Multi-Agent Reinforcement Learning via Double Averaging Primal-Dual Optimization

Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, Mingyi Hong*

August 21, 2018

Abstract

Despite the success of single-agent reinforcement learning, multi-agent reinforcement learning (MARL) remains challenging due to complex interactions between agents. Motivated by decentralized applications such as sensor networks, swarm robotics, and power grids, we study policy evaluation in MARL, where agents with jointly observed state-action pairs and private local rewards collaborate to learn the value of a given policy.

In this paper, we propose a double averaging scheme, where each agent iteratively performs averaging over both space and time to incorporate neighboring gradient information and local reward information, respectively. We prove that the proposed algorithm converges to the optimal solution at a global geometric rate. In particular, such an algorithm is built upon a primal-dual reformulation of the mean squared projected Bellman error minimization problem, which gives rise to a decentralized convex-concave saddle-point problem. To the best of our knowledge, the proposed double averaging primal-dual optimization algorithm is the first to achieve fast finite-time convergence on decentralized convex-concave saddle-point problems.

1 Introduction

Reinforcement learning combined with deep neural networks recently achieves superhuman performance on various challenging tasks such as video games and board games (Mnih et al., 2015; Silver et al., 2017). In these tasks, an agent uses deep neural networks to learn from the environment and adaptively makes optimal decisions. Despite the success of single-agent reinforcement learning, multi-agent reinforcement learning (MARL) remains challenging, since each agent interacts with not only the environment but also other agents.

In this paper, we study collaborative MARL with local rewards. In this setting, all the agents share a joint state whose transition dynamics is determined together by the local actions of individual agents. However, each agent only observes its own reward, which may differ from that of other agents. The agents aim to collectively maximize the global sum of local rewards. To collaboratively make globally optimal decisions, the agents need to exchange local information. Such a setting of MARL is ubiquitous in large-scale applications such as sensor networks (Rabbat and Nowak, 2004; Cortes et al., 2004), swarm robotics (Kober and Peters, 2012; Corke et al., 2005), and power grids (Callaway and Hiskens, 2011; Dall'Anese et al., 2013).

A straightforward idea is to set up a central node that collects and broadcasts the reward information, and assigns the action of each agent. This reduces the multi-agent problem into a single-agent one. However, the central node is often unscalable, susceptible to malicious attacks, and even infeasible in large-scale applications. Moreover, such a central node is a single point of failure, which is susceptible to adversarial attacks. In addition, the agents are likely to be reluctant to reveal their local reward information due to privacy concerns (Chaudhuri et al., 2011; Lin and Ling, 2014), which makes the central node unattainable.

*H.-T. Wai is with Arizona State University, AZ, USA. Z. Yang is with Princeton University, NJ, USA. Z. Wang is with Northwestern University, IL, USA. M. Hong is with University of Minnesota, MN, USA. E-mails: `htwai@asu.edu`, `zy6@princeton.edu`, `zhaoranwang@gmail.com`, `mhong@umn.edu`

To make MARL more scalable and robust, we propose a decentralized scheme for exchanging local information, where each agent only communicates with its neighbors over a network. In particular, we study the policy evaluation problem, which aims to learn a global value function of a given policy. We focus on minimizing a Fenchel duality-based reformulation of the mean squared Bellman error in the model-free setting with infinite horizon, batch trajectory, and linear function approximation. To this end, we propose a decentralized primal-dual optimization algorithm with a global geometric rate of convergence.

At the core of the proposed algorithm is a "double averaging" update scheme, in which the algorithm performs one average over space (across agents to ensure consensus) and one over time (across observations along the trajectory). In detail, each agent locally tracks an estimate of the full gradient and incrementally updates it using two sources of information: (i) the stochastic gradient evaluated on a new pair of joint state and action along the trajectory and the corresponding local reward, and (ii) the local estimates of the full gradient tracked by its neighbors. Based on the updated estimate of the full gradient, each agent then updates its local copy of the primal parameter. By iteratively propagating the local information through the network, the agents reach global consensus and collectively attain the desired primal parameter, which gives an optimal approximation of the global value function.

Related Work The study of MARL in the context of Markov game dates back to Littman (1994). See also Littman (2001); Lauer and Riedmiller (2000); Hu and Wellman (2003) and recent works on collaborative MARL Wang and Sandholm (2003); Arslan and Yüksel (2017). However, most of these works consider the tabular setting, which suffers from the curse of dimensionality. To address this issue, under the collaborative MARL framework, Zhang et al. (2018) and Lee et al. (2018) study actor-critic algorithms and policy evaluation with on linear function approximation, respectively. However, their analysis is asymptotic in nature and largely relies on two-time-scale stochastic approximation using ordinary differential equations (Borkar, 2008), which is tailored towards the continuous-time setting. Meanwhile, most works on collaborative MARL impose the simplifying assumption that the local rewards are identical across agents, making it unnecessary to exchange the local information. More recently, Foerster et al. (2016, 2017); Gupta et al. (2017); Lowe et al. (2017); Omidshafiei et al. (2017) study deep MARL that uses deep neural networks as function approximators. However, most of these works focus on empirical performance and lack theoretical guarantees. Also, they do not emphasize on the efficient exchange of information across agents. In addition to MARL, another line of related works study multi-task reinforcement learning (MTRL), in which an agent aims to solve multiple reinforcement learning problems with shared structures (Wilson et al., 2007; Parisotto et al., 2015; Macua et al., 2015, 2017; Teh et al., 2017).

The primal-dual formulation of reinforcement learning is studied in Liu et al. (2015); Macua et al. (2015, 2017); Lian et al. (2016); Dai et al. (2016); Chen and Wang (2016); Wang (2017); Dai et al. (2017b,a); Du et al. (2017) among others. Except for Macua et al. (2015, 2017) discussed above, most of these works study the single-agent setting. Among them, Lian et al. (2016); Du et al. (2017) are most related to our work. In specific, they develop variance reduction-based algorithms (Johnson and Zhang, 2013; Defazio et al., 2014; Schmidt et al., 2017) to achieve the geometric rate of convergence in the setting with batch trajectory. In comparison, our algorithm is based on the aforementioned double averaging update scheme, which updates the local estimates of the full gradient using both the estimates of neighbors and new states, actions, and rewards. In the single-agent setting, our algorithm is closely related to stochastic average gradient (SAG) (Schmidt et al., 2017) and stochastic incremental gradient (SAGA) (Defazio et al., 2014), with the difference that our objective function is a finite sum convex-concave saddle-point problem.

Our work is also related to prior work in the broader contexts of primal-dual and multi-agent optimization. For example, Palaniappan and Bach (2016) apply variance reduction techniques to convex-concave saddle-point problems to achieve the geometric rate of convergence. However, their algorithm is centralized and it is unclear whether their approach is readily applicable to the multi-agent setting. Another line of related works study multi-agent optimization, for example, Tsitsiklis et al. (1986); Nedic and Ozdaglar (2009); Chen and Sayed (2012); Shi et al. (2015); Qu and Li (2017). However, these works mainly focus on the general setting where the objective function is a sum of convex local cost functions. To the best of our knowledge, our work is the first to address decentralized convex-concave saddle-point problems with sampled observations that arise from MARL.

Contribution In summary, our contribution is threefold: (i) We reformulate the multi-agent policy evaluation problem using Fenchel duality and propose a decentralized primal-dual optimization algorithm with a double averaging update scheme. (ii) We establish the global geometric rate of convergence for the proposed algorithm, making it the first known algorithm to achieve fast finite-time convergence for MARL. (iii) Our proposed algorithm and analysis is of independent interest for solving a broader class of decentralized convex-concave saddle-point problems with sampled observations.

Organization In §2 we introduce the problem formulation of MARL. In §3 we present the proposed algorithm and lay out the convergence analysis. In §4 we illustrate the empirical performance of the proposed algorithm and we conclude the paper with discussions. We defer the detailed proofs to the supplementary material.

2 Problem Formulation

In this section, we introduce the background of MARL, which is modeled as a multi-agent Markov decision process (MDP). Under this model, we formulate the policy evaluation problem as a primal-dual convex-concave optimization problem.

Multi-agent MDP Consider a group of N agents. We are interested in the following multi-agent MDP:

$$\left(\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \mathcal{P}^{\boldsymbol{a}}, \{\mathcal{R}_i\}_{i=1}^N, \gamma\right) ,$$

where $\mathcal{S}$ is the state space and $\mathcal{A}_i$ is the action space for agent i. We write $\boldsymbol{s} \in \mathcal{S}$ and $\boldsymbol{a} := (a_1, ..., a_N) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ as the joint state and action, respectively. The function $\mathcal{R}_i(\boldsymbol{s}, \boldsymbol{a})$ is the local reward received by agent i after taking joint action $\boldsymbol{a}$ at state $\boldsymbol{s}$, and $\gamma \in (0, 1)$ is the discount factor. We assume that both the state and the joint action are available to all the agents, whereas the reward $\mathcal{R}_i$ is *private* for agent i.

In contrast to a single-agent MDP, the agents are coupled together by the state transition matrix $\mathcal{P}^{\boldsymbol{a}} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, whose $(\boldsymbol{s}, \boldsymbol{s}')$-th element is the probability of transiting from state $\boldsymbol{s}$ to state $\boldsymbol{s}$, after taking a joint action $\boldsymbol{a}$. This scenario arises from large-scale applications such as sensor networks (Rabbat and Nowak, 2004; Cortes et al., 2004), swarm robotics (Kober and Peters, 2012; Corke et al., 2005), and power grids (Callaway and Hiskens, 2011; Dall'Anese et al., 2013), which strongly motivates the development of a multi-agent RL strategy.

Moreover, under the collaborative setting, the goal is to maximize the collective return of all agents. Suppose there exists a central controller that collects the rewards of and assigns the action to each individual agent, the problem reduces to the classical MDP with action space $\mathcal{A}$ and global reward function $R_c(\boldsymbol{s}, \boldsymbol{a}) = N^{-1} \sum_{i=1}^N \mathcal{R}_i(\boldsymbol{s}, \boldsymbol{a})$. Thus, without such a central controller, it is essential for the agents to collaborate with each other so as to solve the multi-agent problem based solely on local information.

Furthermore, a joint policy, denoted by $\boldsymbol{\pi}$, specifies the rule of making sequential decisions for the agents. Specifically, $\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s})$ is the conditional probability of taking joint action $\boldsymbol{a}$ given the current state $\boldsymbol{s}$. We define the reward function of joint policy $\boldsymbol{\pi}$ as an average of the local rewards:

$$R_c^{\boldsymbol{\pi}}(\boldsymbol{s}) := \frac{1}{N} \sum_{i=1}^N R_i^{\boldsymbol{\pi}}(\boldsymbol{s}), \qquad \text{where } \; R_i^{\boldsymbol{\pi}}(\boldsymbol{s}) := \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|\boldsymbol{s})}\big[\mathcal{R}_i(\boldsymbol{s}, \boldsymbol{a})\big] \; . \tag{1}$$

That is, $R_c^{\boldsymbol{\pi}}(\boldsymbol{s})$ is the expected value of the average of the rewards when the agents follow policy $\boldsymbol{\pi}$ at state $\boldsymbol{s}$. Besides, any fixed policy $\boldsymbol{\pi}$ induces a Markov chain over $\mathcal{S}$, whose transition matrix is denoted by $\boldsymbol{P}^{\boldsymbol{\pi}}$. The $(\boldsymbol{s}, \boldsymbol{s}')$-th element of $\boldsymbol{P}^{\boldsymbol{\pi}}$ is given by

$$[\boldsymbol{P}^{\boldsymbol{\pi}}]_{\boldsymbol{s},\boldsymbol{s}'} = \sum_{\boldsymbol{a}\in\mathcal{A}} \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s}) \cdot [\mathcal{P}^{\boldsymbol{a}}]_{\boldsymbol{s},\boldsymbol{s}'} .$$

When this Markov chain is aperiodic and irreducible, it induces a stationary distribution $\mu^{\boldsymbol{\pi}}$ over $\mathcal{S}$.

Policy Evaluation A pivotal problem in reinforcement learning is *policy evaluation*, which refers to learning the *value function* of a given policy. This problem appears as a key component in both value-based methods such as policy iteration, and policy-based methods such as actor-critic algorithms (Sutton and Barto, 1998). Thus, efficient estimation of the value functions in multi-agent MDPs enables us to extend the successful approaches in single-agent RL to the setting of MARL.

Specifically, for any given joint policy $\boldsymbol{\pi}$, the value function of $\boldsymbol{\pi}$, denoted by $V^{\boldsymbol{\pi}}\colon \mathcal{S} \to \mathbb{R}$, is defined as the expected value of the discounted cumulative reward when the multi-agent MDP is initialized with a given state and the agents follows policy $\boldsymbol{\pi}$ afterwards. In particular, for any state $\boldsymbol{s} \in \mathcal{S}$, we define

$$V^{\boldsymbol{\pi}}(\boldsymbol{s}) := \mathbb{E}\Big[\textstyle\sum_{p=1}^{\infty} \gamma^t \mathcal{R}_c^{\boldsymbol{\pi}}(\boldsymbol{s}_p) \,|\, \boldsymbol{s}_1 = \boldsymbol{s}, \boldsymbol{\pi}\Big] \,. \tag{2}$$

To simplify the notation, we define the vector $\boldsymbol{V}^{\boldsymbol{\pi}} \in \mathbb{R}^{|\mathcal{S}|}$ through stacking up $V^{\boldsymbol{\pi}}(\boldsymbol{s})$ in (2) for all $\boldsymbol{s}$. By definition, $V^{\boldsymbol{\pi}}$ satisfies the Bellman equation

$$\boldsymbol{V}^{\boldsymbol{\pi}} = \boldsymbol{R}_c^{\boldsymbol{\pi}} + \gamma \boldsymbol{P}^{\boldsymbol{\pi}} \boldsymbol{V}^{\boldsymbol{\pi}} \,, \tag{3}$$

where $\boldsymbol{R}_c^{\boldsymbol{\pi}}$ is obtained by stacking up (1) and $[\boldsymbol{P}^{\boldsymbol{\pi}}]_{\boldsymbol{s},\boldsymbol{s}'} := \mathbb{E}_{\boldsymbol{\pi}}[\mathcal{P}^{\boldsymbol{a}}_{\boldsymbol{s},\boldsymbol{s}'}]$ is the expected transition matrix. Moreover, it can be shown that $\boldsymbol{V}^{\boldsymbol{\pi}}$ is the unique solution of (3).

When the number of states is large, it is impossible to store $\boldsymbol{V}^{\boldsymbol{\pi}}$. Instead, our goal is to learn an approximate version of the value function via function approximation. In specific, we approximate $V^{\boldsymbol{\pi}}(\boldsymbol{s})$ using the family of linear functions

$$\big\{V_{\boldsymbol{\theta}}(\boldsymbol{s}) := \boldsymbol{\phi}^\top(\boldsymbol{s})\boldsymbol{\theta} \colon \boldsymbol{\theta} \in \mathbb{R}^d\big\},$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ is the parameter, $\boldsymbol{\phi}(\boldsymbol{s})\colon \mathcal{S} \to \mathbb{R}^d$ is a known dictionary consisting of d features, e.g., a feature mapping induced by a neural network. To simplify the notation, we define matrix $\boldsymbol{\Phi} := (...; \boldsymbol{\phi}^\top(\boldsymbol{s}); ...) \in \mathbb{R}^{|\mathcal{S}| \times d}$ and similarly let $\boldsymbol{V}_{\boldsymbol{\theta}} \in \mathbb{R}^{|\mathcal{S}|}$ be the vector constructed by stacking up $\{V_{\boldsymbol{\theta}}(\boldsymbol{s})\}_{\boldsymbol{s}\in\mathcal{S}}$.

With function approximation, our problem becomes finding $\boldsymbol{\theta} \in \mathbb{R}^d$ such that $\boldsymbol{V}_{\boldsymbol{\theta}} \approx \boldsymbol{V}^{\boldsymbol{\pi}}$. Specifically, we would like to find $\boldsymbol{\theta}$ such that the mean squared projected Bellman error (MSPBE)

$$\mathsf{MSPBE}^\star(\boldsymbol{\theta}) := \frac{1}{2}\Big\|\boldsymbol{\Pi}_{\boldsymbol{\Phi}}\Big(\boldsymbol{V}_{\boldsymbol{\theta}} - \gamma \boldsymbol{P}^{\boldsymbol{\pi}} \boldsymbol{V}_{\boldsymbol{\theta}} - \boldsymbol{R}_c^{\boldsymbol{\pi}}\Big)\Big\|_{\boldsymbol{D}}^2 + \rho\|\boldsymbol{\theta}\|^2 \tag{4}$$

is minimized, where $\boldsymbol{D} = \mathrm{diag}[\{\mu^{\boldsymbol{\pi}}(\boldsymbol{s})\}_{\boldsymbol{s}\in\mathcal{S}}] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is a diagonal matrix constructed using the stationary distribution of $\boldsymbol{\pi}$, $\boldsymbol{\Pi}_{\boldsymbol{\Phi}}\colon \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ is the projection onto subspace $\{\boldsymbol{\Phi\theta}\colon \boldsymbol{\theta} \in \mathbb{R}^d\}$, and $\rho \geq 0$ is a free parameter controlling the regularization on $\boldsymbol{\theta}$. Here $\|\cdot\|_{\boldsymbol{D}}$ in (4) is the weighted norm induced by $\boldsymbol{D}$. Specifically, for any positive semidefinite matrix $\boldsymbol{A}$, we define $\|\boldsymbol{v}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{v}^\top \boldsymbol{A} \boldsymbol{v}}$ for any vector $\boldsymbol{v}$. By direct computation, when $\boldsymbol{\Phi}^\top \boldsymbol{D} \boldsymbol{\Phi} \in \mathbb{R}^{d\times d}$ is invertible, the MSPBE defined in (4) can be written in closed-form as

$$\mathsf{MSPBE}^\star(\boldsymbol{\theta}) = \frac{1}{2}\Big\|\boldsymbol{\Phi}^\top \boldsymbol{D}\Big(\boldsymbol{V}_{\boldsymbol{\theta}} - \gamma \boldsymbol{P}^{\boldsymbol{\pi}} \boldsymbol{V}_{\boldsymbol{\theta}} - \boldsymbol{R}_c^{\boldsymbol{\pi}}\Big)\Big\|_{(\boldsymbol{\Phi}^\top \boldsymbol{D}\boldsymbol{\Phi})^{-1}}^2 + \rho\|\boldsymbol{\theta}\|^2 = \frac{1}{2}\Big\|\boldsymbol{A\theta} - \boldsymbol{b}\Big\|_{\boldsymbol{C}^{-1}}^2 + \rho\|\boldsymbol{\theta}\|^2, \tag{5}$$

where we define $\boldsymbol{A} := \mathbb{E}\big[\boldsymbol{\phi}(\boldsymbol{s}_p)\big(\boldsymbol{\phi}(\boldsymbol{s}_p) - \gamma\boldsymbol{\phi}(\boldsymbol{s}_{p+1})\big)^\top\big]$, $\boldsymbol{C} := \mathbb{E}\big[\boldsymbol{\phi}(\boldsymbol{s}_p)\boldsymbol{\phi}^\top(\boldsymbol{s}_p)\big]$, and $\boldsymbol{b} := \mathbb{E}\big[\mathcal{R}_c^{\boldsymbol{\pi}}(\boldsymbol{s}_p)\boldsymbol{\phi}(\boldsymbol{s}_p)\big]$. Here the expectations in $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{C}$ are all taken with respect to (*w.r.t.*) the stationary distribution $\mu^{\boldsymbol{\pi}}$. Furthermore, when $\boldsymbol{A}$ is full rank and $\boldsymbol{C}$ is positive definite, it can be shown that the MSPBE in (5) has a unique minimizer.

To obtain a practical optimization problem, we replace the expectations above by their sampled averages. In specific, for a given policy $\boldsymbol{\pi}$, we simulate a finite state-action sequence $\{\boldsymbol{s}_p, \boldsymbol{a}_p\}_{p=1}^M$ from the multi-agent MDP using joint policy $\boldsymbol{\pi}$, where $M > 0$ is the sample size. We also observe $\boldsymbol{s}_{M+1}$, the next state of $\boldsymbol{s}_M$. Then we construct the sampled versions of $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{C}$, denoted respectively by $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{C}}$, and $\hat{\boldsymbol{b}}$, by letting

$$\begin{aligned} &\hat{\boldsymbol{A}} := \frac{1}{M}\sum_{p=1}^M \boldsymbol{A}_p,\ \hat{\boldsymbol{C}} := \frac{1}{M}\sum_{p=1}^M \boldsymbol{C}_p,\ \hat{\boldsymbol{b}} := \frac{1}{M}\sum_{p=1}^M \boldsymbol{b}_p, \quad \text{with} \\ &\boldsymbol{A}_p := \boldsymbol{\phi}(\boldsymbol{s}_p)\big(\boldsymbol{\phi}(\boldsymbol{s}_p) - \gamma\boldsymbol{\phi}(\boldsymbol{s}_{p+1})\big)^\top,\ \boldsymbol{C}_p := \boldsymbol{\phi}(\boldsymbol{s}_p)\boldsymbol{\phi}^\top(\boldsymbol{s}_p),\ \boldsymbol{b}_p := \mathcal{R}_c(\boldsymbol{s}_p, \boldsymbol{a}_p)\boldsymbol{\phi}(\boldsymbol{s}_p)\ , \end{aligned} \tag{6}$$

where $\mathcal{R}_c(\boldsymbol{s}_p, \boldsymbol{a}_p) := N^{-1}\sum_{i=1}^{N}\mathcal{R}_i(\boldsymbol{s}_p, \boldsymbol{a}_p)$ is the average of the local rewards received by each agent when taking action $\boldsymbol{a}_p$ at state $\boldsymbol{s}_p$. Here we assume that M is sufficiently large such that $\hat{\boldsymbol{C}}$ is invertible and $\hat{\boldsymbol{A}}$ is full rank. Using the terms defined in (6), we obtain the empirical MSPBE

$$\mathsf{MSPBE}(\boldsymbol{\theta}) := \frac{1}{2}\left\|\hat{\boldsymbol{A}}\boldsymbol{\theta} - \hat{\boldsymbol{b}}\right\|^2_{\hat{\boldsymbol{C}}^{-1}} + \rho\|\boldsymbol{\theta}\|^2 \,, \tag{7}$$

which converges to $\mathsf{MSPBE}^{\star}(\boldsymbol{\theta})$ as M goes to infinity. Let $\hat{\boldsymbol{\theta}}$ be the minimizer the empirical MSPBE. Then our estimator of $\boldsymbol{V}^{\boldsymbol{\pi}}$ is given by $\boldsymbol{\Phi}\hat{\boldsymbol{\theta}}$. However, since the rewards $\{\mathcal{R}_i(\boldsymbol{s}_p, \boldsymbol{a}_p)\}_{i=1}^{N}$ are private to each agent, it is impossible for any agent to compute $\mathcal{R}_c(\boldsymbol{s}_p, \boldsymbol{a}_p)$. Thus, without a central node that collects and broadcasts the reward information, it is infeasible to directly minimize the empirical MSPBE in (7).

Multi-agent, Primal-dual, Finite-sum Optimization Recall that under the multi-agent MDP, the agents are able to observe the states and the joint actions, but can only observe their local rewards. Thus, each agent is able to compute $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{C}}$ defined in (6), but is unable to obtain $\hat{\boldsymbol{b}}$. To resolve this issue, for any $i \in \{1, \ldots, N\}$ and any $p \in \{1, \ldots, M\}$, we define $\boldsymbol{b}_{p,i} := \mathcal{R}_i(\boldsymbol{s}_p, \boldsymbol{a}_p)\boldsymbol{\phi}(\boldsymbol{s}_p)$ and $\hat{\boldsymbol{b}}_i := M^{-1}\sum_{p=1}^{M}\boldsymbol{b}_{p,i}$, which are known to agent i only. By direct computation, it is easy to verify that minimizing $\mathsf{MSPBE}(\boldsymbol{\theta})$ in (7) is equivalent to solving

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N}\mathsf{MSPBE}_i(\boldsymbol{\theta}), \qquad \text{where } \; \mathsf{MSPBE}_i(\boldsymbol{\theta}) := \frac{1}{2}\left\|\hat{\boldsymbol{A}}\boldsymbol{\theta} - \hat{\boldsymbol{b}}_i\right\|^2_{\hat{\boldsymbol{C}}^{-1}} + \rho\|\boldsymbol{\theta}\|^2 \,, \tag{8}$$

where the equivalence can be seen by comparing the optimality conditions of these two optimization problems.

Importantly, (8) falls into the class of *multi-agent optimization problems* (Nedic and Ozdaglar, 2009) whose objective is to minimize a summation of N local functions coupled together by a common parameter. Here $\mathsf{MSPBE}_i(\boldsymbol{\theta})$ is private to agent i and the same parameter $\boldsymbol{\theta}$ is shared by all agents. As inspired by (Nedić and Bertsekas, 2003; Liu et al., 2015; Du et al., 2017), using Fenchel duality, we obtain the conjugate form of $\mathsf{MSPBE}_i(\boldsymbol{\theta})$, *i.e.*,

$$\frac{1}{2}\left\|\hat{\boldsymbol{A}}\boldsymbol{\theta} - \hat{\boldsymbol{b}}_i\right\|^2_{\hat{\boldsymbol{C}}^{-1}} + \rho\|\boldsymbol{\theta}\|^2 = \max_{\boldsymbol{w}_i\in\mathbb{R}^d}\left(\boldsymbol{w}_i^{\top}\left(\hat{\boldsymbol{A}}\boldsymbol{\theta} - \hat{\boldsymbol{b}}_i\right) - \frac{1}{2}\boldsymbol{w}_i^{\top}\hat{\boldsymbol{C}}\boldsymbol{w}_i\right) + \rho\|\boldsymbol{\theta}\|^2 \,. \tag{9}$$

Observe that each of $\hat{\boldsymbol{A}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{b}}_i$ can be expressed as a finite sum of matrices/vectors. By (9), the optimization problem in (8) is equivalent to a *multi-agent*, *primal-dual* and *finite-sum* optimization problem:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{w}_i\in\mathbb{R}^d, i=1,\ldots,N} \frac{1}{NM}\sum_{i=1}^{N}\sum_{p=1}^{M}\underbrace{\left(\boldsymbol{w}_i^{\top}\boldsymbol{A}_p\boldsymbol{\theta} - \boldsymbol{b}_{p,i}^{\top}\boldsymbol{w}_i - \frac{1}{2}\boldsymbol{w}_i^{\top}\boldsymbol{C}_p\boldsymbol{w}_i + \frac{\rho}{2}\|\boldsymbol{\theta}\|^2\right)}_{:=J_{i,p}(\boldsymbol{\theta},\boldsymbol{w}_i)} \,. \tag{10}$$

Hereafter, the global objective function is denoted by $J(\boldsymbol{\theta}, \{\boldsymbol{w}_i\}_{i=1}^{N}) := (1/NM)\sum_{i=1}^{N}\sum_{p=1}^{M}J_{i,p}(\boldsymbol{\theta}, \boldsymbol{w}_i)$, which is convex *w.r.t.* the primal variable $\boldsymbol{\theta}$ and is concave *w.r.t.* the dual variable $\{\boldsymbol{w}_i\}_{i=1}^{N}$.

It is worth noting that the challenges in solving (10) are three-fold. First, to obtain a saddle-point solution $(\{\boldsymbol{w}_i\}_{i=1}^{N}, \boldsymbol{\theta})$, any algorithm for (10) needs to update the primal and dual variables simultaneously, which can be difficult as objective function needs not be strongly convex with respect to $\boldsymbol{\theta}$ as we allow ρ to be zero. In this case, it is nontrivial to find a solution with computational efficiency. Second, the objective function of (10) consists of a sum of M functions, with $M \gg 1$ potentially, such that conventional primal-dual methods (Chambolle and Pock, 2016) can no longer be applied due to the increased complexity. Lastly, since $\boldsymbol{\theta}$ is shared by all the agents, when solving (10), the N agents need to reach a consensus on $\boldsymbol{\theta}$ without sharing the local functions, e.g., $J_{i,p}(\cdot)$ has to remain unknown to all agents except for agent i due to privacy concerns. Although finite-sum convex optimization problems with shared variables are well-studied, new algorithms and theory are needed for convex-concave saddle-point problems. In the next section, we propose a novel decentralized first-order algorithm that tackles these difficulties and is proven to converge to a saddle-point solution of (10) with linear rate.

3 Primal-dual Distributed Incremental Aggregated Gradient Method

We are ready to introduce our algorithm for solving the optimization problem in (10). Since $\boldsymbol{\theta}$ is shared by all the N agents, the agents need to exchange information so as to reach a consensual solution. Let us first specify the communication model. We assume that the N agents communicate over a network specified by a connected and undirected graph $G = (V, E)$, with $V = [N] = \{1, ..., N\}$ and $E \subseteq V \times V$ being its vertex set and edge set, respectively. Over this graph, it is possible to define a doubly stochastic matrix $\boldsymbol{W}$ such that $W_{ij} = 0$ if $(i,j) \notin E$ and $\boldsymbol{W}\mathbf{1} = \boldsymbol{W}^\top \mathbf{1} = \mathbf{1}$. Moreover, we have $\lambda := \|\boldsymbol{W} - N^{-1}\mathbf{1}\mathbf{1}^\top\|_{1,\infty} < 1$ since G is connected. Moreover, it is important to note that the edges in G may be formed independently of the coupling between agents in the MDP induced by the stochastic policy $\boldsymbol{\pi}$.

We handle problem (10) by judiciously combining the techniques of *dynamic consensus* (Qu and Li, 2017; Zhu and Martínez, 2010) and *stochastic (or incremental) average gradient* (SAG) (Gurbuzbalaban et al., 2017; Schmidt et al., 2017), which have been developed independently in the control and machine learning communities, respectively. From a high level viewpoint, our method utilizes a gradient estimator which tracks the gradient over *space* (across N agents) and *time* (across M samples). To proceed with our development while explaining the intuitions, we first investigate a centralized and batch algorithm for solving (10).

Centralized Primal-dual Optimization Consider the primal-dual gradient updates. Specifically, for any $t \geq 1$, at the t-th iteration, we update the primal and dual variables by

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \gamma_1 \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \{\boldsymbol{w}_i^t\}_{i=1}^N), \qquad \boldsymbol{w}_i^{t+1} = \boldsymbol{w}_i^t + \gamma_2 \nabla_{\boldsymbol{w}_i} J(\boldsymbol{\theta}^t, \{\boldsymbol{w}_i^t\}_{i=1}^N),\ i \in [N]\,, \tag{11}$$

where $\gamma_1, \gamma_2 > 0$ are step sizes, which is a simple application of a gradient descent/ascent update to the primal/dual variables. As shown by Du et al. (2017), when $\hat{\boldsymbol{A}}$ is full rank and $\hat{\boldsymbol{C}}$ is invertible, the Jacobian matrix of the primal-dual optimal condition is a full rank matrix as long as $\rho \geq 0$. Thus, within a certain range of step size (γ_1, γ_2), the primal dual algorithm converges linearly to the optimal solution of (10).

Proposed Method The primal-dual gradient method in (11) serves as a reasonable template for developing an efficient decentralized algorithm for (10). Let us focus on the update of the primal variable $\boldsymbol{\theta}$ in (11), which is a more challenging part since $\boldsymbol{\theta}$ is shared by all the N agents. To evaluate the gradient *w.r.t.* $\boldsymbol{\theta}$, we observe that – (a) agent i does not have access to the functions, $\{J_{j,p}(\cdot), j \neq i\}$, of the other agents; (b) computing the gradient requires summing up the contributions from M samples. As $M \gg 1$, doing so is highly undesirable since the computation complexity would be $\mathcal{O}(Md)$.

We circumvent the above issues by utilizing a *double gradient tracking* scheme for the primal $\boldsymbol{\theta}$-update and an incremental update scheme for the local dual $\boldsymbol{w}_i$-update in the following primal-dual distributed incremental aggregated gradient (PD-DistIAG) method. Here each agent $i \in [N]$ maintains a local copy of the primal parameter $\{\boldsymbol{\theta}_i^t\}_{t\geq 1}$. Moreover, we construct sequences $\{\boldsymbol{s}_i^t\}_{t\geq 1}$ and $\{\boldsymbol{d}_i^t\}_{t\geq 1}$ to track the gradients with respect to $\boldsymbol{\theta}$ and $\boldsymbol{w}_i$, respectively. Similar to (11), in the t-th iteration, we update the dual variable via gradient update using $\boldsymbol{d}_i^t$. As for the primal variable, to achieve consensus, each $\boldsymbol{\theta}_i^{t+1}$ is obtained by first combining $\{\boldsymbol{\theta}_i^t\}_{i\in[N]}$ using the weight matrix $\boldsymbol{W}$, and then update in the direction of $\boldsymbol{s}_i^t$. The details of our PD-DistIAG method are presented in Algorithm 1.

Let us explain the intuition behind the PD-DistIAG method through studying the update in (13). Recall that the global gradient desired at iteration t is given by $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^t, \{\boldsymbol{w}_i^t\}_{i=1}^N)$, which represents a double average – one over space (across agents) and one over time (across samples). Now in the case of (13), the first summand on the right hand side computes a local average among the neighbors of agent i, and thereby tracking the global gradient over *space*. This is in fact akin to the technique known as *gradient tracking* in the context of distributed optimization (Qu and Li, 2017). The remaining terms on the right hand side of (13) utilize an incremental update rule akin to the SAG method (Schmidt et al., 2017), involving a swap-in swap-out operation for the gradients. This achieves tracking of the global gradient over *time*.

To gain insights on why the scheme works, we note that $\boldsymbol{s}_i^t$ and $\boldsymbol{d}_i^t$ represent some surrogate functions for the primal and dual gradients. Moreover, for the counter variable, using (12) we can alternatively represent

Algorithm 1 PD-DistIAG **Method** for Multi-agent, Primal-dual, Finite-sum Optimization

```
Input: Initial estimators $\{\boldsymbol{\theta}_i^1, \boldsymbol{w}_i^1\}_{i \in [N]}$, initial gradient estimators $\boldsymbol{s}_i^0 = \boldsymbol{d}_i^0 = \mathbf{0}$, $\forall\ i \in [N]$, initial counter $\tau_p^0 = 0$, $\forall\ p \in [M]$, and stepsizes $\gamma_1, \gamma_2 > 0$.
for $t \geq 1$ do
    The agents pick a common sample indexed by $p_t \in \{1, ..., M\}$.
    Update the counter variable as:
        $\tau_{p_t}^t = t, \quad \tau_p^t = \tau_p^{t-1}, \ \forall\ p \neq p_t \ . \qquad (12)$
    for each agent $i \in \{1, \ldots, N\}$ do
        Update the gradient surrogates by
            $\boldsymbol{s}_i^t = \sum_{j=1}^N W_{ij} \boldsymbol{s}_j^{t-1} + \frac{1}{M} \Big[ \nabla_{\boldsymbol{\theta}} J_{i,p_t}(\boldsymbol{\theta}_i^t, \boldsymbol{w}_i^t) - \nabla_{\boldsymbol{\theta}} J_{i,p_t}(\boldsymbol{\theta}_i^{\tau_{p_t}^{t-1}}, \boldsymbol{w}_i^{\tau_{p_t}^{t-1}}) \Big] \ , \qquad (13)$
            $\boldsymbol{d}_i^t = \boldsymbol{d}_i^{t-1} + \frac{1}{M} \Big[ \nabla_{\boldsymbol{w}_i} J_{i,p_t}(\boldsymbol{\theta}_i^t, \boldsymbol{w}_i^t) - \nabla_{\boldsymbol{w}_i} J_{i,p_t}(\boldsymbol{\theta}_i^{\tau_{p_t}^{t-1}}, \boldsymbol{w}_i^{\tau_{p_t}^{t-1}}) \Big] \ , \qquad (14)$
        where we define $\nabla_{\boldsymbol{\theta}} J_{i,p}(\boldsymbol{\theta}_i^0, \boldsymbol{w}_i^0) = \mathbf{0}$ and $\nabla_{\boldsymbol{w}_i} J_{i,p}(\boldsymbol{\theta}_i^0, \boldsymbol{w}_i^0) = \mathbf{0}$ for all $p \in [M]$ for the initialization.
        Perform primal-dual updates using $\boldsymbol{s}_i^t, \boldsymbol{d}_i^t$ as surrogates for the gradients w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{w}_i$:
            $\boldsymbol{\theta}_i^{t+1} = \sum_{j=1}^N W_{ij} \boldsymbol{\theta}_j^t - \gamma_1 \boldsymbol{s}_i^t, \qquad \boldsymbol{w}_i^{t+1} = \boldsymbol{w}_i^t + \gamma_2 \boldsymbol{d}_i^t \ . \qquad (15)$
    end for
end for
```

it as $\tau_p^t = \max\{\ell \geq 0 \,:\, \ell \leq t, \ p_\ell = p\}$. In other words, τ_p^t is the iteration index where the p-th sample is last visited by the agents prior to iteration t, and if the p-th sample has never been visited, we have $\tau_p^t = 0$.

For any $t \geq 1$, define $\boldsymbol{g}_{\boldsymbol{\theta}}(t) := (1/N) \sum_{i=1}^N \boldsymbol{s}_i^t$. The following lemma shows that $\boldsymbol{g}_{\boldsymbol{\theta}}(t)$ is a double average of the primal gradient – it averages over the local gradients across the agents, and for each local gradient; it also averages over the past gradients for all the samples evaluated up till iteration $t+1$. This shows that the average over network for $\{\boldsymbol{s}_i^t\}_{i=1}^N$ can always track the double average of the local and past gradients, *i.e.,* the gradient estimate $\boldsymbol{g}_{\boldsymbol{\theta}}(t)$ is 'unbiased' with respect to the network-wide average.

Lemma 1 *For all $t \geq 1$ and consider Algorithm 1, it holds that*

$$\boldsymbol{g}_{\boldsymbol{\theta}}(t) = \frac{1}{NM} \sum_{i=1}^N \sum_{p=1}^M \nabla_{\boldsymbol{\theta}} J_{i,p}(\boldsymbol{\theta}_i^{\tau_p^t}, \boldsymbol{w}_i^{\tau_p^t}) \ . \tag{16}$$

Proof. We shall prove the statement using induction. For the base case with $t = 1$, using (13) and the update rule specified in the algorithm, we have

$$\boldsymbol{g}_{\boldsymbol{\theta}}(1) = \frac{1}{N} \sum_{i=1}^N \frac{1}{M} \nabla_{\boldsymbol{\theta}} J_{i,p_1}(\boldsymbol{\theta}_i^1, \boldsymbol{w}_i^1) = \frac{1}{NM} \sum_{i=1}^N \sum_{p=1}^M \nabla_{\boldsymbol{\theta}} J_{i,p_t}(\boldsymbol{\theta}_i^{\tau_p^1}, \boldsymbol{w}_i^{\tau_p^1}) \ , \tag{17}$$

where we use the fact $\nabla_{\boldsymbol{\theta}} J_{i,p}(\boldsymbol{\theta}_i^{\tau_p^1}, \boldsymbol{w}_i^{\tau_p^1}) = \nabla_{\boldsymbol{\theta}} J_{i,p}(\boldsymbol{\theta}_i^0, \boldsymbol{w}_i^0) = \mathbf{0}$ for all $p \neq p_1$ in the above equality. For the induction step, suppose (16) holds up to iteration t. Since $\boldsymbol{W}$ is doubly stochastic, (13) implies that

$$\begin{aligned} \boldsymbol{g}_{\boldsymbol{\theta}}(t+1) &= \frac{1}{N} \sum_{i=1}^N \Big\{ \sum_{j=1}^N W_{ij} \boldsymbol{s}_j^t + \frac{1}{M} \Big[\nabla_{\boldsymbol{\theta}} J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{t+1}, \boldsymbol{w}_i^{t+1}) - \nabla_{\boldsymbol{\theta}} J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{\tau_{p_{t+1}}^t}, \boldsymbol{w}_i^{\tau_{p_{t+1}}^t}) \Big] \Big\} \\ &= \boldsymbol{g}_{\boldsymbol{\theta}}(t) + \frac{1}{NM} \sum_{i=1}^N \Big[\nabla_{\boldsymbol{\theta}} J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{t+1}, \boldsymbol{w}_i^{t+1}) - \nabla_{\boldsymbol{\theta}} J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{\tau_{p_{t+1}}^t}, \boldsymbol{w}_i^{\tau_{p_{t+1}}^t}) \Big] \ . \end{aligned} \tag{18}$$

Notice that we have $\tau_{p_{t+1}}^{t+1} = t+1$ and $\tau_p^{t+1} = \tau_p^t$ for all $p \neq p_{t+1}$. The induction assumption in (16) can be written as

$$\boldsymbol{g}_{\boldsymbol{\theta}}(t) = \frac{1}{NM} \sum_{i=1}^{N} \Bigg[\sum_{p \neq p_{t+1}} \nabla_{\boldsymbol{\theta}} J_{i,p}(\boldsymbol{\theta}_i^{\tau_p^{t+1}}, \boldsymbol{w}_i^{\tau_p^{t+1}}) \Bigg] + \frac{1}{NM} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{\tau_{p_{t+1}}^t}, \boldsymbol{w}_i^{\tau_{p_{t+1}}^t}) \,. \tag{19}$$

Finally, combining (18) and (19), we obtain that

$$\boldsymbol{g}_{\boldsymbol{\theta}}(t+1) = \frac{1}{NM} \sum_{i=1}^{N} \sum_{p=1}^{M} \nabla_{\boldsymbol{\theta}} J_{i,p}(\boldsymbol{\theta}_i^{\tau_p^{t+1}}, \boldsymbol{w}_i^{\tau_p^{t+1}}) \,,$$

which, together with (17), establishes Lemma 1. **Q.E.D.**

As for the dual update (14), we observe the variable $\boldsymbol{w}_i$ is local to agent i. Therefore its gradient surrogate, $\boldsymbol{d}_i^t$, involves only the tracking step over *time* [cf. (14)], *i.e.,* it only averages the gradient over samples. Combining with Lemma 1 shows that the PD-DistIAG method uses gradient surrogates that are averages over samples despite the disparities across agents. Since the average over samples are done in a similar spirit as the SAG method, the proposed method is expected to converge linearly.

Storage and Computation Complexities Let us comment on the computational and storage complexity of PD-DistIAG method. First of all, since the method requires accessing the previously evaluated gradients, each agent has to store $2M$ such vectors in the memory to avoid re-evaluating these gradients. Each agent needs to store a total of $2Md$ real numbers. On the other hand, the per-iteration computation complexity for each agent is only $\mathcal{O}(d)$ as each iteration only requires to evaluate the gradient over one sample, as delineated in (15)–(14).

Communication Overhead The PD-DistIAG method described in Algorithm 1 requires an information exchange round [of $\boldsymbol{s}_i^t$ and $\boldsymbol{\theta}_i^t$] among the agents at every iteration. From an implementation stand point, this may incur significant communication overhead when $d \gg 1$, and it is especially ineffective when the progress made in successive updates of the algorithm is not significant. A natural remedy is to perform multiple *local* updates at the agent using different samples *without* exchanging information with the neighbors. In this way, the communication overhead can be reduced. Actually, this modification to the PD-DistIAG method can be generally described using a time varying weight matrix $\boldsymbol{W}(t)$, such that we have $\boldsymbol{W}(t) = \boldsymbol{I}$ for most of the iteration. The convergence of PD-DistIAG method in this scenario is part of the future work.

3.1 Convergence Analysis

The PD-DistIAG method is built using the techniques of (a) primal-dual batch gradient descent, (b) gradient tracking for distributed optimization and (c) stochastic average gradient, where each of them has been independently shown to attain linear convergence under certain conditions; see (Qu and Li, 2017; Schmidt et al., 2017; Gurbuzbalaban et al., 2017; Du et al., 2017). Naturally, the PD-DistIAG method is also anticipated to converge at a linear rate.

To see this, let us consider the condition for the sample selection rule of PD-DistIAG:

Assumption 1 *A sample is selected at least once for every M iterations, $|t - \tau_p^t| \leq M$ for all $p \in [M]$ and all $t \geq 1$.*

The assumption requires that every samples are visited infinitely often. For example, this can be enforced by using a cyclical selection rule, *i.e.,* $p_t = (t \bmod M) + 1$; or a random sampling scheme *without replacement* (*i.e.,* random shuffling) from the pool of M samples. Finally, it is possible to relax the assumption such that a sample can be selected once for every K iterations only, with $K \geq M$. The present assumption is made solely for the purpose of ease of presentation.

To ensure that the solution to (10) is unique, we consider:

Assumption 2 *The sampled correlation matrix $\hat{\boldsymbol{A}}$ is full rank, and the sampled covariance $\hat{\boldsymbol{C}}$ is non-singular.*

Under mild conditions on the dictionary $\boldsymbol{\phi}(\cdot)$ and state transition matrix of the MDP, the assumption holds if the number of samples collected, M, is large. The following theorem confirms the linear convergence of PD-DistIAG when a sufficiently small step size is used.

Theorem 1 *Under Assumptions 1 and 2, we denote by $(\boldsymbol{\theta}^\star, \{\boldsymbol{w}_i^\star\}_{i=1}^N)$ the primal-dual optimal solution to the optimization problem in (10). Set the step sizes as $\gamma_2 = \beta\gamma_1$ with $\beta := 8(\rho + \lambda_{\sf max}(\hat{\boldsymbol{A}}^\top \hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{A}}))/\lambda_{\sf min}(\hat{\boldsymbol{C}})$. Define $\overline{\boldsymbol{\theta}}(t) := \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^t$ as the average of parameters at iteration t. If the primal step size γ_1 is sufficiently small, then there exists a constant $0 < \sigma < 1$ such that*

$$\left\|\overline{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star\right\|^2 + (1/\beta N)\textstyle\sum_{i=1}^N \left\|\boldsymbol{w}_i^t - \boldsymbol{w}_i^\star\right\|^2 = \mathcal{O}(\sigma^t)\;, \qquad (1/N)\textstyle\sum_{i=1}^N \left\|\boldsymbol{\theta}_i^t - \overline{\boldsymbol{\theta}}(t)\right\| = \mathcal{O}(\sigma^t)\;. \tag{20}$$

Moreover, if $N, M \gg 1$ and the graph is geometric such that $\lambda = 1 - c/N$ for some positive c, a sufficient condition for convergence is to set $\gamma = \mathcal{O}(1/\max\{N^2, M^2\})$ and the resultant rate is $\sigma = 1 - \mathcal{O}(1/\max\{MN^2, M^3\})$.

The result in (20) shows the desirable convergence properties for PD-DistIAG method – the primal dual solution $(\overline{\boldsymbol{\theta}}(t), \{\boldsymbol{w}_i^t\}_{i=1}^N)$ converges to $(\boldsymbol{\theta}^\star, \{\boldsymbol{w}_i^\star\}_{i=1}^N)$ at a linear rate; also, the consensual error of the local parameters $\overline{\boldsymbol{\theta}}_i^t$ converges to zero linearly. A distinguishing feature of our analysis is that it handles the *worst case* convergence of the proposed method, rather than the *expected* convergence rate popular for stochastic / incremental gradient methods.

Proof Sketch Our proof is divided into three steps. The first step studies the progress made by the algorithm in one iteration, taking into account the non-idealities due to imperfect tracking of the gradient over space and time. This leads to the characterization of a *Lyapunov vector*. The second step analyzes the *coupled* system of one iteration progress made by the Lyapunov vector. An interesting feature of it is that it consists of a series of independently *delayed* terms in the Lyapunov vector. The latter is resulted from the incremental update schemes employed in the method. Here, we study a sufficient condition for the coupled and delayed system to converge linearly. The last step is to derive condition on the step size γ_1 where the sufficient convergence condition is satisfied.

Specifically, in the supplementary material, we study the progress of the following Lyapunov functions:

$$\|\underline{\widehat{\boldsymbol{v}}}(t)\|^2 = \Theta\Big(\left\|\overline{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star\right\|^2 + (1/\beta N)\textstyle\sum_{i=1}^N \left\|\boldsymbol{w}_i^t - \boldsymbol{w}_i^\star\right\|^2\Big)\;, \qquad \mathcal{E}_c(t) := \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{\theta}_i^t - \overline{\boldsymbol{\theta}}(t)\|,$$

$$\mathcal{E}_g(t) := \frac{1}{N}\sum_{i=1}^N \Big\|\boldsymbol{s}_i^t - \frac{1}{NM}\sum_{j=1}^N\sum_{p=1}^M \nabla_{\boldsymbol{\theta}} J_{j,p}(\boldsymbol{\theta}_j^{\tau_p^t}, \boldsymbol{w}_j^{\tau_p^t})\Big\|\;.$$

That is, $\underline{\widehat{\boldsymbol{v}}}(t)$ is a vector whose squared norm is equivalent to a weighted distance to the optimal primal-dual solution, $\mathcal{E}_c(t)$ and $\mathcal{E}_g(t)$ are respectively the consensus errors of the primal parameter and of the primal *aggregated* gradient. We show that these Lyapunov functions forms a non-negative vector that evolves according to the following system:

$$\begin{pmatrix} \|\underline{\widehat{\boldsymbol{v}}}(t+1)\| \\ \mathcal{E}_c(t+1) \\ \mathcal{E}_g(t+1) \end{pmatrix} \leq \boldsymbol{Q}(\gamma_1) \begin{pmatrix} \max_{(t-2M)_+ \leq q \leq t} \|\underline{\widehat{\boldsymbol{v}}}(q)\| \\ \max_{(t-2M)_+ \leq q \leq t} \mathcal{E}_c(q) \\ \max_{(t-2M)_+ \leq q \leq t} \mathcal{E}_g(q) \end{pmatrix}\;, \tag{21}$$

where the matrix $\boldsymbol{Q}(\gamma_1) \in \mathbb{R}^{3\times 3}$ is defined by

$$\boldsymbol{Q}(\gamma_1) = \begin{pmatrix} 1 - \gamma_1 a_0 + \gamma_1^2 a_1 & \gamma_1 a_2 & 0 \\ 0 & \lambda & \gamma_1 \\ \gamma_1 a_3 & a_4 + \gamma_1 a_5 & \lambda + \gamma_1 a_6 \end{pmatrix}\;. \tag{22}$$

In the above, $\lambda := \|\boldsymbol{W}\|_{1,\infty} < 1$, and $a_0, ..., a_6$ are some non-negative constants that depends on the problem parameters N, M, the spectral properties of $\boldsymbol{A}$, $\boldsymbol{C}$, etc. Moreover, it can be shown that a_0 is positive. To obtain some insight, if we focus only on the first row of the inequality system on $\|\underline{\widehat{\boldsymbol{v}}}(t+1)\|$, we obtain that

$$\|\underline{\widehat{\boldsymbol{v}}}(t+1)\| \leq \big(1 - \gamma_1 a_0 + \gamma_1^2 a_1\big) \max_{(t-2M)_+ \leq q \leq t} \|\underline{\widehat{\boldsymbol{v}}}(q)\| + \gamma_1 a_2 \max_{(t-2M)_+ \leq q \leq t} \mathcal{E}_c(q) \; .$$

In fact, when the contribution from $\mathcal{E}_c(q)$ can be ignored, then applying (Feyzmahdavian et al., 2014, Lemma 3) shows that $\|\underline{\widehat{\boldsymbol{v}}}(t+1)\|$ converges linearly if $-\gamma_1 a_0 + \gamma_1^2 a_1 < 0$, which is possible as $a_0 > 0$. Therefore, if $\mathcal{E}_c(t)$ also converges linearly, then it is anticipated that $\mathcal{E}_g(t)$ would do so as well. In other words, the linear convergence of $\|\underline{\widehat{\boldsymbol{v}}}(t)\|$, $\mathcal{E}_c(t)$ and $\mathcal{E}_g(t)$ are all coupled in the inequality system (21).

Formalizing the above observations, Lemma 1 in the supplementary material shows a sufficient condition on γ_1 for linear convergence. Specifically, if there exists $\gamma_1 > 0$ such that the spectral radius of $\boldsymbol{Q}(\gamma_1)$ in (22) is strictly less than one, then each of the Lyapunov functions, $\|\underline{\widehat{\boldsymbol{v}}}(t)\|$, $\mathcal{E}_c(t)$, $\mathcal{E}_g(t)$, would enjoy linear convergence. Furthermore, Lemma 2 in the supplementary material gives an existence proof for such an γ_1 to exist. This concludes the proof and we shall relegate the details to the Appendix.

Remark on the Proof While delayed inequality system has been studied in (Feyzmahdavian et al., 2014; Gurbuzbalaban et al., 2017) for optimization algorithms, the coupled system in (21) is a non-trivial generalization of the above. Importantly, the challenge here is due to the asymmetry of the system matrix $\boldsymbol{Q}$ and the maximum over the past sequences on the right hand side are taken *independently*. To the best of our knowledge, our result is the first to characterize the (linear) convergence of such coupled and delayed system of inequalities. We believe that our analysis would be useful for the future study of similar algorithms.

Extension Our analysis and algorithm may in fact be applied to solve general problems that involves multi-agent and finite-sum optimization, e.g.,

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \; J(\boldsymbol{\theta}) := \tfrac{1}{NM} \textstyle\sum_{i=1}^{N} \sum_{p=1}^{M} J_{i,p}(\boldsymbol{\theta}) \; . \tag{23}$$

For instance, these problems may arise in empirical risk minimization for a multi-agent setting, where data samples are kept independently by the agents. Our analysis, especially with the convergence results for inequality systems of the form (21), can be applied to study a similar double averaging algorithm with just the primal variable. In particular, we only require the sum function $J(\boldsymbol{\theta})$ to be strongly convex, and the objective functions $J_{i,p}(\cdot)$ to be smooth in order to achieve linear convergence. We believe that such extension is of independent interest to the community.

At the time of submission, a recent work (Pu and Nedić, 2018) applied a related double averaging distributed algorithm to a *stochastic version* of (23). However, their convergence rate is sub-linear as they considered a stochastic optimization setting.

4 Numerical Experiments

To verify the performance of our proposed method, we conduct an experiment on the `mountaincar` dataset (Sutton and Barto, 1998) under a setting similar to (Du et al., 2017) – to collect the dataset, we ran Sarsa with $d = 300$ features to obtain the policy, then we generate the trajectories of actions and states according to the policy with M samples. For each sample p, we generate the local reward, $R_i(s_{p,i}, a_{p,i})$ by assigning a random portion for the reward to each agent such that the average of the local rewards equals to that of $\mathcal{R}_c(\boldsymbol{s}_p, \boldsymbol{a}_p)$.

We compare our method to several centralized methods – PDBG is the primal-dual gradient descent method in (11), GTD2 (Sutton et al., 2009), and SAGA (Du et al., 2017). Notably, SAGA has linear convergence while only requiring an incremental update step of low complexity. For PD-DistIAG, we simulate a communication network with $N = 10$ agents, connected on an Erdos-Renyi graph generated with connectivity of 0.2; for the step sizes, we set $\gamma_1 = 0.005/\lambda_{\max}(\hat{\boldsymbol{A}})$, $\gamma_2 = 2.5 \times 10^{-3}/\lambda_{\max}(\hat{\boldsymbol{C}})$.

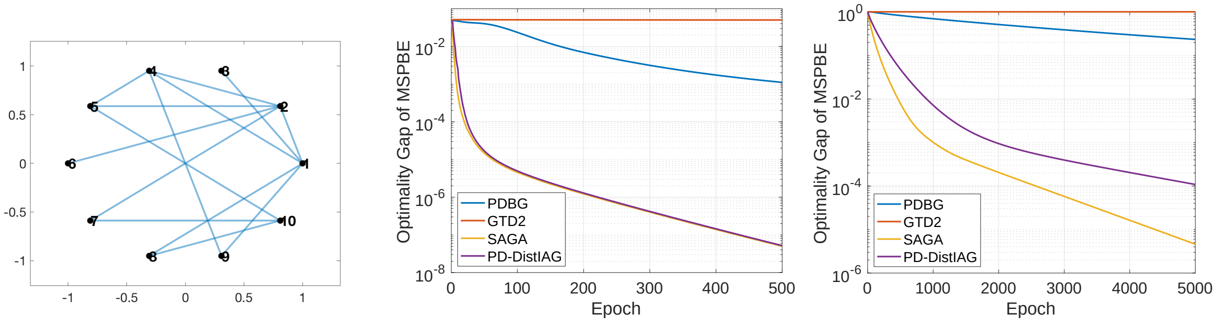

Figure 1: Experiment with `mountaincar` dataset. For this problem, we have $d = 300$, $M = 5000$ samples, and there are $N = 10$ agents. (Left) Graph Topology. (Middle) $\rho = 0.01$. (Right) $\rho = 0$.

Figure 1 compares the optimality gap in terms of MSPBE of different algorithms against the epoch number, defined as (t/M). For PD-DistIAG, we compare its optimality gap in MSPBE as the average objective, *i.e.*, it is $(1/N)\sum_{i=1}^{N} \mathsf{MSPBE}(\boldsymbol{\theta}_i^t) - \mathsf{MSPBE}(\boldsymbol{\theta}^\star)$. As seen in the left panel, when the regularization factor is high with $\rho > 0$, the convergence speed of PD-DistIAG is comparable to that of SAGA; meanwhile with $\rho = 0$, the PD-DistIAG converges at a slower speed than SAGA. Nevertheless, in both cases, the PD-DistIAG method converges faster than the other methods except for SAGA. Additional experiments are presented in the supplementary materials to compare the performance at different topology and regularization parameter.

Conclusion In this paper, we have studied the policy evaluation problem in *multi-agent* reinforcement learning. Utilizing Fenchel duality, a double averaging scheme is proposed to tackle the primal-dual, multi-agent, and finite-sum optimization arises. The proposed algorithm, PD-DistIAG method, is the first of its kind to demonstrate linear convergence in such setting. Future work includes characterizing the exact linear convergence rate, and studying the impact of various sampling schemes used in the distributed algorithm.

A Proof of Theorem 1

Notation We first define a set of notations pertaining to the proof. For any $\beta > 0$, observe that the primal-dual optimal solution, $(\boldsymbol{\theta}^\star, \{\boldsymbol{w}_i^\star\}_{i=1}^N)$, to the optimization problem (10) can be written as

$$\underbrace{\begin{pmatrix} \rho \boldsymbol{I} & \sqrt{\frac{\beta}{N}}\hat{\boldsymbol{A}}^\top & \cdots & \sqrt{\frac{\beta}{N}}\hat{\boldsymbol{A}}^\top \\ -\sqrt{\frac{\beta}{N}}\hat{\boldsymbol{A}} & \beta\hat{\boldsymbol{C}} & \cdots & \cdots \\ \vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ -\sqrt{\frac{\beta}{N}}\hat{\boldsymbol{A}} & \cdots & \cdots & \beta\hat{\boldsymbol{C}} \end{pmatrix}}_{:=\boldsymbol{G}} \begin{pmatrix} \boldsymbol{\theta}^\star \\ \frac{1}{\sqrt{\beta N}}\boldsymbol{w}_1^\star \\ \vdots \\ \frac{1}{\sqrt{\beta N}}\boldsymbol{w}_N^\star \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{b}_1 \\ \vdots \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{b}_N \end{pmatrix}, \tag{24}$$

where we denote the matrix on the left hand side as $\boldsymbol{G}$. This equation can be obtained by checking the first-order optimality condition. In addition, for any $p \in \{1, \ldots, M\}$, we define the $\boldsymbol{G}_p$ as

$$\boldsymbol{G}_p := \begin{pmatrix} \rho \boldsymbol{I} & \sqrt{\frac{\beta}{N}}\boldsymbol{A}_p^\top & \cdots & \sqrt{\frac{\beta}{N}}\boldsymbol{A}^\top \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{A}_p & \beta\boldsymbol{C}_p & \cdots & \cdots \\ \vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{A}_p & \cdots & \cdots & \beta\boldsymbol{C}_p \end{pmatrix}. \tag{25}$$

By definition, $\boldsymbol{G}$ is the sample average of $\{\boldsymbol{G}_p\}_{p=1}^M$. Define $\bar{\boldsymbol{\theta}}(t) := (1/N)\sum_{i=1}^N \boldsymbol{\theta}_i^t$ as the average of the local parameters at iteration t. Furthermore, we define

$$\boldsymbol{h}_{\boldsymbol{\theta}}(t) := \rho\bar{\boldsymbol{\theta}}(t) + \frac{1}{N}\sum_{i=1}^N \hat{\boldsymbol{A}}^\top \boldsymbol{w}_i^t, \qquad \boldsymbol{g}_{\boldsymbol{\theta}}(t) := \frac{1}{NM}\sum_{i=1}^N\sum_{p=1}^M \left(\rho\boldsymbol{\theta}_i^{\tau_p^t} + \boldsymbol{A}_p^\top \boldsymbol{w}_i^{\tau_p^t}\right), \tag{26}$$

$$\boldsymbol{h}_{\boldsymbol{w}_i}(t) := \hat{\boldsymbol{A}}\bar{\boldsymbol{\theta}}(t) - \hat{\boldsymbol{C}}\boldsymbol{w}_i^t - \hat{\boldsymbol{b}}_i, \qquad \boldsymbol{g}_{\boldsymbol{w}_i}(t) := \frac{1}{M}\sum_{p=1}^M \left(\boldsymbol{A}_p\boldsymbol{\theta}_i^{\tau_p^t} - \boldsymbol{C}_p\boldsymbol{w}_i^{\tau_p^t} - \boldsymbol{b}_{p,i}\right), \tag{27}$$

where $\boldsymbol{h}_{\boldsymbol{\theta}}(t)$ and $\boldsymbol{h}_{\boldsymbol{w}}(t) := [\boldsymbol{h}_{\boldsymbol{w}_1}(t), \cdots, \boldsymbol{h}_{\boldsymbol{w}_N}(t)]$ represent the gradients evaluated by a *centralized* and *batch* algorithm. Note that $\boldsymbol{g}_{\boldsymbol{\theta}}(t)$ defined in (26) coincides with that in (16). Using Lemma 1, it can be checked that $\bar{\boldsymbol{\theta}}(t+1) = \bar{\boldsymbol{\theta}}(t) - \gamma_1\boldsymbol{g}_{\boldsymbol{\theta}}(t)$ and $\boldsymbol{w}_i^{t+1} = \boldsymbol{w}_i^t - \gamma_2\boldsymbol{g}_{\boldsymbol{w}_i}(t)$ for all $t \geq 1$. That is, $\bar{\boldsymbol{\theta}}(t+1)$ and $\boldsymbol{w}_i^{t+1}$ can be viewed as primal-dual updates using $\boldsymbol{g}_{\boldsymbol{\theta}}(t)$ and $\boldsymbol{g}_{\boldsymbol{w}_i}(t)$, which are decentralized counterparts of gradients $\boldsymbol{h}_{\boldsymbol{\theta}}(t)$ and $\boldsymbol{h}_{\boldsymbol{w}_i}(t)$ defined in (26) (27).

To simplify the notation, hereafter, we define vectors $\underline{\boldsymbol{h}}(t)$, $\underline{\boldsymbol{g}}(t)$, and $\underline{\boldsymbol{v}}(t)$ by

$$\underline{\boldsymbol{h}}(t) := \begin{pmatrix} \boldsymbol{h}_{\boldsymbol{\theta}}(t) \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{h}_{\boldsymbol{w}_1}(t) \\ \vdots \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{h}_{\boldsymbol{w}_N}(t) \end{pmatrix}, \ \underline{\boldsymbol{g}}(t) := \begin{pmatrix} \boldsymbol{g}_{\boldsymbol{\theta}}(t) \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{g}_{\boldsymbol{w}_1}(t) \\ \vdots \\ -\sqrt{\frac{\beta}{N}}\boldsymbol{g}_{\boldsymbol{w}_N}(t) \end{pmatrix}, \ \underline{\boldsymbol{v}}(t) := \begin{pmatrix} \bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star \\ \frac{1}{\sqrt{\beta N}}(\boldsymbol{w}_1^t - \boldsymbol{w}_1^\star) \\ \vdots \\ \frac{1}{\sqrt{\beta N}}(\boldsymbol{w}_N^t - \boldsymbol{w}_N^\star) \end{pmatrix}. \tag{28}$$

Using (24), it can be verified that (see the detailed derivation in Section A.2)

$$\underline{\boldsymbol{h}}(t) = \boldsymbol{G}\underline{\boldsymbol{v}}(t). \tag{29}$$

By adopting the analysis in (Du et al., 2017) and under Assumption 2, it can be shown that with

$$\beta := \frac{8(\rho + \lambda_{\mathsf{max}}(\hat{\boldsymbol{A}}^\top\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{A}}))}{\lambda_{\mathsf{min}}(\hat{\boldsymbol{C}})},$$

then $\boldsymbol{G}$ is full rank with its eigenvalues satisfying

$$\lambda_{\mathsf{max}}(\boldsymbol{G}) \leq \left|\frac{\lambda_{\mathsf{max}}(\hat{\boldsymbol{C}})}{\lambda_{\mathsf{min}}(\hat{\boldsymbol{C}})}\right| \lambda_{\mathsf{max}}(\rho\boldsymbol{I} + \hat{\boldsymbol{A}}^\top\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{A}}), \qquad \lambda_{\mathsf{min}}(\boldsymbol{G}) \geq \frac{8}{9}\lambda_{\mathsf{min}}(\hat{\boldsymbol{A}}^\top\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{A}}) > 0. \tag{30}$$

Moreover, let $\boldsymbol{G} := \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1}$. be the eigen-decomposition of $\boldsymbol{G}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix consists of the eigenvalues of $\boldsymbol{G}$, and the columns of $\boldsymbol{U}$ are the eigenvectors. Then, $\boldsymbol{U}$ is full rank with

$$\|\boldsymbol{U}\| \leq 8(\rho + \lambda_{\mathsf{max}}(\hat{\boldsymbol{A}}^\top\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{A}}))\left|\frac{\lambda_{\mathsf{max}}(\hat{\boldsymbol{C}})}{\lambda_{\mathsf{min}}(\hat{\boldsymbol{C}})}\right|, \qquad \|\boldsymbol{U}^{-1}\| \leq \frac{1}{\rho + \lambda_{\mathsf{max}}(\hat{\boldsymbol{A}}^\top\hat{\boldsymbol{C}}^{-1}\hat{\boldsymbol{A}})}. \tag{31}$$

Furthermore, we also define the following upper bounds on the spectral norms

$$G := \|\boldsymbol{G}\|, \quad \overline{G} := \max_{p=1,\ldots,M}\|\boldsymbol{G}_p\|, \quad \overline{A} := \max_{p=1,\ldots,M}\|\boldsymbol{A}_p\|, \quad \overline{C} := \max_{p=1,\ldots,M}\|\boldsymbol{C}_p\|. \tag{32}$$

Lastly, we define the following two Lyapunov functions

$$\mathcal{E}_c(t) := \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{\theta}_i^t - \bar{\boldsymbol{\theta}}(t)\|, \qquad \mathcal{E}_g(t) := \frac{1}{N}\sum_{i=1}^N \|\boldsymbol{s}_i^t - \boldsymbol{g}_{\boldsymbol{\theta}}(t)\|. \tag{33}$$

Convergence Analysis We denote that $\gamma_1 = \gamma$ and $\gamma_2 = \beta\gamma$. To study the linear convergence of the PD-DistIAG method, our first step is to establish a bound on the difference from the primal-dual optimal solution, $\underline{\boldsymbol{v}}(t)$. Observe with the choice of our step size ratio,

$$\underline{\boldsymbol{v}}(t+1) = (\boldsymbol{I} - \gamma \boldsymbol{G})\underline{\boldsymbol{v}}(t) + \gamma\big(\underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{g}}(t)\big) \,. \tag{34}$$

Consider the difference vector $\underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{g}}(t)$. Its first block can be evaluated as

$$\begin{aligned}\big[\underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{g}}(t)\big]_1 &= \frac{1}{NM}\sum_{i=1}^{N}\sum_{p=1}^{M}\Big(\rho\big(\bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}_i^{\tau_p^t}\big) + \boldsymbol{A}_p^\top\big(\boldsymbol{w}_i^t - \boldsymbol{w}_i^{\tau_p^t}\big)\Big) \\ &= \frac{1}{NM}\sum_{i=1}^{N}\sum_{p=1}^{M}\Big(\rho\big(\bar{\boldsymbol{\theta}}(t) - \bar{\boldsymbol{\theta}}(\tau_p^t)\big) + \boldsymbol{A}_p^\top\big(\boldsymbol{w}_i^t - \boldsymbol{w}_i^{\tau_p^t}\big)\Big) + \frac{\rho}{NM}\sum_{i=1}^{N}\sum_{p=1}^{M}\big(\bar{\boldsymbol{\theta}}(\tau_p^t) - \boldsymbol{\theta}_i^{\tau_p^t}\big) \,.\end{aligned} \tag{35}$$

Meanwhile, for any $i \in \{1, \ldots, N\}$, the $(i+1)$-th block is

$$\begin{aligned}\big[\underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{g}}(t)\big]_{i+1} &= \sqrt{\frac{\beta}{N}}\frac{1}{M}\sum_{p=1}^{M}\Big(\boldsymbol{A}_p\big(\boldsymbol{\theta}_i^{\tau_p^t} - \bar{\boldsymbol{\theta}}(t)\big) + \boldsymbol{C}_p\big(\boldsymbol{w}_i^t - \boldsymbol{w}_i^{\tau_p^t}\big)\Big) \\ &= \sqrt{\frac{\beta}{N}}\frac{1}{M}\sum_{p=1}^{M}\Big(\boldsymbol{A}_p\big(\bar{\boldsymbol{\theta}}(\tau_p^t) - \bar{\boldsymbol{\theta}}(t)\big) + \boldsymbol{C}_p\big(\boldsymbol{w}_i^t - \boldsymbol{w}_i^{\tau_p^t}\big)\Big) + \sqrt{\frac{\beta}{N}}\frac{1}{M}\sum_{p=1}^{M}\boldsymbol{A}_p\big(\boldsymbol{\theta}_i^{\tau_p^t} - \bar{\boldsymbol{\theta}}(\tau_p^t)\big) \,.\end{aligned} \tag{36}$$

For ease of presentation, we stack up and denote the residual terms (related to consensus error) in (35) and (36) as the vector $\underline{\boldsymbol{\mathcal{E}}}_c(t)$. That is, the first block of $\underline{\boldsymbol{\mathcal{E}}}_c(t)$ is $\rho/(NM) \cdot \sum_{i=1}^N \sum_{p=1}^M \big(\bar{\boldsymbol{\theta}}(\tau_p^t) - \boldsymbol{\theta}_i^{\tau_p^t}\big)$, and the remaining blocks are given by $\sqrt{\beta/N} \cdot 1/M \cdot \sum_{p=1}^M \boldsymbol{A}_p\big(\boldsymbol{\theta}_i^{\tau_p^t} - \bar{\boldsymbol{\theta}}(\tau_p^t)\big)$, $\forall i \in \{1, \ldots, N\}$. Then by the definition of $\boldsymbol{G}_p$ in (25), we obtain the following simplification:

$$\underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{g}}(t) - \underline{\boldsymbol{\mathcal{E}}}_c(t) = \frac{1}{M}\sum_{p=1}^{M}\boldsymbol{G}_p\Big(\textstyle\sum_{j=\tau_p^t}^{t-1}\underline{\Delta\boldsymbol{v}}(j)\Big), \tag{37}$$

where we define

$$\underline{\Delta\boldsymbol{v}}(j) := \begin{pmatrix} \bar{\boldsymbol{\theta}}(j+1) - \bar{\boldsymbol{\theta}}(j) \\ \frac{1}{\sqrt{\beta N}}\big(\boldsymbol{w}_1^{j+1} - \boldsymbol{w}_1^{j}\big) \\ \vdots \\ \frac{1}{\sqrt{\beta N}}\big(\boldsymbol{w}_N^{j+1} - \boldsymbol{w}_N^{j}\big) \end{pmatrix} . \tag{38}$$

Clearly, we can express $\Delta\underline{\boldsymbol{v}}(j)$ as $\Delta\underline{\boldsymbol{v}}(j) = \underline{\boldsymbol{v}}(j+1) - \underline{\boldsymbol{v}}(j)$ with $\underline{\boldsymbol{v}}(t)$ defined in (28). Combining (29) and (34), we can also write $\Delta\underline{\boldsymbol{v}}(j)$ in (38) as

$$\Delta\underline{\boldsymbol{v}}(j) = \gamma\big[\underline{\boldsymbol{h}}(j) - \underline{\boldsymbol{g}}(j)\big] - \gamma\underline{\boldsymbol{h}}(j) \,. \tag{39}$$

Denoting $\underline{\widehat{\boldsymbol{v}}}(t) := \boldsymbol{U}^{-1}\underline{\boldsymbol{v}}(t)$, multiplying $\boldsymbol{U}^{-1}$ on both sides of (34) yields

$$\underline{\widehat{\boldsymbol{v}}}(t+1) = (\boldsymbol{I} - \gamma\boldsymbol{\Lambda})\underline{\widehat{\boldsymbol{v}}}(t) + \gamma\, \boldsymbol{U}^{-1}\big(\underline{\boldsymbol{h}}(t) - \underline{\boldsymbol{g}}(t)\big) \,. \tag{40}$$

Combining (37), (39), and (40), by triangle inequality, we have

$$\|\underline{\widehat{\boldsymbol{v}}}(t+1)\| \le \big\|\boldsymbol{I} - \gamma\boldsymbol{\Lambda}\big\|\|\underline{\widehat{\boldsymbol{v}}}(t)\| + \gamma\|\boldsymbol{U}^{-1}\|\bigg\{\|\underline{\boldsymbol{\mathcal{E}}}_c(t)\| + \frac{\gamma\overline{G}}{M}\sum_{p=1}^{M}\sum_{j=\tau_p^t}^{t-1}\big[\|\underline{\boldsymbol{h}}(j)\| + \|\underline{\boldsymbol{h}}(j) - \underline{\boldsymbol{g}}(j)\|\big]\bigg\}, \tag{41}$$

where $\overline{G}$ appears in (32) and $\underline{\boldsymbol{\mathcal{E}}}_c(t)$ is the residue term of the consensus. Furthermore, simplifying the right-hand side of (41) yields

$$\begin{aligned}
\|\widehat{\underline{\boldsymbol{v}}}(t+1)\| &\leq \big\|\boldsymbol{I}-\gamma\boldsymbol{\Lambda}\big\|\|\widehat{\underline{\boldsymbol{v}}}(t)\| + \gamma\|\boldsymbol{U}^{-1}\|\bigg\{\|\underline{\boldsymbol{\mathcal{E}}}_c(t)\| + \gamma\overline{G}\sum_{j=(t-M)_+}^{t-1}\big[\|\underline{\boldsymbol{h}}(j)\| + \|\underline{\boldsymbol{h}}(j)-\underline{\boldsymbol{g}}(j)\|\big]\bigg\} \\
&\leq \big\|\boldsymbol{I}-\gamma\boldsymbol{\Lambda}\big\|\|\widehat{\underline{\boldsymbol{v}}}(t)\| + \gamma\|\boldsymbol{U}^{-1}\|\Bigg(\|\underline{\boldsymbol{\mathcal{E}}}_c(t)\| \\
&\qquad + \gamma\overline{G}\sum_{j=(t-M)_+}^{t-1}\bigg\{\|\underline{\boldsymbol{\mathcal{E}}}_c(j)\| + G\|\boldsymbol{U}\|\|\widehat{\underline{\boldsymbol{v}}}(j)\| + \overline{G}\|\boldsymbol{U}\|\cdot\sum_{\ell=(j-M)_+}^{j-1}\big[\|\widehat{\underline{\boldsymbol{v}}}(\ell+1)\| + \|\widehat{\underline{\boldsymbol{v}}}(\ell)\|\big]\bigg\}\Bigg)\,.
\end{aligned} \tag{42}$$

Moreover, by definition, we can upper bound $\|\underline{\boldsymbol{\mathcal{E}}}_c(t)\|$ by

$$\|\underline{\boldsymbol{\mathcal{E}}}_c(t)\| \leq \frac{1}{M}\sum_{p=1}^{M}\bigg[\big(\rho+\overline{A}\sqrt{\beta N}\big)\cdot\bigg(\frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{\tau_p^t} - \bar{\boldsymbol{\theta}}(\tau_p^t)\|\bigg)\bigg] \leq \big(\rho+\overline{A}\sqrt{\beta N}\big)\max_{(t-M)_+\leq q\leq t}\mathcal{E}_c(q)\,. \tag{43}$$

Thus, combining (42) and (43), we bound $\|\widehat{\underline{\boldsymbol{v}}}(t+1)\|$ by

$$\|\widehat{\underline{\boldsymbol{v}}}(t+1)\| \leq \big\|\boldsymbol{I}-\gamma\boldsymbol{\Lambda}\big\|\|\widehat{\underline{\boldsymbol{v}}}(t)\| + C_1(\gamma)\max_{(t-2M)_+\leq q\leq t-1}\|\widehat{\underline{\boldsymbol{v}}}(q)\| + C_2(\gamma)\max_{(t-2M)_+\leq q\leq t}\mathcal{E}_c(q)\,, \tag{44}$$

where constants $C_1(\gamma)$ and $C_2(\gamma)$ are given by

$$C_1(\gamma) := \gamma^2\ \|\boldsymbol{U}\|\cdot\|\boldsymbol{U}^{-1}\|\cdot\overline{G}M\cdot\big(G+2\overline{G}M\big)\,, \qquad C_2(\gamma) := \gamma\cdot\|\boldsymbol{U}^{-1}\|\cdot\big(1+\gamma\overline{G}M\big)\cdot\big(\rho+\overline{A}\sqrt{\beta N}\big)\,.$$

Notice that since $\boldsymbol{U}^{-1}$ is full rank, the norm $\|\widehat{\underline{\boldsymbol{v}}}(t)\|$ is equivalent to $\|\overline{\boldsymbol{\theta}}(t)-\boldsymbol{\theta}^\star\|^2 + (1/\beta N)\sum_{i=1}^N\|\boldsymbol{w}_i^\star - \boldsymbol{w}_i^t\|^2$, *i.e.,* the optimality gap at the t-th iteration.

Next, seen from (44), it remains to upper bound $\mathcal{E}_c(t+1)$ defined in (33). By the primal-dual update in (15) and the triangle inequality, we have

$$\mathcal{E}_c(t+1) = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i^{t+1} - \bar{\boldsymbol{\theta}}(t+1)\| \leq \frac{1}{N}\sum_{i=1}^{N}\bigg(\bigg\|\sum_{j=1}^{N}W_{ij}(\boldsymbol{\theta}_j^t - \bar{\boldsymbol{\theta}}(t))\bigg\| + \gamma\|\boldsymbol{s}_i^t - \boldsymbol{g}_{\boldsymbol{\theta}}(t)\|\bigg). \tag{45}$$

Moreover, notice that we have $1/N\cdot\sum_{i=1}^N\boldsymbol{\theta}_i^t = \bar{\boldsymbol{\theta}}(t)$ for any $t\geq 1$ and that $\lambda := \max_{j=1,\ldots,N}\sum_{i=1}^N|W_{ij} - (1/N)| = \|\boldsymbol{W}-(1/N)\mathbf{1}\mathbf{1}^\top\|_{1,\infty} < 1$. The right-hand side of (45) can be bounded by

$$\begin{aligned}
\mathcal{E}_c(t+1) &\leq \frac{1}{N}\sum_{i=1}^{N}\bigg(\bigg\|\sum_{j=1}^{N}\big(W_{ij}-\frac{1}{N}\big)(\boldsymbol{\theta}_j^t - \bar{\boldsymbol{\theta}}(t))\bigg\| + \gamma\|\boldsymbol{s}_i^t - \boldsymbol{g}_{\boldsymbol{\theta}}(t)\|\bigg) \\
&\leq \lambda\ \mathcal{E}_c(t) + \frac{\gamma}{N}\sum_{i=1}^{N}\|\boldsymbol{s}_i^t - \boldsymbol{g}_{\boldsymbol{\theta}}(t)\| = \lambda\ \mathcal{E}_c(t) + \gamma\ \mathcal{E}_g(t)\,,
\end{aligned} \tag{46}$$

where the Lyapunov function $\mathcal{E}_g(t)$ is defined in (33).

Finally, seen from (43) and (46), to conclude the proof, we need to further upper bound $\mathcal{E}_g(t+1)$. To simplify the notation, for any $i\in[N]$ and any $t\geq 1$, we define $\Delta_i(t|t+1)$ by

$$\Delta_i(t|t+1) := \bigg\|\frac{1}{M}\Big(\nabla J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{t+1};\boldsymbol{w}_i^{t+1}) - \nabla J_{i,p_{t+1}}(\boldsymbol{\theta}_i^{\tau_{p_{t+1}}^t};\boldsymbol{w}_i^{\tau_{p_{t+1}}^t})\Big) + \boldsymbol{g}_{\boldsymbol{\theta}}(t) - \boldsymbol{g}_{\boldsymbol{\theta}}(t+1)\bigg\|\,. \tag{47}$$

Then combining (15), (33), and (47), we have

$$\mathcal{E}_g(t+1) \leq \frac{1}{N}\sum_{i=1}^{N}\bigg[\bigg\|\sum_{j=1}^{N}\big(W_{ij}-\frac{1}{N}\big)\big(\boldsymbol{s}_j^t - \boldsymbol{g}_{\boldsymbol{\theta}}(t)\big)\bigg\| + \Delta_i(t|t+1)\bigg] \leq \lambda\,\mathcal{E}_g(t) + \frac{1}{N}\sum_{i=1}^{N}\Delta_i(t|t+1)\,, \tag{48}$$

Besides, $\Delta_i(t|t+1)$ can be upper bounded by

$$\begin{aligned}\Delta_i(t|t+1) \leq \frac{\rho}{M}\Big(\|\boldsymbol{\theta}_i^{t+1} - \bar{\boldsymbol{\theta}}(t+1)\| + \|\boldsymbol{\theta}_i^{\tau^t_{p_{t+1}}} - \bar{\boldsymbol{\theta}}(\tau^t_{p_{t+1}})\|\Big) \\ + \frac{\|\boldsymbol{A}^\top_{p_{t+1}}\|}{M}\left(\left\|\boldsymbol{w}_i^{t+1} - \frac{1}{N}\sum_{j=1}^N \boldsymbol{w}_j^{t+1} - \boldsymbol{w}_i^{\tau^t_{p_{t+1}}} + \frac{1}{N}\sum_{j=1}^N \boldsymbol{w}_j^{\tau^t_{p_{t+1}}}\right\|\right) .\end{aligned} \tag{49}$$

To bound the last term on the right-hand side of (49), note that it holds for all $t' \leq t$ that

$$\boldsymbol{w}_i^{t+1} - \boldsymbol{w}_i^{t'} = -\frac{\gamma}{\beta M}\sum_{\ell=t'}^{t}\sum_{p=1}^{M}\Big[\boldsymbol{A}_p(\boldsymbol{\theta}_i^{\tau_p^\ell} - \boldsymbol{\theta}^\star) - \boldsymbol{C}_p(\boldsymbol{w}_i^{\tau_p^\ell} - \boldsymbol{w}_i^\star)\Big] .$$

Thus, by triangle inequality, we have

$$\begin{aligned}&\frac{1}{N}\sum_{i=1}^N\left[\frac{\|\boldsymbol{A}^\top_{p_{t+1}}\|}{M}\left(\left\|\boldsymbol{w}_i^{t+1} - \frac{1}{N}\sum_{j=1}^N \boldsymbol{w}_j^{t+1} - \boldsymbol{w}_i^{\tau^t_{p_{t+1}}} + \frac{1}{N}\sum_{j=1}^N \boldsymbol{w}_j^{\tau^t_{p_{t+1}}}\right\|\right)\right] \\ &\quad\leq \frac{2\gamma\overline{A}}{\beta N M^2}\sum_{i=1}^N\sum_{\ell=(t-M)_+}^{t}\sum_{p=1}^{M}\Big(\|\boldsymbol{A}_p(\boldsymbol{\theta}_i^{\tau_p^\ell} - \boldsymbol{\theta}^\star) - \boldsymbol{C}_p(\boldsymbol{w}_i^{\tau_p^\ell} - \boldsymbol{w}_i^\star)\|\Big) \\ &\quad\leq \frac{2\gamma\overline{A}}{\beta N M}\sum_{i=1}^N\sum_{\ell=(t-M)_+}^{t}\left[\max_{(\ell-M)_+\leq q\leq\ell}\Big(\overline{A}\|\boldsymbol{\theta}_i^q - \boldsymbol{\theta}^\star\| + \overline{C}\|\boldsymbol{w}_i^q - \boldsymbol{w}_i^\star\|\Big)\right] .\end{aligned} \tag{50}$$

Thus, combining (46), (49), (50), and the definition of $\mathcal{E}_c$ in (33), we have

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^N\Delta_i(t|t+1) &\leq \frac{\rho}{M}\big[\mathcal{E}_c(\tau^t_{p_{t+1}}) + \mathcal{E}_c(t+1)\big] + \frac{2\gamma\overline{A}(M+1)}{\beta N M}\sum_{i=1}^N\max_{(t-2M)_+\leq q\leq t}\Big(\overline{A}\|\boldsymbol{\theta}_i^q - \boldsymbol{\theta}^\star\| + \overline{C}\|\boldsymbol{w}_i^q - \boldsymbol{w}_i^\star\|\Big) \\ &\leq \frac{\rho}{M}\big[\mathcal{E}_c(\tau^t_{p_{t+1}}) + \lambda\,\mathcal{E}_c(t) + \gamma\,\mathcal{E}_g(t)\big] \\ &\quad + \frac{2\gamma\overline{A}(M+1)}{\beta M}\max_{(t-2M)_+\leq q\leq t}\left(\overline{A}\,\mathcal{E}_c(q) + \overline{A}\,\|\bar{\boldsymbol{\theta}}(q) - \boldsymbol{\theta}^\star\| + \frac{\overline{C}}{N}\sum_{i=1}^N\|\boldsymbol{w}_i^q - \boldsymbol{w}_i^\star\|\right) .\end{aligned} \tag{51}$$

Finally, combining (48) and (51), we obtain that

$$\begin{aligned}\mathcal{E}_g(t+1) \leq \Big(\lambda + \frac{\gamma\rho}{M}\Big)\,\mathcal{E}_g(t) + \left[\frac{2\gamma\overline{A}^2(M+1)}{\beta M} + \frac{2(1+\lambda)}{M}\right]\max_{(t-2M)_+\leq q\leq t}\mathcal{E}_c(q) \\ + \frac{2\gamma\overline{A}(M+1)}{\beta M}\max_{(t-2M)_+\leq q\leq t}\left(\overline{A}\,\|\bar{\boldsymbol{\theta}}(q) - \boldsymbol{\theta}^\star\| + \frac{\overline{C}}{N}\sum_{i=1}^N\|\boldsymbol{w}_i^q - \boldsymbol{w}_i^\star\|\right) .\end{aligned} \tag{52}$$

To bound the last term on the right-hand side of (52), For all q, we observe that:

$$\begin{aligned}\left(\overline{A}\,\|\bar{\boldsymbol{\theta}}(q) - \boldsymbol{\theta}^\star\| + \frac{\overline{C}}{N}\sum_{i=1}^N\|\boldsymbol{w}_i^q - \boldsymbol{w}_i^\star\|\right)^2 &\leq (N+1)(\overline{A})^2\left[\|\bar{\boldsymbol{\theta}}(q) - \boldsymbol{\theta}^\star\|^2 + \beta\Big(\frac{\overline{C}}{\overline{A}}\Big)^2\frac{1}{\beta N}\sum_{i=1}^N\|\boldsymbol{w}_i^q - \boldsymbol{w}_i^\star\|^2\right] \\ &\leq (N+1)\,\|\boldsymbol{U}\|\max\Big\{(\overline{A})^2, \beta(\overline{C})^2\Big\}\,\|\underline{\boldsymbol{v}}(q)\|^2 ,\end{aligned}$$

which further implies that

$$\begin{aligned}\mathcal{E}_g(t+1) \leq \Big(\lambda + \frac{\gamma\rho}{M}\Big)\,\mathcal{E}_g(t) + \Big(\frac{2\gamma\overline{A}^2(M+1)}{\beta M} + \frac{2(1+\lambda)}{M}\Big)\max_{(t-2M)_+\leq q\leq t}\mathcal{E}_c(q) \\ + \frac{2\gamma\overline{A}\sqrt{N+1}(M+1)}{\beta M}\,\|\boldsymbol{U}\|\max\{\overline{A}, \sqrt{\beta}\overline{C}\}\max_{(t-2M)_+\leq q\leq t}\|\underline{\widehat{\boldsymbol{v}}}(q)\| .\end{aligned} \tag{53}$$

Finally, combining (44), (46), (53) shows:

$$\begin{pmatrix} \|\widehat{\underline{\boldsymbol{v}}}(t+1)\| \\ \mathcal{E}_c(t+1) \\ \mathcal{E}_g(t+1) \end{pmatrix} \leq \boldsymbol{Q} \begin{pmatrix} \max_{(t-2M)_+ \leq q \leq t} \|\widehat{\underline{\boldsymbol{v}}}(q)\| \\ \max_{(t-2M)_+ \leq q \leq t} \mathcal{E}_c(q) \\ \max_{(t-2M)_+ \leq q \leq t} \mathcal{E}_g(q) \end{pmatrix} , \tag{54}$$

where the inequality sign is applied element-wisely, and $\boldsymbol{Q}$ is a non-negative 3×3 matrix, defined as:

$$\boldsymbol{Q} := \begin{pmatrix} \theta(\gamma) + \gamma^2 \|\boldsymbol{U}\| \|\boldsymbol{U}^{-1}\| \overline{G} M (G + 2\overline{G} M) & \gamma \|\boldsymbol{U}\| (1 + \gamma \overline{G} M)(\rho + \overline{A} \sqrt{\beta N}) & 0 \\ 0 & \lambda & \gamma \\ \frac{2\gamma \overline{A} \sqrt{N+1}(M+1)}{\beta M} \|\boldsymbol{U}\| \max\{\overline{A}, \sqrt{\beta}\, \overline{C}\} & \frac{2\gamma \overline{A}^2 (M+1)}{\beta M} + \frac{2(1+\lambda)}{M} & \left(\lambda + \frac{\gamma \rho}{M}\right) \end{pmatrix}, \tag{55}$$

where $\theta(\gamma) := \|\boldsymbol{I} - \gamma \boldsymbol{\Lambda}\| = \|\boldsymbol{I} - \gamma \boldsymbol{G}\|$. Note that the upper bounds for $\|\boldsymbol{U}\|$ and $\|\boldsymbol{U}^{-1}\|$ are provided in (31). Furthermore, also note that the eigenvalues of $\boldsymbol{G}$ are bounded in (30). We could set the stepsize γ to be sufficiently small such that such that $\theta(\gamma) := \|\boldsymbol{I} - \gamma \boldsymbol{G}\| < 1$.

Finally, we apply Lemmas 2 and 3 presented in Section A.1 to the recursive inequality in (53), which shows that each of $\|\underline{\boldsymbol{v}}(t)\|$, $\mathcal{E}_c(t)$, $\mathcal{E}_g(t)$ converges linearly with t. Therefore, we conclude the proof of Theorem 1.

A.1 Two Useful Lemmas

In this section, we present two auxiliary lemmas that are used in the proof of Theorem 1. Our first lemma establish the linear convergence of vectors satisfying recursive relations similar to (53), provided the spectral radius of $\boldsymbol{Q}$ is less than one. In addition, the second lemma verifies this condition for $\boldsymbol{Q}$ defined in in (55).

Lemma 2 *Consider a sequence of non-negative vectors $\{\boldsymbol{e}(t)\}_{t \geq 1} \subseteq \mathbb{R}^n$ whose evolution is characterized by $\boldsymbol{e}(t+1) \leq \boldsymbol{Q}\, \boldsymbol{e}([(t-M+1)_+, t])$ for all $t \geq 1$ and some fixed integer $M > 0$, where $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is a matrix whose entries are nonnegative, and we define*

$$\boldsymbol{e}(\mathcal{S}) := \begin{pmatrix} \max_{q \in \mathcal{S}}\ e_1(q) \\ \vdots \\ \max_{q \in \mathcal{S}}\ e_n(q) \end{pmatrix} \in \mathbb{R}^n \qquad \textit{for any subset } \mathcal{S} \subseteq \mathbb{N}\ .$$

Moreover, if $\boldsymbol{Q}$ irreducible in the sense that there exists an integer m such that the entries of $\boldsymbol{Q}^m$ are all positive, and the spectral radius of $\boldsymbol{Q}$, denoted by $\rho(\boldsymbol{Q})$, is strictly less than one, then for any $t \geq 1$, we have

$$\boldsymbol{e}(t) \leq \rho(\boldsymbol{Q})^{\lceil \frac{t-1}{M} \rceil} C_1 \boldsymbol{u}_1\ , \tag{56}$$

where $\boldsymbol{u}_1 \in \mathbb{R}^n_{++}$ is the top right eigenvector of $\boldsymbol{Q}$ and C_1 is a constant that depends on the initialization.

Proof. We shall prove the lemma using induction. By the Perron-Frobenius theorem, the eigenvector $\boldsymbol{u}_1$ associated with $\rho(\boldsymbol{Q})$ is unique and is an all-positive vector. Therefore, there exists C_1 such that

$$\boldsymbol{e}(1) \leq C_1\, \boldsymbol{u}_1\ . \tag{57}$$

Let us first consider the base case with $t = 2, ..., M+1$, i.e., $\lceil (t-1)/M \rceil = 1$. When $t = 2$, by (57) we have,

$$\boldsymbol{e}(2) \leq \boldsymbol{Q}\boldsymbol{e}(1) \leq C_1\, \boldsymbol{Q}\, \boldsymbol{u}_1 = \rho(\boldsymbol{Q})\, C_1\, \boldsymbol{u}_1\ , \tag{58}$$

which is valid as $\boldsymbol{Q}$, $\boldsymbol{e}(1)$, $\boldsymbol{u}_1$ are all non-negative. Furthermore, we observe that $\boldsymbol{e}(2) \leq C_1 \boldsymbol{u}_1$. Next when $t = 3$, we have

$$\boldsymbol{e}(3) \leq \boldsymbol{Q}\boldsymbol{e}([1,2]) \overset{(a)}{\leq} C_1\, \boldsymbol{Q}\, \boldsymbol{u}_1 = \rho(\boldsymbol{Q})\, C_1\, \boldsymbol{u}_1\ ,$$

where (a) is due to the non-negativity of vectors/matrix and the fact $\boldsymbol{e}(1), \boldsymbol{e}(2) \leq C_1 \boldsymbol{u}_1$ as shown in (58). Telescoping using similar steps, one can show $\boldsymbol{e}(t) \leq \rho(\boldsymbol{Q})\, C_1\, \boldsymbol{u}_1$ for any $t = 2, ..., M+1$.

For the induction step, let us assume that (56) holds true for any t up to $t = pM + 1$. That is, we assume that the result holds for all t such that $\lceil (t-1)/M \rceil \le p$. We shall show that it also holds for any $t = pM + 2, ..., (p+1)M + 1$, i.e., $\lceil (t-1)/M \rceil = p + 1$. Observe that

$$\boldsymbol{e}(pM+2) \le \boldsymbol{Q}\, \boldsymbol{e}([(p-1)M+2, pM+1]) \le C_1\, \rho(\boldsymbol{Q})^p \boldsymbol{Q} \boldsymbol{u}_1 = \rho(\boldsymbol{Q})^{p+1}\, C_1\, \boldsymbol{u}_1 \;, \tag{59}$$

where we have used the induction hypothesis. It is clear that (59) is equivalent to (56) with $t = pM + 2$. Similar upper bound can be obtained for $\boldsymbol{e}(pM + 3)$ as well. Repeating the same steps, we show that (56) is true for any $t = pM + 2, ..., (p + 1)M + 1$. Therefore, we conclude the proof of this lemma. **Q.E.D.**

The following Lemma shows that $\boldsymbol{Q}$ defined in (55) satisfies the conditions required in the previous lemma. Combining these two lemmas yields the final step of the proof of Theorem 1.

Lemma 3 *Consider the matrix* $\boldsymbol{Q}$ *defined in* (55)*, it can be shown that (a)* $\boldsymbol{Q}$ *is an irreducible matrix in* $\mathbb{R}^{3\times 3}$*; (b) there exists a sufficiently small* γ *such that* $\rho(\boldsymbol{Q}) < 1$*; and (c) as* $N, M \gg 1$ *and the graph is geometric, we can set* $\gamma = \mathcal{O}(1/\max\{N^2, M^2\})$ *and* $\rho(\boldsymbol{Q}) \le 1 - \mathcal{O}(1/\max\{N^2, M^2\})$*.*

Proof. Our proof is divided into three parts. The first part shows the straightforward irreducibility of $\boldsymbol{Q}$; the second part gives an upper bound to the spectral radius of $\boldsymbol{Q}$; and the last part derives an asymptotic bound on $\rho(\boldsymbol{Q})$ when $N, M \gg 1$.

Irreducibility of $\boldsymbol{Q}$ To see that $\boldsymbol{Q}$ is irreducible, notice that $\boldsymbol{Q}^2$ is a positive matrix, which could be verified by direct computation.

Spectral Radius of $\boldsymbol{Q}$ In the sequel, we compute an upper bound to the spectral radius of $\boldsymbol{Q}$, and show that if γ is sufficiently small, then its spectral radius will be strictly less than one. First we note that $\theta(\gamma) = 1 - \gamma\alpha$ for some $\alpha > 0$ and the network connectivity satisfies $\lambda < 1$. Also note that $\rho > 0$. For notational simplicity let us define the following

$$a_1 = \|\boldsymbol{U}\|\|\boldsymbol{U}^{-1}\|\, \overline{G}M(G + 2\overline{G}M), \quad a_2 = \|\boldsymbol{U}\|(\rho + \overline{A}\sqrt{\beta N}), \quad a_3 = \overline{G}M\|\boldsymbol{U}\|(\rho + \overline{A}\sqrt{\beta N})$$
$$a_4 = \frac{2\overline{A}\sqrt{N+1}(M+1)}{\beta M}\, \|\boldsymbol{U}\| \max\{\overline{A}, \sqrt{\beta \overline{C}}\}, \quad a_5 = \frac{2\overline{A}^2(M+1)}{\beta M}, \quad a_6 = \frac{2(1+\lambda)}{M}.$$

With the above shorthand definitions, the characteristic polynomial for $\boldsymbol{Q}$, denoted by $g\colon \mathbb{R} \to \mathbb{R}$, is given by

$$g(\sigma) = \det \begin{pmatrix} \sigma - (1 - \gamma\alpha + \gamma^2 a_1) & -\gamma a_2 - \gamma^2 a_3 & 0 \\ 0 & \sigma - \lambda & -\gamma \\ -\gamma a_4 & -\gamma a_5 - a_6 & \sigma - \left(\lambda + \frac{\gamma\rho}{M}\right) \end{pmatrix}.$$

By direct computation, we have

$$g(\sigma) = (\sigma - (1 - \gamma\alpha + \gamma^2 a_1))\, g_0(\sigma) - \gamma^3(a_2 + \gamma a_3)a_4 \tag{60}$$

where

$$g_0(\sigma) := (\sigma - \lambda)^2 - \frac{\gamma\rho}{M}(\sigma - \lambda) - \gamma(\gamma a_5 + a_6) \;. \tag{61}$$

Notice that the two roots of the above polynomial can be upper bounded by:

$$\lambda + \frac{\gamma\rho}{2M} \pm \sqrt{\left(\frac{\gamma\rho}{2M}\right)^2 + \gamma(\gamma a_5 + a_6)} \le \overline{\sigma} := \lambda + \frac{\gamma\rho}{M} + \sqrt{\gamma(\gamma a_5 + a_6)} \tag{62}$$

In particular, for all $\sigma \ge \overline{\sigma}$, we have

$$g_0(\sigma) \ge (\sigma - \overline{\sigma})^2 \;. \tag{63}$$

Now, let us define

$$\sigma^\star := \max\left\{ \frac{\gamma\alpha}{4} + 1 - \gamma\alpha + \gamma^2 a_1, \overline{\sigma} + \gamma\sqrt{\frac{4(a_2+\gamma a_3)a_4}{\alpha}} \right\} \tag{64}$$

Observe that for all $\sigma \geq \sigma^\star$, it holds that

$$\begin{aligned} g(\sigma) &\geq (\sigma - (1-\gamma\alpha+\gamma^2 a_1))(\sigma-\overline{\sigma})^2 - \gamma^3(a_2+\gamma a_3)a_4 \\ &\geq \frac{\gamma\alpha}{4}\gamma^2\frac{4(a_2+\gamma a_3)a_4}{\alpha} - \gamma^3(a_2+\gamma a_3)a_4 = 0 \,. \end{aligned} \tag{65}$$

Lastly, observe that $g(\sigma)$ is strictly increasing for all $\sigma \geq \sigma^\star$. Combining with the Perron Frobenius theorem shows that $\rho(\boldsymbol{Q}) \leq \sigma^\star$. Moreover, as $\lambda < 1$ and $\alpha > 0$, there exists a sufficiently small γ such that $\sigma^\star < 1$. We conclude that $\rho(\boldsymbol{Q}) < 1$ in the latter case.

Asymptotic Rate when $M, N \gg 1$ We first evaluate a sufficient condition on γ for the proposed algorithm to converge, *i.e.,* when $\sigma^\star < 1$. Let us consider (64) and the first operand in the $\max\{\cdot\}$. The first operand is guaranteed to be less than one if:

$$\gamma \leq \frac{\alpha}{2a_1} \Longrightarrow \frac{\gamma\alpha}{4} + 1 - \gamma\alpha + \gamma^2 a_1 \leq 1 - \frac{\gamma\alpha}{4} \,. \tag{66}$$

Moreover, from the definition of a_1, we note that this requires $\gamma = \mathcal{O}(1/M^2)$ if $M \gg 1$.

Next, we notice that for geometric graphs, we have $\lambda = 1 - c/N$ for some positive c. Substituting this into the second operand in (64) gives

$$1 - \frac{c}{N} + \frac{\gamma\rho}{M} + \sqrt{\gamma(\gamma a_5 + a_6)} + \gamma\sqrt{\frac{4(a_2+\gamma a_3)a_4}{\alpha}} < 1 \,. \tag{67}$$

Therefore, (66) and (67) together give a sufficient condition for $\sigma^\star < 1$.

To obtain an asymptotic rate when $N, M \gg 1$. Observe that $a_2 = \Theta(\sqrt{N})$, $a_3 = \Theta(\sqrt{N}M)$, $a_4 = \Theta(\sqrt{N})$ $a_5 = \Theta(1)$, $a_6 = \Theta(1/M)$. Moreover, the condition (66) gives $\gamma = \mathcal{O}(1/M^2)$, therefore Eq. (67) can be approximated by

$$1 - \frac{c}{N} + \gamma\,\Theta\left(\sqrt{N}\right) + \sqrt{\gamma}\,\Theta(1/M) \tag{68}$$

Setting the above to $1 - c/(2N)$ requires one to have $\gamma = \mathcal{O}(1/N^2)$.

Finally, the above discussions show that setting $\gamma = \mathcal{O}(1/\max\{N^2, M^2\})$ guarantees that $\sigma^\star < 1$. In particular, we have $\sigma^\star \leq \max\{1 - \gamma\frac{\alpha}{4}, 1 - c/(2N)\} = 1 - \mathcal{O}(1/\max\{N^2, M^2\})$ **Q.E.D.**

A.2 Derivation of Equation (29)

We we establish (29) with details. Recall that $\underline{\boldsymbol{h}}(t)$ and $\underline{\boldsymbol{v}}(t)$ are defined in (28). We verify this equation for each block of $\underline{\boldsymbol{h}}(t)$. To begin with, for the first block, for $\boldsymbol{h}_{\boldsymbol{\theta}}(t)$ defined in (26), we have

$$\boldsymbol{h}_{\boldsymbol{\theta}}(t) = \rho\bar{\boldsymbol{\theta}}(t) + \frac{1}{N}\sum_{i=1}^N \hat{\boldsymbol{A}}^\top \boldsymbol{w}_i^t = \rho\big(\bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star + \boldsymbol{\theta}^\star\big) + \frac{1}{N}\sum_{i=1}^N \hat{\boldsymbol{A}}^\top \boldsymbol{w}_i^t \,.$$

Recall from (24) that $\rho\boldsymbol{\theta}^\star = -\frac{1}{N}\sum_{i=1}^N \hat{\boldsymbol{A}}^\top \boldsymbol{w}_i^\star$, which implies that

$$\boldsymbol{h}_{\boldsymbol{\theta}}(t) = \rho\big(\bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star\big) + \sum_{i=1}^N \sqrt{\frac{\beta}{N}}\hat{\boldsymbol{A}}^\top \frac{1}{\sqrt{\beta N}}\big(\boldsymbol{w}_i^t - \boldsymbol{w}_i^\star\big) = [\boldsymbol{G}\underline{\boldsymbol{v}}(t)]_1 \,, \tag{69}$$

where $[\boldsymbol{G}\underline{\boldsymbol{v}}(t)]_1$ denotes the first block of $\boldsymbol{G}\underline{\boldsymbol{v}}(t)$.

It remains to establish the equation for the remaining blocks. For any $i \in \{1, \ldots, N\}$, let us focus on the $i+1$-th block. By the definition of $\boldsymbol{h}_{\boldsymbol{w}_i}(t)$ in (27), we have

$$-\sqrt{\frac{\beta}{N}}\boldsymbol{h}_{\boldsymbol{w}_i}(t) = -\sqrt{\frac{\beta}{N}}\big(\hat{\boldsymbol{A}}\bar{\boldsymbol{\theta}}(t) - \hat{\boldsymbol{C}}\boldsymbol{w}_i^t - \hat{\boldsymbol{b}}_i\big) = -\sqrt{\frac{\beta}{N}}\big(\hat{\boldsymbol{A}}(\bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star) + \hat{\boldsymbol{A}}\boldsymbol{\theta}^\star - \hat{\boldsymbol{C}}\boldsymbol{w}_i^t - \hat{\boldsymbol{b}}_i\big) \ .$$

Again from (24), it holds that $\hat{\boldsymbol{A}}\boldsymbol{\theta}^\star = \boldsymbol{b}_i + \hat{\boldsymbol{C}}\boldsymbol{w}_i^\star$. Therefore,

$$-\sqrt{\frac{\beta}{N}}\big(\hat{\boldsymbol{A}}\bar{\boldsymbol{\theta}}(t) - \hat{\boldsymbol{C}}\boldsymbol{w}_i^t - \hat{\boldsymbol{b}}_i\big) = -\sqrt{\frac{\beta}{N}}\hat{\boldsymbol{A}}(\bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}^\star) + \beta\hat{\boldsymbol{C}}\frac{\boldsymbol{w}_i^t - \boldsymbol{w}_i^\star}{\sqrt{\beta N}} = [\boldsymbol{G}\underline{\boldsymbol{v}}(t)]_{i+1} \ , \tag{70}$$

where $[\boldsymbol{G}\underline{\boldsymbol{v}}(t)]_{i+1}$ denotes the $i+1$-th block of $\boldsymbol{G}\underline{\boldsymbol{v}}(t)$. Combining (69) and (70) gives the desired equality.

B Additional Experiments

An interesting observation from Theorem 1 is that the convergence rate of PD-DistIAG depends on M and the topology of the graph. The following experiments will demonstrate the effects of these on the algorithm, along with the effects of regularization parameter ρ.

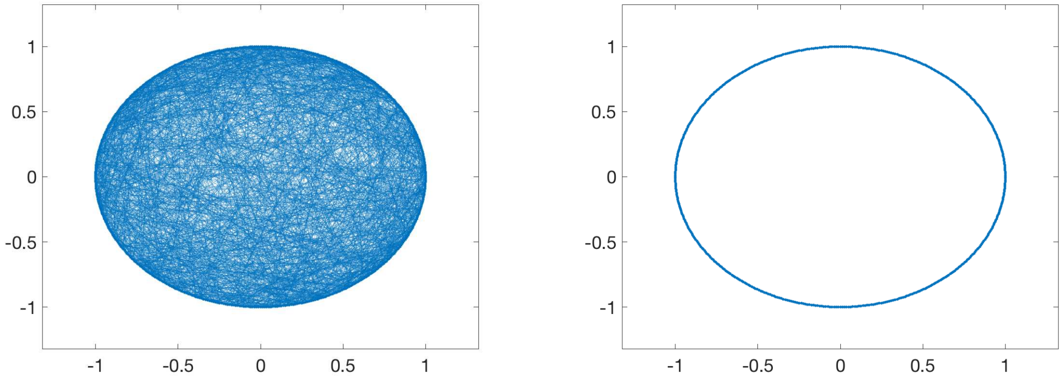

Figure 2: Illustrating the graph topologies used in the additional experiments. (Left) ER graph with connectivity probability of $1.01 \log N/N$. (Right) Ring graph.

To demonstrate the dependence of PD-DistIAG on the graph topology, we fix the number of agents at $N = 500$ and compare the performances on the ring and the ER graph set with probability of connection of $p = 1.01 \log N/N$, as illustrated in Fig. 2. Notice that the ring graph is not a geometric graph and its connectivity parameter, defined as $\lambda := \|\boldsymbol{W} - (1/N)\mathbf{1}\mathbf{1}^\top\|_{1,\infty}$ from the previous section can be much closer to 1 than the ER graph. Therefore, we expect the PD-DistIAG algorithm to converge slower on the ring graph. This is corroborated by Fig. 3. Furthermore, from the figure, we observe that with a larger regularization ρ, the disadvantage for using the ring graph has exacerbated. We suspect that this is due to the fact that the convergence speed is limited by the graph connectivity, as seen in (55); while in the case of ER graph, the algorithm is able to exploit the improved problem's condition number.

Next, we consider the same set of experiment but increase the number of samples to $M = 5000$.

Interestingly, for this example, the performances of the ring graph and the ER graph settings are almost identical in this setting with large sample size M. This is possible as we recall from Theorem 1 that the algorithm converges at a rate of $\mathcal{O}(\sigma^t)$ where $\sigma = 1 - \mathcal{O}(1/\max\{MN^2, M^3\})$. As we have $M \gg N$, the impact from the sample size M becomes dominant, and is thus insensitive to the graph's connectivity.

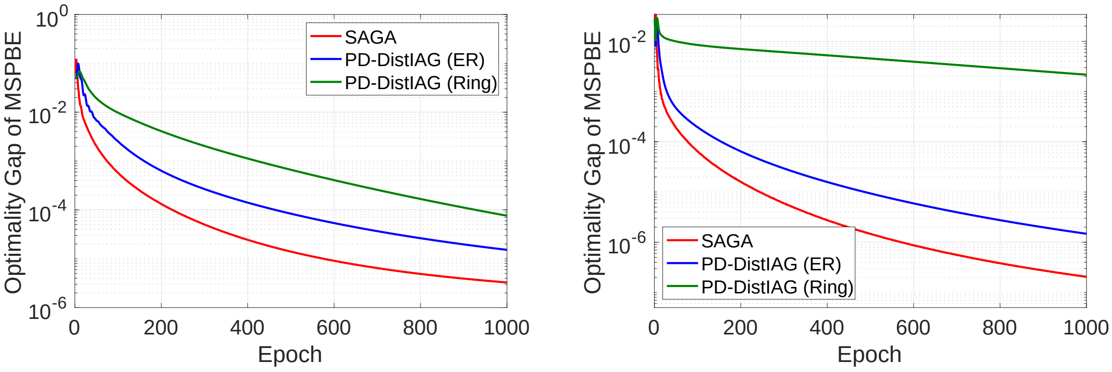

Figure 3: Experiment with `mountaincar` dataset. For this problem, we only have $d = 300$, $M = 500$ samples, but yet there are $N = 500$ agents. (Left) We set $\rho = 0.01$. (Right) We set $\rho = 0.1$.

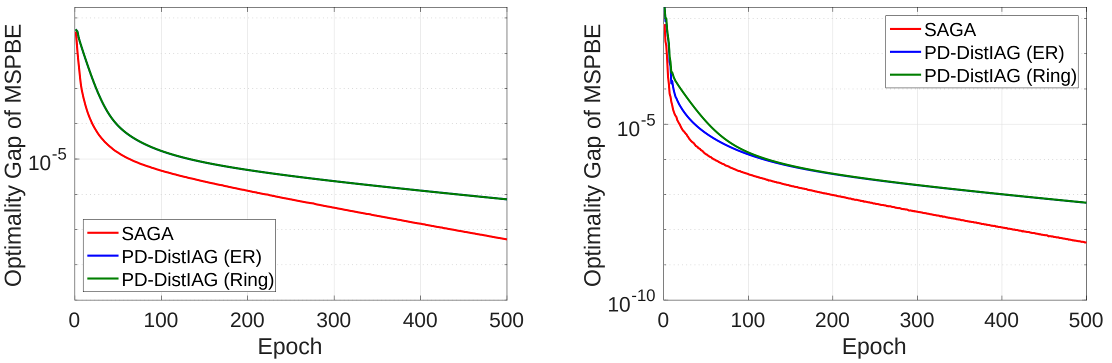

Figure 4: Experiment with `mountaincar` dataset. For this problem, we only have $d = 300$, $M = 500$ samples, but yet there are $N = 500$ agents. (Left) We set $\rho = 0.01$. (Right) We set $\rho = 0.1$.

References

Arslan, G. and Yüksel, S. (2017). Decentralized Q-learning for stochastic teams and games. *IEEE Transactions on Automatic Control*, **62** 1545–1558.

Borkar, V. S. (2008). *Stochastic approximation: A dynamical systems viewpoint*. Cambridge University Press.

Callaway, D. S. and Hiskens, I. A. (2011). Achieving controllability of electric loads. *Proceedings of the IEEE*, **99** 184–199.

Chambolle, A. and Pock, T. (2016). On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, **159** 253–287.

Chaudhuri, K., Monteleoni, C. and Sarwate, A. D. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, **12** 1069–1109.

Chen, J. and Sayed, A. H. (2012). Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Transactions on Signal Processing*, **60** 4289–4305.

Chen, Y. and Wang, M. (2016). Stochastic primal-dual methods and sample complexity of reinforcement learning. *arXiv preprint arXiv:1612.02516*.

Corke, P., Peterson, R. and Rus, D. (2005). Networked robots: Flying robot navigation using a sensor net. *Robotics Research* 234–243.

Cortes, J., Martinez, S., Karatas, T. and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, **20** 243–255.

Dai, B., He, N., Pan, Y., Boots, B. and Song, L. (2016). Learning from conditional distributions via dual embeddings. *arXiv preprint arXiv:1607.04579*.

Dai, B., Shaw, A., He, N., Li, L. and Song, L. (2017a). Boosting the actor with dual critic. *arXiv preprint arXiv:1712.10282*.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Chen, J. and Song, L. (2017b). Smoothed dual embedding control. *arXiv preprint arXiv:1712.10285*.

Dall'Anese, E., Zhu, H. and Giannakis, G. B. (2013). Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid*, **4** 1464–1475.

Defazio, A., Bach, F. and Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*.

Du, S. S., Chen, J., Li, L., Xiao, L. and Zhou, D. (2017). Stochastic variance reduction methods for policy evaluation. *arXiv preprint arXiv:1702.07944*.

Feyzmahdavian, H. R., Aytekin, A. and Johansson, M. (2014). A delayed proximal gradient method with linear convergence rate. In *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*. IEEE.

Foerster, J., Assael, Y. M., de Freitas, N. and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*.

Foerster, J., Nardelli, N., Farquhar, G., Torr, P., Kohli, P., Whiteson, S. et al. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1702.08887*.

Gupta, J. K., Egorov, M. and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multi-agent Systems*.

Gurbuzbalaban, M., Ozdaglar, A. and Parrilo, P. A. (2017). On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, **27** 1035–1048.

Hu, J. and Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, **4** 1039–1069.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*.

Kober, J. and Peters, J. (2012). Reinforcement learning in robotics: A survey. In *Reinforcement Learning*. Springer, 579–610.

Lauer, M. and Riedmiller, M. (2000). An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *International Conference on Machine Learning*.

Lee, D., Yoon, H. and Hovakimyan, N. (2018). Primal-dual algorithm for distributed reinforcement learning: Distributed gtd2. *arXiv preprint arXiv:1803.08031*.

Lian, X., Wang, M. and Liu, J. (2016). Finite-sum composition optimization via variance reduced gradient descent. *arXiv preprint arXiv:1610.04674*.

Lin, A. and Ling, Q. (2014). Decentralized and privacy-preserving low-rank matrix completion. Preprint.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*.

Littman, M. L. (2001). Value-function reinforcement learning in Markov games. *Cognitive Systems Research*, **2** 55–66.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S. and Petrik, M. (2015). Finite-sample analysis of proximal gradient td algorithms. In *UAI*.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P. and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*.

Macua, S. V., Chen, J., Zazo, S. and Sayed, A. H. (2015). Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, **60** 1260–1274.

Macua, S. V., Tukiainen, A., Hernández, D. G.-O., Baldazo, D., de Cote, E. M. and Zazo, S. (2017). Diff-dac: Distributed actor-critic for multitask deep reinforcement learning. *arXiv preprint arXiv:1710.10363*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, **518** 529.

Nedić, A. and Bertsekas, D. P. (2003). Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, **13** 79–110.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, **54** 48–61.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P. and Vian, J. (2017). Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*.

Palaniappan, B. and Bach, F. (2016). Stochastic variance reduction methods for saddle-point problems. In *Advances in Neural Information Processing Systems*.

Parisotto, E., Ba, J. L. and Salakhutdinov, R. (2015). Actor-mimic: Deep multi-task and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*.

Pu, S. and Nedić, A. (2018). Distributed stochastic gradient tracking methods. *arXiv preprint arXiv:1805.11454*.

Qu, G. and Li, N. (2017). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*.

Rabbat, M. and Nowak, R. (2004). Distributed optimization in sensor networks. In *International Symposium on Information Processing in Sensor Networks*.

Schmidt, M., Le Roux, N. and Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, **162** 83–112.

Shi, W., Ling, Q., Wu, G. and Yin, W. (2015). Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, **25** 944–966.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. et al. (2017). Mastering the game of Go without human knowledge. *Nature*, **550** 354–359.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge.

Sutton, R. S., Maei, H. R. and Szepesvári, C. (2009). A convergent $o(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In *Advances in neural information processing systems*.

Teh, Y. W., Bapst, V., Czarnecki, W. M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N. and Pascanu, R. (2017). Distral: Robust multi-task reinforcement learning. *arXiv preprint arXiv:1707.04175*.

Tsitsiklis, J., Bertsekas, D. and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, **31** 803–812.

Wang, M. (2017). Primal-dual π learning: Sample complexity and sublinear run time for ergodic markov decision problems. *arXiv preprint arXiv:1710.06100*.

Wang, X. and Sandholm, T. (2003). Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Advances in Neural Information Processing Systems*.

Wilson, A., Fern, A., Ray, S. and Tadepalli, P. (2007). Multi-task reinforcement learning: A hierarchical Bayesian approach. In *International Conference on Machine Learning*.

Zhang, K., Yang, Z., Liu, H., Zhang, T. and Başar, T. (2018). Fully decentralized multi-agent reinforcement learning with networked agents. *arXiv preprint arXiv:1802.08757*.

Zhu, M. and Martínez, S. (2010). Discrete-time dynamic average consensus. *Automatica*, **46** 322–329.