the communication network whose structure is represented by the graph $\mathcal{G}$, i.e., agents can share their parameters only with their neighbors over the network to solve the global problem. The connections among the agents can be represented by graph Laplacian matrix [Anderson Jr and Morley, 1985], $\boldsymbol{L} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, which characterizes the graph $\mathcal{G}$, i.e., $[\boldsymbol{L}]_{ij} = -1$ if $(i,j) \in \mathcal{E}$ and $[\boldsymbol{L}]_{ij} = 0$ if $(i,j) \notin \mathcal{E}$, and $[\boldsymbol{L}]_{ii} = |\mathcal{N}_i|$ for $i \in \mathcal{V}$. Note that $\boldsymbol{L}$ is symmetric positive semi-definite matrix and $\boldsymbol{L}\boldsymbol{1}_{|\mathcal{S}|} = 0$. To proceed, let us first introduce a set of matrix notations:

$$\bar{\boldsymbol{L}} := \boldsymbol{L} \otimes \boldsymbol{I}_q, \quad \bar{\boldsymbol{D}}^\pi := \boldsymbol{I}_N \otimes \boldsymbol{D}^\pi, \quad \bar{\boldsymbol{P}}^\pi := \boldsymbol{I}_N \otimes \boldsymbol{P}^\pi,$$

$$\bar{\boldsymbol{R}}^\pi = \begin{bmatrix} (\boldsymbol{R}_1^\pi)^\top & (\boldsymbol{R}_2^\pi)^\top & \cdots & (\boldsymbol{R}_N^\pi)^\top \end{bmatrix}^\top, \ \bar{\boldsymbol{\Phi}} := \boldsymbol{I}_N \otimes \boldsymbol{\Phi},$$

$$\bar{\boldsymbol{A}} = \boldsymbol{I}_N \otimes \boldsymbol{A}, \ \ \bar{\boldsymbol{b}} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_N \end{bmatrix}, \ \ \bar{\boldsymbol{\theta}} = \begin{bmatrix} \boldsymbol{\theta}^1 \\ \boldsymbol{\theta}^2 \\ \vdots \\ \boldsymbol{\theta}^N \end{bmatrix}, \ \ \bar{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w}^1 \\ \boldsymbol{w}^2 \\ \vdots \\ \boldsymbol{w}^N \end{bmatrix},$$

where $\otimes$ denotes Kronecker product, and $\bar{\boldsymbol{w}}$ is another collection of learnable parameters $\{\boldsymbol{w}^i \in \mathbb{R}^q\}_{i=1}^N$, where $\boldsymbol{w}^i$ assigned to each agent i and $\boldsymbol{b}_i$ is defined in (3).

Meanwhile, [Wang and Elia, 2011] studied distributed optimization algorithms [Tsitsiklis, 1984] from the control system perspectives in continuous-time domain, which can be represented as an Lagrangian problem [Hestenes, 1969]. Compared to other distributed optimization algorithms [Nedic and Ozdaglar, 2009; Pu and Nedić, 2021], the method in [Wang and Elia, 2011] does not require any specific initialization, diminishing step-sizes, and doubly stochastic matrix that corresponds to the underlying communication graph. Due to these advantages, this framework has been further studied in [Hatanaka *et al.*, 2018; Bin *et al.*, 2022]. Inspired by [Wang and Elia, 2011], [Lee, 2023] developed a continuous-time distributed TD-learning algorithm. The analysis relies on Barbalat's lemma [Khalil, 2015], which makes extension to the non-asymptotic finite-time analysis difficult for its discrete-time counterpart. Moreover, they focus on the deterministic continuous-time algorithms. The corresponding discrete-time distributed TD-learning is summarized in Algorithm 1, where each agent updates its local parameter using the local TD-error in (6). The updates in (7) and (8) in Algorithm 1 can be obtained by discretizing the continuous-time ODE introduced in [Wang and Elia, 2011] with stochastic samples.

Algorithm 1 Distributed TD-learning

```
Initialize $\alpha_0 \in (0,1), \{\boldsymbol{\theta}_0^i, \boldsymbol{w}_0^i \in \mathbb{R}^q\}_{i=1}^N, \eta \in (0,\infty)$.
for $k = 1, 2, \dots, T$ do
  for $i = 1, 2, \dots, N$ do
    Agent $i$ observes $o_k^i := (s_k, s_k', r_k^i)$.
    Update as follows:
      $\delta(o_k^i;\boldsymbol{\theta}_k^i) = r_k^i + \gamma\boldsymbol{\phi}^\top(s_k')\boldsymbol{\theta}_k^i - \boldsymbol{\phi}^\top(s_k)\boldsymbol{\theta}_k^i$   (6)
      $\boldsymbol{\theta}_{k+1}^i = \boldsymbol{\theta}_k^i + \alpha_k(\delta(o_k^i;\boldsymbol{\theta}_k^i)\boldsymbol{\phi}(s_k)$
          $- \eta(|\mathcal{N}_i|\boldsymbol{\theta}_k^i - \sum_{j\in\mathcal{N}_i}\boldsymbol{\theta}_k^j)$
          $- \eta(|\mathcal{N}_i|\boldsymbol{w}_k^i - \sum_{j\in\mathcal{N}_i}\boldsymbol{w}_k^j))$   (7)
      $\boldsymbol{w}_{k+1}^i = \boldsymbol{w}_k^i + \alpha_k\eta(|\mathcal{N}_i|\boldsymbol{\theta}_k^i - \sum_{j\in\mathcal{N}_i}\boldsymbol{\theta}_k^j)$   (8)
  end for
end for
```

Using the stacked vector representation, the updates in (7) and (8) in Algorithm 1 can be rewritten in compact form:

$$\begin{aligned} \begin{bmatrix} \bar{\boldsymbol{\theta}}_{k+1} \\ \bar{\boldsymbol{w}}_{k+1} \end{bmatrix} =& \begin{bmatrix} \bar{\boldsymbol{\theta}}_{k} \\ \bar{\boldsymbol{w}}_{k} \end{bmatrix} + \alpha_k \begin{bmatrix} \bar{\boldsymbol{A}} - \eta\bar{\boldsymbol{L}} & -\eta\bar{\boldsymbol{L}} \\ \eta\bar{\boldsymbol{L}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{\theta}}_{k} \\ \bar{\boldsymbol{w}}_{k} \end{bmatrix} \\ & + \alpha_k \begin{bmatrix} \bar{\boldsymbol{b}} \\ \boldsymbol{0} \end{bmatrix} + \alpha_k \bar{\boldsymbol{\epsilon}}(o_k;\bar{\boldsymbol{\theta}}_k), \end{aligned} \tag{9}$$

where, $o_k := \{o_k^i\}_{i=1}^N$, and for $1 \le i \le N$,

$$\begin{aligned} \boldsymbol{\epsilon}^i(o_k^i;\boldsymbol{\theta}_k^i) :=& \delta(o_k^i;\boldsymbol{\theta}_k^i)\boldsymbol{\phi}(s_k) - \boldsymbol{A}\boldsymbol{\theta}_k^i - \boldsymbol{b}^i, \\ \bar{\boldsymbol{\epsilon}}(o_k;\bar{\boldsymbol{\theta}}_k) :=& \begin{bmatrix} \boldsymbol{\epsilon}_k^{1\top} & \boldsymbol{\epsilon}_k^{2\top} & \cdots & \boldsymbol{\epsilon}_k^{N\top} & \boldsymbol{0}^\top \end{bmatrix}^\top, \end{aligned} \tag{10}$$

where we denoted $\boldsymbol{\epsilon}_k^i := \boldsymbol{\epsilon}^i(o_k^i;\boldsymbol{\theta}_k^i)$. Note that the superscript of $\boldsymbol{\epsilon}_k^i$ corresponds to the i-th agent. Compared to the continuous-time algorithm in [Lee, 2023], we introduce an additional positive variable $\eta > 0$ multiplied with the graph Laplacian matrix, which results in the factor η multiplied with the mixing part in Algorithm 1 in order to control the variance of the update. We note that when the the number of neighbors of an agent $i \in \mathcal{V}$ is large, then so is the variance of the corresponding updates of the agent. In this case, the variance can be controlled by adjusting η to be small.

The behavior of stochastic algorithm is known to be closely related to its continuous-time O.D.E. counterpart [Borkar and Meyn, 2000; Srikant and Ying, 2019]. In this respect, the corresponding O.D.E. model of (9) is given by

$$\frac{d}{dt}\begin{bmatrix}\bar{\boldsymbol{\theta}}_t\\ \bar{\boldsymbol{w}}_t\end{bmatrix} = \begin{bmatrix}\bar{\boldsymbol{A}}-\eta\bar{\boldsymbol{L}} & -\eta\bar{\boldsymbol{L}}\\ \eta\bar{\boldsymbol{L}} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{\theta}}_t\\ \bar{\boldsymbol{w}}_t\end{bmatrix} + \begin{bmatrix}\bar{\boldsymbol{b}}\\ \boldsymbol{0}\end{bmatrix}, \tag{11}$$

for $\bar{\boldsymbol{\theta}}_0, \bar{\boldsymbol{w}}_0 \in \mathbb{R}^{Nq}$, and $t \in \mathbb{R}^+$. The above linear system is closely related to the primal-dual gradient dynamics in (5) in Section 3. Compared to (5), the difference lies in the fact that the above system corresponds to the the dynamics of the distributed TD-learning represented by matrix $\bar{\boldsymbol{A}}$ instead of the gradient of a particular objective function. It is straightforward to check that the equilibrium point of the above system is $\boldsymbol{1}_N \otimes \boldsymbol{\theta}_c$ and $\frac{1}{\eta}\bar{\boldsymbol{w}}_\infty$ such that $\bar{\boldsymbol{L}}\bar{\boldsymbol{w}}_\infty = \bar{\boldsymbol{A}}(\boldsymbol{1}_N \otimes \boldsymbol{\theta}_c) + \bar{\boldsymbol{b}}$.

In what follows, we will analyze finite-time behavior of (9) based on the Lyapunov equation in Lemma 4. For the analysis, we will follow the spirit of [Srikant and Ying, 2019], which studied the standard single-agent TD-learning based on the Lyapunov method [Sontag, 2013]. To proceed further, let us consider the coordinate change of $\tilde{\boldsymbol{\theta}}_k := \bar{\boldsymbol{\theta}}_k - \boldsymbol{1}_N \otimes \boldsymbol{\theta}_c$ and $\tilde{\boldsymbol{w}}_k := \bar{\boldsymbol{w}}_k - \frac{1}{\eta}\bar{\boldsymbol{w}}_\infty$, with which we can rewrite (9) by

$$\begin{bmatrix}\tilde{\boldsymbol{\theta}}_{k+1}\\ \tilde{\boldsymbol{w}}_{k+1}\end{bmatrix} = \begin{bmatrix}\tilde{\boldsymbol{\theta}}_k\\ \tilde{\boldsymbol{w}}_k\end{bmatrix} + \alpha_k\begin{bmatrix}\bar{\boldsymbol{A}}-\eta\bar{\boldsymbol{L}} & -\eta\bar{\boldsymbol{L}}\\ \eta\bar{\boldsymbol{L}} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\theta}}_k\\ \tilde{\boldsymbol{w}}_k\end{bmatrix} + \alpha_k\bar{\boldsymbol{\epsilon}}(o_k;\bar{\boldsymbol{\theta}}_k). \tag{12}$$

We will now derive a Lyapunov inequality for the above system based on the results in Lemma 4, To this end, we will rely on the analysis in [Qu and Li, 2018], which proved exponential convergence of the continuous-time primal-dual gradient dynamics based on the Lyapunov method. However, the newly introduced singularity of $\bar{\boldsymbol{L}}$ imposes difficulty in directly applying the results from [Qu and Li, 2018] which does not allow the singularity. To overcome this difficulty, we will multiply $\bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger$ to the dual update $\tilde{\boldsymbol{w}}_{k+1}$ in (12), which is the projection to the range space of $\bar{\boldsymbol{L}}$. The symmetric assumption of $\bar{\boldsymbol{L}}$ helps to construct an explicit solution of the Lyapunov inequality in Lemma 4. Multiplying $\bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger$ to $\tilde{\boldsymbol{w}}_{k+1}$ in (12) yields

$$\begin{bmatrix}\tilde{\boldsymbol{\theta}}_{k+1}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}_{k+1}\end{bmatrix} = \left(\boldsymbol{I}_{2N} + \alpha_k\begin{bmatrix}\bar{\boldsymbol{A}}-\eta\bar{\boldsymbol{L}} & -\eta\bar{\boldsymbol{L}}\\ \eta\bar{\boldsymbol{L}} & \boldsymbol{0}\end{bmatrix}\right)\begin{bmatrix}\tilde{\boldsymbol{\theta}}_k\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}_k\end{bmatrix} + \alpha_k\bar{\boldsymbol{\epsilon}}_k(o_k;\bar{\boldsymbol{\theta}}_k), \tag{13}$$

which can be proved using Lemma 2 in the Appendix C. For this system, we now derive the following Lyapunov inequality.

Lemma 4. *There exists a positive symmetric definite matrix* $\boldsymbol{G} \in \mathbb{R}^{2Nq\times 2Nq}$ *such that* $\frac{8+\eta+4\eta^2\lambda_{\max}(\bar{\boldsymbol{L}})^2}{2\eta(1-\gamma)w}\boldsymbol{I}_{2Nq} \prec \boldsymbol{G} \prec 2\frac{8+\eta+4\eta^2\lambda_{\max}(\bar{\boldsymbol{L}})^2}{\eta(1-\gamma)w}\boldsymbol{I}_{2Nq}$, *and for* $\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{w}} \in \mathbb{R}^{Nq}$,

$$2\begin{bmatrix}\tilde{\boldsymbol{\theta}}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}\end{bmatrix}^\top \boldsymbol{G}\begin{bmatrix}\bar{\boldsymbol{A}}-\eta\bar{\boldsymbol{L}} & -\eta\bar{\boldsymbol{L}}\\ \eta\bar{\boldsymbol{L}} & \boldsymbol{0},\end{bmatrix}\begin{bmatrix}\tilde{\boldsymbol{\theta}}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}\end{bmatrix} \leq -\min\{1,\eta\lambda^+_{\min}(\bar{\boldsymbol{L}})^2\}\left\|\begin{bmatrix}\tilde{\boldsymbol{\theta}}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}\end{bmatrix}\right\|_2^2.$$

The proof is given in Appendix Section D.1. The proof can be done by noting that $\bar{\boldsymbol{A}} - \eta\bar{\boldsymbol{L}}$ is negative semi-definite and $\bar{\boldsymbol{L}}$ is rank-deficient, and applying Lemma 2.

4.1 I.I.D. Observation Case

We are now in position to provide the first main result, a finite-time analysis of Algorithm 1 under the i.i.d. observation model, which is a common assumption in the literature, and provides simple and clean theoretical insights.

Theorem 5. *1. Suppose we use constant step-size* $\alpha_0 = \alpha_1 = \cdots = \alpha_k$ *for* $k \in \mathbb{N}_0$, *and* $\alpha_0 \leq \bar{\alpha}$ *for some positive constant* $\bar{\alpha} \in (0,1)$. *Then, we have*

$$\frac{1}{N}\mathbb{E}\left[\left\|\begin{bmatrix}\tilde{\boldsymbol{\theta}}_{k+1}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}_{k+1}\end{bmatrix}\right\|_2^2\right] = \mathcal{O}\left(\exp\left(-(1-\gamma)w\frac{\min\{1,\eta\lambda^+_{\min}(\bar{\boldsymbol{L}})^2\}}{\frac{8}{\eta}+4\eta\lambda_{\max}(\bar{\boldsymbol{L}})^2}k\alpha_0\right)\right) + \mathcal{O}\left(\alpha_0\frac{R^2_{\max}}{w^3(1-\gamma)^3}\frac{2+\eta^2\lambda_{\max}(\bar{\boldsymbol{L}})^2}{\eta\min\{1,\eta\lambda_{\min}(\bar{\boldsymbol{L}})^2\}}\right).$$

2. Suppose we have $\alpha_k = \frac{h_1}{k+h_2}$. *There exist* $\bar{h}_1$ *and* $\bar{h}_2$ *such that letting* $h_1 = \Theta(\bar{h}_1)$ *and* $h_2 = \Theta(\bar{h}_2)$ *yields*

$$\frac{1}{N}\mathbb{E}\left[\left\|\begin{bmatrix}\tilde{\boldsymbol{\theta}}_{k+1}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}_{k+1}\end{bmatrix}\right\|_2^2\right] = \mathcal{O}\left(\frac{1}{k}\frac{(2+\eta^2\lambda_{\max}(\bar{\boldsymbol{L}})^2)^2}{\eta^2\min\{1,\eta\lambda^+_{\min}(\bar{\boldsymbol{L}})^2\}^2}\frac{R^2_{\max}}{w^4(1-\gamma)^4}\right).$$

The proof and the exact constants can be found in Appendix Section E.1. Using constant step-size, we can guarantee exponential convergence rate with small bias term $\mathcal{O}\left(\alpha_0\frac{R^2_{\max}\lambda_{\max}(\bar{\boldsymbol{L}})}{w^3(1-\gamma)^3}\right)$ when $\eta \approx \frac{\sqrt{2}}{\lambda_{\max}(\bar{\boldsymbol{L}})}$ and $\lambda^+_{\min}(\bar{\boldsymbol{L}})^2 \geq \sqrt{2}\lambda_{\max}(\bar{\boldsymbol{L}})$. Appropriate choice of η allows wider range of step-size, and this will be clear in the experimental results in Section 5. Furthermore, the algorithm's performance is closely tied to the properties of the graph structure. $\lambda^+_{\min}(\bar{\boldsymbol{L}})$, the smallest non-zero eigenvalue of graph Laplacian, characterizes the connectivity of the graph [Chung, 1997], and a graph with lower connectivity will yield slower convergence rate and larger bias. $\lambda_{\max}(\bar{\boldsymbol{L}})$ is the largest eigenvalue of the graph Laplacian, and it can be upper bounded by twice the maximum degree of the graph [Anderson Jr and Morley, 1985]. That is, a graph with higher maximum degree could incur slower convergence rate and larger bias. However, compared to $\lambda^+_{\min}(\boldsymbol{M})$, we experimentally verify in Section 5 that $\lambda_{\max}(\bar{\boldsymbol{L}})$ does not appear to be an important factor under particular cases, and there could exist a tighter bound without $\lambda_{\max}(\bar{\boldsymbol{L}})$. As for diminishing step-size, we achieve $\mathcal{O}\left(\frac{1}{k}\right)$ convergence rate from the second item in Theorem 5, and similar observations hold as in the constant step-size, i.e., the convergence rate depends on the smallest non-zero and maximum eigenvalue of graph Laplacian. Lastly, as in [Wang *et al.*, 2020], our bound does not explicitly depend on the number of agents, N, compared to the bound in [Doan *et al.*, 2019] and [Sun *et al.*, 2020], where the bound scales at the order of N.

Furthermore, the known constant error bound for (single-agent) TD-learning, which is Theorem 2 of [Bhandari *et al.*, 2018] is $O\left(\frac{1}{(1-\gamma)^4 w^2}\right)$. Meanwhile our bound in Theorem 4.2 is $O\left(\frac{1}{(1-\gamma)^3 w^3}\right)$ for the constant step-size case. The difference only comes from the choice on the bound in θ_c, the solution of the Bellman equation. We use the bound $\|\theta_c\|_2 \leq O\left(\frac{1}{(1-\gamma)w}\right)$ in Lemma 6 in Appendix C, whereas the bound $O\left(\frac{1}{(1-\gamma)^{\frac{3}{2}} w^{\frac{1}{2}}}\right)$ is used in [Bhandari *et al.*, 2018].

4.2 Markovian Observation Case

In this section, we consider the Markovian observation model, where the sequence of observations $\{s_k\}_{k=1}^{\infty}$ follows a Markov chain. Compared to the i.i.d. observation model, the correlation between the observation and the updated iterates imposes difficulty in the analysis. To overcome this issue, an assumption on the Markov chain that ensures a geometric mixing property is helpful. In particular, the so-called ergodic Markov chain can be characterized by the metric called total variation distance [Levin and Peres, 2017], $d_{\mathrm{TV}}(P,Q) = \frac{1}{2}\sum_{x\in\mathcal{S}} |P(x) - Q(x)|$, where P and Q is probability measure on $\mathcal{S}$. A Markov chain is said to be ergodic if it is irreducible and aperiodic [Levin and Peres, 2017]. An ergodic Markov chain is known to converge to its unique stationary exponentially fast, i.e., for $k \in \mathbb{N}_0$, $\sup_{1\leq i\leq|\mathcal{S}|} d_{\mathrm{TV}}(\boldsymbol{e}_i^\top(\boldsymbol{P}^\pi)^k, \mu_\infty) \leq m\rho^k$, where $\boldsymbol{e}_i \in \mathbb{R}^{|\mathcal{S}|}$ for $1 \leq i \leq N$ is the $|\mathcal{S}|$-dimensional vector whose i-th element is one and others are zero, $\mu_\infty \in \mathbb{R}^{|\mathcal{S}|}$ is the stationary distribution of the Markov chain induced by transition matrix $\boldsymbol{P}^\pi$, $m \in \mathbb{R}$ is a positive constant, and $\rho \in (0,1)$. The assumption on the geometric mixing property of the Markov chain is common in the literature [Srikant and Ying, 2019; Wang *et al.*, 2020]. The mixing time of Markov chain is an important quantity of a Markov chain, defined as

$$\tau(\delta) := \min\{k \in \mathbb{N} \mid m\rho^k \leq \delta\}. \tag{14}$$

For simplicity, we will use $\tau := \tau(\alpha_T)$, where $T \in \mathbb{N}_0$ denotes the total number of iterations, and α_k, is the step-size at k-th iteration. If we use the step-size $\alpha_k = \frac{1}{1+k}$, the mixing time τ only contributes to the logarithmic factor, $\log T$ in the finite-time bound [Bhandari *et al.*, 2018]. As in the proof of i.i.d. case, using the Lypaunov argument in Lemma 4, we can prove the finite-time bound on the mean-squared error, following the spirit of [Srikant and Ying, 2019]. To simplify the proof, we will investigate the case $\eta = 1$.

Theorem 6. *1. Suppose we use constant step-size $\alpha_0 = \alpha_1 = \cdots = \alpha_T$ such that $\alpha_0 \leq \bar{\alpha}$ for some positive constant $\bar{\alpha} \in (0,1)$. Then, we have, for $\tau \leq k \leq T$,*

$$\frac{1}{N}\mathbb{E}\left[\left\|\begin{bmatrix}\tilde{\boldsymbol{\theta}}_{k+1}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}_{k+1}\end{bmatrix}\right\|_2^2\right] =$$

$$\mathcal{O}\left(\exp\left(-\frac{(1-\gamma)w\min\{1,\lambda_{\min}^+(\boldsymbol{L})^2\}}{\lambda_{\max}(\boldsymbol{L})^2}\alpha_0(k-\tau)\right)\right)$$

$$+\,\mathcal{O}\left(\alpha_0\tau\frac{R_{\max}^2}{w^3(1-\gamma)^3}\frac{\lambda_{\max}(\boldsymbol{L})^2}{\min\{1,\lambda_{\min}^+(\boldsymbol{L})^2\}}\right).$$

2. Considering diminishing step-size, with $\alpha_k = \frac{h_1}{k+h_2}$ for $k \in \mathbb{N}_0$, there exits $\bar{h}_1$ and $\bar{h}_2$ such that for $h_1 = \Theta(\bar{h}_1)$ and $h_2 = \Theta(\bar{h}_2)$, we have for $\tau \leq k \leq T$,

$$\frac{1}{N}\mathbb{E}\left[\left\|\begin{bmatrix}\tilde{\boldsymbol{\theta}}_{k+1}\\ \bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^\dagger\tilde{\boldsymbol{w}}_{k+1}\end{bmatrix}\right\|_2^2\right]$$

$$=\mathcal{O}\left(\frac{\tau}{k}\frac{qR_{\max}^2}{w^4(1-\gamma)^4}\frac{\lambda_{\max}(\boldsymbol{L})^5}{\min\{1,\lambda_{\min}^+(\boldsymbol{L})^2\}^2}\right).$$

The proof and the exact values can be found in Appendix F.1. For the constant step-size, we can see that the bounds have additional mixing time factors compared to the i.i.d. case. Considering diminishing step-size, the convergence rate of $\mathcal{O}\left(\frac{\tau}{k}\right)$ can be verified, incorporating a multiplication by the mixing time τ.

As summarized in Table 1, the proposed distributed TD-learning does not require doubly stochastic matrix or any specific initializations. The algorithms requiring the doubly stochastic matrix, whose definition is given in Appendix B, face challenges when extending to directed graph and time-varying graph scenarios. However, our algorithm does not require major modifications. Meanwhile, push-sum [Nedić and Olshevsky, 2014] or push-pull [Pu *et al.*, 2020] algorithms have been developed to cope with the assumption of doubly stochastic matrix in directed graph scenario. Nonetheless, both methods require knowledge of out-degree, which are often difficult to know in presence including broadcast communications [Hendrickx and Tsitsiklis, 2015]. Moreover, the performance of the algorithm is sensitive to the choice of doubly stochastic matrix as can be seen in Appendix G.

5 Experiments

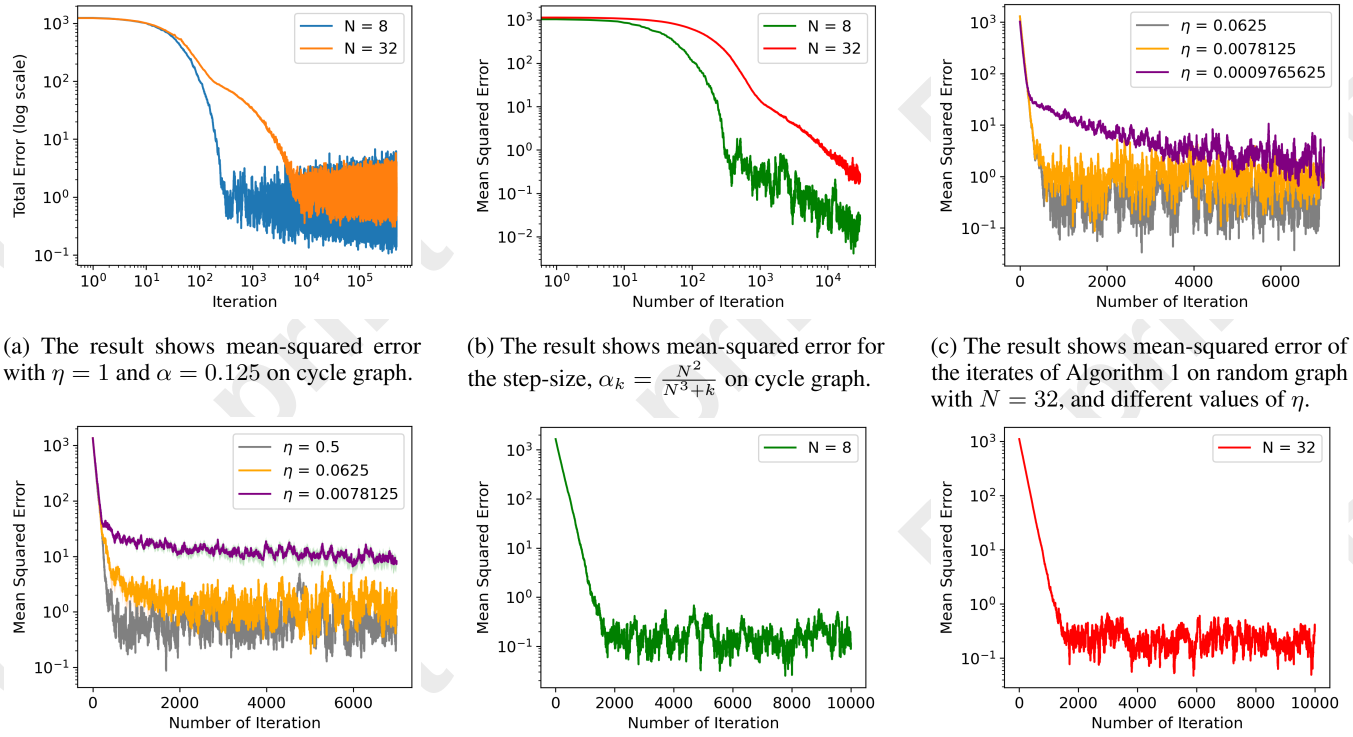
[2]This section provides the experimental results of Algorithm 1. First, we give an explanation of the MAMDP setup, where the number of states is three and the dimension of the feature is two. An agent can transit to every state with uniform probability. The feature matrix is set as $\boldsymbol{\Phi}^\top = \begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}$. The rewards are generated uniformly random between the interval $(0,10)$. The discount factor is set as 0.8.

For each experiment with $N \in \{2^3, 2^5\}$ number of agents, we construct a cycle, a graph $\mathcal{G}$ consisting of $\mathcal{V} := \{1,2,\ldots,N\}$ and $\mathcal{E} := \{(i,i+1)\}_{i=1}^{N-1} \cup \{(N,1)\}$. The smallest non-zero eigenvalue of graph Laplacian corresponding to a cycle with even number of vertices decreases as the number of vertices increases, while maximum eigenvalue remains same. The smallest non-zero eigenvalue is $2 - 2\cos\left(\frac{2\pi}{N}\right)$, and the largest eigenvalue is four [Mohar, 1997]. As N gets larger, the smallest non-zero eigenvalue gets smaller, which becomes 0.59 and 0.04 for $N = 2^3, 2^5$, respectively. Therefore, as number of agents increases, the convergence rate will be slower as expected in Theorem 5, and this can be verified in Figure (1a) and Figure (3) in the Appendix. The plots show the result for constant step-size $\alpha_0 \in \{2^{-3}, 2^{-4}, 2^{-5}, 2^{-6}\}$. Moreover, the convergence under a diminishing step-size can

[2] The code is provided in this link.

	Method	Observation model	Step-size	Requirement	Doubly stochastic matrix
[Doan *et al.*, 2019]	[Nedic and Ozdaglar, 2009]	i.i.d.	Constant/$\frac{1}{\sqrt{k+1}}$	Projection	✓
[Doan *et al.*, 2021]	[Nedic and Ozdaglar, 2009]	Markovian	Constant/$\frac{h_1}{k+1}$	✗	✓
[Sun *et al.*, 2020]	[Nedic and Ozdaglar, 2009]	i.i.d./Markovian	Constant	✗	✓
[Zeng *et al.*, 2022]	[Nedic and Ozdaglar, 2009]	i.i.d./Markovian	Constant	✗	✓
[Wang *et al.*, 2020]	[Pu and Nedić, 2021]	i.i.d./Markovian	Constant	Specific initialization	✓
Ours	[Wang and Elia, 2011]	i.i.d./Markovian	Constant/$\frac{h_1}{k+h_2}$	✗	✗

Table 1: Comparison with existing works.



(a) The result shows mean-squared error with $\eta = 1$ and $\alpha = 0.125$ on cycle graph.

(b) The result shows mean-squared error for the step-size, $\alpha_k = \frac{N^2}{N^3+k}$ on cycle graph.

(c) The result shows mean-squared error of the iterates of Algorithm 1 on random graph with $N = 32$, and different values of η.

(d) $N = 8$, $\alpha = 0.1$ on random graph with different values of η. If $\eta = 2$, the algorithm diverges.

(e) Mean squared error of Algorithm 1 on star graph after 10,000 iterations. We set $\eta = 1$ with step-size $1/2^5$.

(f) Mean squared error of Algorithm 1 on star graph after 10,000 iterations. We set $\eta = 1$ with step-size $1/2^5$.

Figure 1: Experiment results of Algorithm 1. The experiments were averaged over 50 runs.

be seen in Figure (1b). To investigate the effect of $\lambda_{\max}(\bar{\boldsymbol{L}})$, we construct a star graph, where one vertex has degree $N-1$ and the others have degree one. The maximum eigenvalue of star graph is N and the smallest non-zero eigenvalue is one [Nica, 2016]. Even though N gets larger, we could see in Figure (1e) and (1f) that the convergence rate or bias term does not vary. Therefore, we can expect that there could be a tighter bound without $\lambda_{\max}(\bar{\boldsymbol{L}})$ under particular cases.

To verify the effect of η, we use a random graph model [Erdős *et al.*, 1960], where among possible $N(N-1)/2$ edges, $(N-3)(N-4)/2$ edges are randomly selected. Figure (1c) shows the evolution of the mean squared error for $N = 32$, and step-size 0.1 with different η values. When $\eta = 0.5$ or $\eta = 1$, the algorithm diverges. Moreover, the bias gets smaller around $\frac{\sqrt{2}}{\lambda_{\max}(\boldsymbol{L})} \approx 0.046$. This implies that appropriate choice of η can control the variance when the number of neighbors is large but if η is too small or large, Algorithm 1 may cause divergence or large bias. This matches the result of the bound in Theorem 5. Similar arguments hold when $N = 8$, and the result is given in Figure (1d).

Lastly, the comparison with other algorithms are given in Appendix G. In summary, while no single algorithm consistently outperforms the others, the performance of methods that rely on the doubly stochastic matrix is highly sensitive to the choice of this matrix.

6 Conclusion

In this study, we have studied primal-dual gradient dynamics subject to some null-space constraints and its application to a distributed TD-learning. We have derived finite-time bounds for both the gradient dynamics and the distributed TD-learning. The results have been experimentally demonstrated. Future studies include extending the analysis to distributed TD-learning with nonlinear function approximation.

Acknowledgements

The work was supported by the Institute of Information Communications Technology Planning Evaluation (IITP) funded by the Korea government under Grant 2022-0-00469.

References

[Anderson Jr and Morley, 1985] William N Anderson Jr and Thomas D Morley. Eigenvalues of the Laplacian of a graph. *Linear and multilinear algebra*, 18(2):141–145, 1985.

[Arrow *et al.*, 1958] Kenneth Joseph Arrow, Leonid Hurwicz, and Hollis Burnley Chenery. Studies in linear and nonlinear programming. *(No Title)*, 1958.

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

[Bin *et al.*, 2022] Michelangelo Bin, Ivano Notarnicola, and Thomas Parisini. Stability, Linear Convergence, and Robustness of the Wang-Elia Algorithm for Distributed Consensus Optimization. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1610–1615. IEEE, 2022.

[Borkar and Meyn, 2000] Vivek S Borkar and Sean P Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM Journal on Control and Optimization*, 38(2):447–469, 2000.

[Boyd and Vandenberghe, 2004] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Cassano *et al.*, 2020] Lucas Cassano, Kun Yuan, and Ali H Sayed. Multiagent fully decentralized value function learning with linear convergence rates. *IEEE Transactions on Automatic Control*, 66(4):1497–1512, 2020.

[Chung, 1997] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[Cisneros-Velarde *et al.*, 2020] Pedro Cisneros-Velarde, Saber Jafarpour, and Francesco Bullo. Distributed and time-varying primal-dual dynamics via contraction analysis. *arXiv preprint arXiv:2003.12665*, 2020.

[De Pasquale *et al.*, 2023] Giulia De Pasquale, Kevin D Smith, Francesco Bullo, and M Elena Valcher. Dual seminorms, ergodic coefficients and semicontraction theory. *IEEE Transactions on Automatic Control*, 2023.

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

[Doan *et al.*, 2021] Thinh T Doan, Siva Theja Maguluri, and Justin Romberg. Finite-time performance of distributed temporal-difference learning with linear function approximation. *SIAM Journal on Mathematics of Data Science*, 3(1):298–320, 2021.

[Erdős *et al.*, 1960] Paul Erdős, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60, 1960.

[Gokhale *et al.*, 2023] Anand Gokhale, Alexander Davydov, and Francesco Bullo. Contractivity of Distributed Optimization and Nash Seeking Dynamics. *arXiv preprint arXiv:2309.05873*, 2023.

[Hatanaka *et al.*, 2018] Takeshi Hatanaka, Nikhil Chopra, Takayuki Ishizaki, and Na Li. Passivity-based distributed optimization with communication delays using PI consensus algorithm. *IEEE Transactions on Automatic Control*, 63(12):4421–4428, 2018.

[Hendrickx and Tsitsiklis, 2015] Julien M Hendrickx and John N Tsitsiklis. Fundamental limitations for anonymous distributed systems with broadcast communications. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing*, pages 9–16. IEEE, 2015.

[Hestenes, 1969] Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[Horn and Johnson, 2012] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[Khalil, 2015] Hassan K Khalil. *Nonlinear Control*. Pearson Education, 2015.

[Lee and Kim, 2022] Donghwan Lee and Do Wan Kim. Analysis of Temporal Difference Learning: Linear System Approach. *arXiv preprint arXiv:2204.10479*, 2022.

[Lee *et al.*, 2018] Donghwan Lee, Hyungjin Yoon, and Naira Hovakimyan. Primal-dual algorithm for distributed reinforcement learning: Distributed GTD. In *2018 IEEE Conference on Decision and Control*, pages 1967–1972. IEEE, 2018.

[Lee *et al.*, 2022] Donghwan Lee, Han-Dong Lim, Jihoon Park, and Okyong Choi. New Versions of Gradient Temporal Difference Learning. *IEEE Transactions on Automatic Control*, 2022.

[Lee, 2023] Donghwan Lee. Distributed Dynamic Programming and an ODE Framework of Distributed TD-Learning for Networked Multi-Agent Markov Decision Processes. *arXiv e-prints*, pages arXiv–2307, 2023.

[Levin and Peres, 2017] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[Macua *et al.*, 2014] Sergio Valcarcel Macua, Jianshu Chen, Santiago Zazo, and Ali H Sayed. Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, 60(5):1260–1274, 2014.

[Mathkar and Borkar, 2016] Adwaitvedant Mathkar and Vivek S Borkar. Distributed reinforcement learning via gossip. *IEEE Transactions on Automatic Control*, 62(3):1465–1470, 2016.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[Mohar, 1997] Bojan Mohar. Some applications of Laplace eigenvalues of graphs. In *Graph symmetry: Algebraic methods and applications*, pages 225–275. Springer, 1997.

[Nedić and Olshevsky, 2014] Angelia Nedić and Alex Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3):601–615, 2014.

[Nedic and Ozdaglar, 2009] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[Nica, 2016] Bogdan Nica. A brief introduction to spectral graph theory. *arXiv preprint arXiv:1609.08072*, 2016.

[Notarnicola *et al.*, 2023] Ivano Notarnicola, Michelangelo Bin, Lorenzo Marconi, and Giuseppe Notarstefano. The Gradient Tracking Is a Distributed Integral Action. *IEEE Transactions on Automatic Control*, 2023.

[Ozaslan and Jovanović, 2023] Ibrahim K Ozaslan and Mihailo R Jovanović. On the global exponential stability of primal-dual dynamics for convex problems with linear equality constraints. In *2023 American Control Conference (ACC)*, pages 210–215. IEEE, 2023.

[Pu and Nedić, 2021] Shi Pu and Angelia Nedić. Distributed stochastic gradient tracking methods. *Mathematical Programming*, 187:409–457, 2021.

[Pu *et al.*, 2020] Shi Pu, Wei Shi, Jinming Xu, and Angelia Nedić. Push–pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 66(1):1–16, 2020.

[Qu and Li, 2018] Guannan Qu and Na Li. On the exponential stability of primal-dual gradient dynamics. *IEEE Control Systems Letters*, 3(1):43–48, 2018.

[Robbins and Monro, 1951] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[Sontag, 2013] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media, 2013.

[Srikant and Ying, 2019] Rayadurgam Srikant and Lei Ying. Finite-time error bounds for linear stochastic approximation and td learning. In *Conference on Learning Theory*, pages 2803–2830. PMLR, 2019.

[Stanković *et al.*, 2023] Miloš S Stanković, Marko Beko, and Srdjan S Stanković. Distributed consensus-based multi-agent temporal-difference learning. *Automatica*, 151:110922, 2023.

[Sun *et al.*, 2020] Jun Sun, Gang Wang, Georgios B Giannakis, Qinmin Yang, and Zaiyue Yang. Finite-time analysis of decentralized temporal-difference learning with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, pages 4485–4495. PMLR, 2020.

[Sutton *et al.*, 2009] Richard S Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th annual international conference on machine learning*, pages 993–1000, 2009.

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

[Tsitsiklis and Van Roy, 1996] John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

[Tsitsiklis, 1984] John N Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[Wai *et al.*, 2018] Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

[Wang and Elia, 2011] Jing Wang and Nicola Elia. A control perspective for centralized and distributed convex optimization. In *2011 50th IEEE conference on decision and control and European control conference*, pages 3800–3805. IEEE, 2011.

[Wang *et al.*, 2019] Gang Wang, Bingcong Li, and Georgios B Giannakis. A multistep Lyapunov approach for finite-time analysis of biased stochastic approximation. *arXiv preprint arXiv:1909.04299*, 2019.

[Wang *et al.*, 2020] Gang Wang, Songtao Lu, Georgios Giannakis, Gerald Tesauro, and Jian Sun. Decentralized TD tracking with linear function approximation and its finite-time analysis. *Advances in Neural Information Processing Systems*, 33:13762–13772, 2020.

[Wang *et al.*, 2022] Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5064–5078, 2022.

[Zeng *et al.*, 2022] Sihan Zeng, Thinh T Doan, and Justin Romberg. Finite-Time Convergence Rates of Decentralized Stochastic Approximation With Applications in Multi-Agent and Multi-Task Learning. *IEEE Transactions on Automatic Control*, 2022.

[Zhou and Luo, 2018] Bin Zhou and Weiwei Luo. Improved Razumikhin and Krasovskii stability criteria for time-varying stochastic time-delay systems. *Automatica*, 89:382–391, 2018.