Byzantine-Resilient Decentralized Parallel Policy Gradient

Anonymous

Abstract—**Parallel reinforcement learning (RL) is an important approach to dealing with the challenge of data inefficiency in RL. However, the existing distributed parallel RL framework requires a central server to collect messages from multiple agents for cooperation, and suffers from the communication bottleneck. On the other hand, malicious agents can send random or well-designed messages for the sake of hindering or destroying the learning process (termed as Byzantine attacks). To address these issues, we first develop a decentralized parallel RL algorithm, named as decentralized parallel policy gradient (DP-PG), where the agents exchange learning parameters through a peer-to-peer network, avoiding the communication bottleneck caused by the central server. Then, we propose Byzantine-resilient decentralized parallel policy gradient (BRDP-PG) that replaces the vulnerable weighted mean aggregation in DP-PG with the robust coordinate trimmed mean (CTM) aggregation. We analyze the two proposed algorithms and provide their theoretical guarantees. Finally, we conduct numerical experiments to reveal the effectiveness of DP-PG and BRDP-PG.**

Index Terms—**Reinforcement learning, decentralized parallel policy gradient, Byzantine-resilience.**

I. Introduction

Reinforcement learning (RL) aims at training one or multiple agents in an environment to make decisions via trial and error. As a promising control technology, RL has been successfully applied in various fields, such as games [1], smart grids [2], [3], autonomous driving [4], [5], sensor networks [6], [7], robotics [8], to name a few.

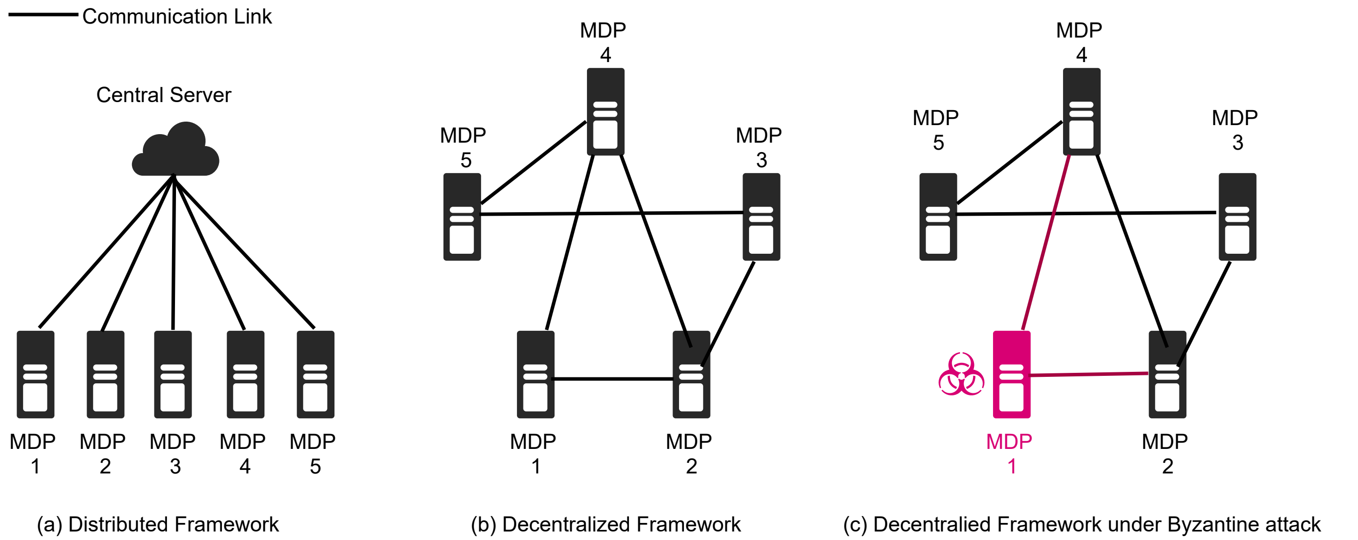
Data efficiency is an essential issue in RL, since RL requires a large number of interactions with the environment to obtain sufficient trajectories for training a satisfactory policy. Such interactions can be described as a Markov decision process (MDP). The issue of data efficiency is particularly important when the state-action space is large, and becomes more critical with the emergence of deep RL. A remedy to address this issue is to interact with multiple duplicated instances of the environment and obtain multiple trajectories simultaneously, as known as parallel RL [9]–[14]. In addition to help train a better policy, parallel RL is also beneficial for stabilizing the training process [15]. An example of parallel RL is that multiple autonomous vehicles run across a city to obtain sufficient data for training a satisfactory policy in a short time. Note that parallel RL is different from another class of multi-agent RL that is often abbreviated as MARL. In MARL, multiple agents are placed in a *common environment* and cooperate to train a *joint policy*, while the actions of each agent can affect the transitions of the common environment. In contrast, in parallel RL, each agent is placed in an *instance of the environment* and the agents cooperate to train a *consensual policy*, while the actions of each agent are only able to affect the transitions of its own instance of the environment.

Traditionally, the architecture of parallel RL is distributed [9], [10], [16]; see Fig. 1(a). Distributed parallel RL relies on a central server to collect messages from the agents to enable cooperation, leading to the communication bottleneck [17]–[19]. Motivated by the advances in decentralized control, optimization and learning, we consider the decentralized architecture of parallel RL, as illustrated in Fig. 1(b). In decentralized parallel RL, each agent only need to exchange messages with its neighbors instead of a central server, which effectively avoids the communication bottleneck. Although decentralized algorithms have been extensively studied in MARL [20], [21], decentralized parallel RL remains an untouched territory.

Similar to its distributed counterpart, decentralized parallel RL also inevitably faces the threats of Byzantine attacks. For example, some of the autonomous vehicles could be hacked by intruders or encounter sensor failures, and send wrong messages to their neighbors; see Fig. 1(c). We characterize such abnormalities with the worst-case Byzantine attack model: The abnormal agents (also called as Byzantine agents) send arbitrarily wrong messages during the communication stages to hinder or even destroy the learning process. The messages transmitted by the Byzantine agents can be random or well-designed. In the context of distributed and decentralized learning, Byzantine-resilient algorithms often replace the vulnerable mean or weighted mean aggregation rule, which aggregates the received messages, with robust ones including coordinate-wise median [22], geometric median [23], coordinate-wise trimmed mean (CTM) [24], etc. Byzantine-robust aggregation has also been successfully applied to Byzantine-resilient decentralized MARL [25]. However, its effectiveness in decentralized parallel RL is still unknown.

In this paper, we first consider the Byzantine-free case, and propose a policy gradient algorithm for decentralized parallel RL, abbreviated as DP-PG. Therein, each agent maintains its local policy parameters. At each iteration, each agent performs a local parameter update step using a policy gradient algorithm (for example, gradient of a partially observable MDP that is abbreviated as GPOMDP [26]), and then sends the local policy parameters to its neighbors. Upon receiving all neighboring messages, each agent aggregates them with weighted mean. We also establish the convergence guarantee for DP-PG.

Second, we investigate the Byzantine case, and develop a Byzantine-resilient variant of DP-PG, called as BRDP-PG. BRDP-PG replaces the vulnerable weighted mean aggregation by CTM, and thus mitigates the impact of Byzantine attacks. For each coordinate, each agent sorts the received neighboring



(a) Distributed Framework (b) Decentralized Framework (c) Decentralied Framework under Byzantine attack

Fig. 1. The overview of parallel reinforcement learning. (a) Agents exchange messages with a central server. (b) Agents exchange messages with each other over a peer-to-peer network. (c) Agent 1 in the peer-to-peer network is Byzantine and sends malicious messages to its neighbors.

messages, discards several largest ones and several smallest ones, and then averages the rest. We prove the convergence of BRDP-PG, and theoretically provide the topology-dependent asymptotic learning error.

The remainder of this paper is organized as follows. Section II surveys the related work. Section III proposes the DP-PG algorithm and establishes the convergence. Section IV proposes the BRDP-PG algorithm and analyzes the convergence and asymptotic learning error. Section V conducts numerical experiments in two standard environments and demonstrates the effectiveness of the proposed algorithms. Section VI concludes this paper.

II. Related Works

Our work belongs to the line of research in parallel RL. The core idea of parallel RL is to employ parallelism to scale up the ability of handling the large state-action space in RL. The work of [11] proposes a parallel RL framework for learning and planning to improve the sample efficiency of temporal difference (TD) control. In [27], an asynchronous parallel SARSA(λ) algorithm is proposed to accelerate convergence towards the optimal policy. Therein, each agent independently learns in a sperate environment using a linear function approximation, and exchanges information with a central server in an asynchronous manner. The work of [10] applies MapReduce to parallelize RL algorithms, including both tabular and linear function approximation settings. In the general RL architecture (Gorila) [16], where some agents (actors) interact with multiple instances of the same environment to generate training samples, which are then stored in a global replay memory via a distributed database. Some other agents (learners) retrieve the training samples to calculate gradients and send them to a central server. The central server updates the learning parameter and coordinates with the actors and learners. The work of [9] aims at parallel on-policy and off-policy RL, where the agents execute on multiple instances of the environment, and asynchronously update the global shared learning parameter. A distributed policy iteration framework is proposed in [28] for finding a class of non-convex functionals.

Though the above-mentioned works have achieved remarkable improvements in terms of data efficiency, all of them are implemented in a distributed network and suffer from the communication bottleneck caused by the central server [17]. For parallel RL over a decentralized network, a novel asynchronous push-pull incremental aggregated gradient (APP-IAG) method is proposed in [29] to solve the policy evaluation problem. In contrast, we study the policy optimization problem in the context of decentralized parallel RL.

Another line of research closely related to our work is multi-agent RL (MARL). Different from parallel RL, in MARL, the agents are deployed in a common environment. For example, the work of [30] investigates the multi-agent cooperation problem with MARL. In [31], a fully decentralized actor-critic algorithm is proposed. The work of [32] devises a distributed TD(0) method for policy evaluation. In contrast, parallel RL can be intrinsically viewed as an effective way to improve the data efficiency in single-agent RL.

Our work is also related to Byzantine-resilient distributed or decentralized learning. In the distributed scenario, various robust aggregation rules have been developed to handle Byzantine attacks, such as coordinate-wise median [22], geometric median [23], coordinate trimmed mean (CTM) [24], robust stochastic model aggregation (RSA) [33], centered clipping (CC) [34], etc.

In the decentralized scenario, the work of [35] combines CTM and the projection operator to limit the influence of Byzantine attacks. A theoretical guarantee of the learning error is provided when the learning parameter is a scalar. In [36], a decentralized variant of RSA is developed for both static and time-varying networks, by adding a total variation norm-penalized term in the cost function to force the local learning parameters of the honest agents to be sufficiently close. Self-centered clipping (SCCLIP) is a decentralized version of CC, by clipping the received messages prior to weighted mean aggregation [37]. In [38], CTM and comparative gradient elimination (CGE) are performed on learning parameters and gradients, respectively, followed by a projection operator. The uniform Byzantine-resilient aggregation rule (UBAR) is a two-stage method [39]. At the first stage, each honest agent selects a candidate pool of potential honest neighboring agents based on the Euclidean distances. At the second stage, each honest agents picks some neighboring agents from the candidate pool in terms of the local cost values. In [40], a similarity-based reweighting scheme is proposed. The idea is to ensure that the aggregated messages have similar directions and magnitudes. The work of [41] provides a systematic framework for designing robust aggregation rules, and proposes iterative outlier scissor (IOS), which iteratively discards outliers and then performs weighted average aggregation on the rest.

To the best of our knowledge, there is only one work that considers Byzantine-resilience in parallel RL, but for a distributed network. Federated policy gradient with Byzantine-resilience (FedPG-BR) designs a filtering method, where outliers are constrained in a small ball centered by the true aggregated learning parameter [12]. In contrast, we consider Byzantine-resilient decentralized parallel RL in this paper.

III. Decentralized Parallel Reinforcement Learning

A. Problem Statement

Suppose that N agents are deployed in N instances of the same environment, respectively. The agents are bidirectionally connected and form a decentralized communication network $\mathcal{G} := \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{N} := \{1, \dots, N\}$ is the set of agents and $\mathcal{E}$ is the set of edges. Each pair $(i, j) \in \mathcal{E}$ means that agents i and j are neighbors, and can exchange messages with each other. For agent i, the set of its neighbors is defined as $\mathcal{N}_i$.

Each agent independently interacts with its corresponding instance to generate trajectories. Because all the instances are associated with the same environment, such interactions can be characterized by a unique MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma \rangle$, where $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ respectively stand for the state, action and reward spaces, $\mathcal{P}$ is the probability function, and $\gamma \in [0, 1]$ is the discount factor. Given a state $s_i \in \mathcal{S}$, agent i selects an action $a_i \in \mathcal{A}$ according to its policy $\pi_{\boldsymbol{\theta}_i}(a_i|s_i) : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that is parameterized by $\boldsymbol{\theta}_i \in \mathbb{R}^D$ with $D \ll |\mathcal{S} \times \mathcal{A}|$. Then, the instance transits to the next state $s'_i \in \mathcal{S}$ with probability $p(s'_i|s_i, a_i) \in \mathcal{P}$, and an immediate reward $r_i \in \mathcal{R}$ is given as the feedback. Taking H consecutive actions yields a trajectory $\tau_i := (s_i^0, a_i^0, r_i^0, s_i^1, a_i^1, r_i^1, \dots, s_i^{H-1}, a_i^{H-1}, r_i^{H-1})$. The long-term return of this trajectory is $\mathcal{R}(\tau_i) := \sum_{h=0}^{H-1} (\gamma)^h r_i^h$, and the expected long-term return of the policy $\pi_{\boldsymbol{\theta}_i}$ is given by $J_i(\boldsymbol{\theta}_i) := \mathbb{E}_{s_i^0 \sim \rho_i(s_i^0), \tau_i \sim \pi_{\boldsymbol{\theta}_i}} [\mathcal{R}(\tau_i)]$, where $\rho_i(s_i^0)$ is the distribution of the initial state s_i^0 of agent i. Note that the distributions are different across the agents in general. The goal of the agents is to collaboratively find a common optimal policy $\pi_{\boldsymbol{\theta}^*}$, parameterized by a common $\boldsymbol{\theta}^*$, to maximize the average expected long-term return of the corresponding MDP, given by

$$\boldsymbol{\theta}^* \in \arg\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^{N} J_i(\boldsymbol{\theta}). \tag{1}$$

B. Decentralized Parallel Policy Gradient (DP-PG)

We propose a decentralized parallel policy gradient (DP-PG) algorithm to solve (1). At each iteration, DP-PG has two steps: local parameter update and neighboring message aggregation. Consider agent i with the parameter $\boldsymbol{\theta}_i^t$ at iteration t. The ideal policy gradient is given by

$$\nabla J_i(\boldsymbol{\theta}_i^t) = \mathbb{E}_{\tau_i \sim p(\tau_i|\boldsymbol{\theta}_i^t)} [\nabla \log p(\tau_i|\boldsymbol{\theta}_i^t) \mathcal{R}(\tau_i)], \tag{2}$$

where

$$p(\tau_i|\boldsymbol{\theta}_i^t) := \rho_i(s_i^0) \pi_{\boldsymbol{\theta}_i^t}(a_i^0|s_i^0) \prod_{h=1}^{H-1} p(s_i^h|s_i^{h-1}, a_i^{h-1}) \pi_{\boldsymbol{\theta}_i^t}(a_i^h|s_i^h).$$

Ascent along the ideal policy gradient direction shall yield a favorable local parameter update.

Nevertheless, calculating $\nabla J_i(\boldsymbol{\theta}_i^t)$ requires all possible trajectories given the parameter $\boldsymbol{\theta}_i^t$, and is impossible when the environment is complicated and/or the state-action space is large. This fact motivates the stochastic policy gradient ascent method. Instead of calculating the ideal policy gradient in (2), at iteration t, agent i samples one trajectory τ_i^t (or a batch of trajectories), calculates a stochastic policy gradient $\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)$, and ascends to update

$$\boldsymbol{\theta}_i^{t+\frac{1}{2}} = \boldsymbol{\theta}_i^t + \alpha^t \cdot \nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t), \tag{3}$$

where $\alpha^t > 0$ is the step size.

The stochastic policy gradient $\nabla J(\tau_i^t|\boldsymbol{\theta}_i^t)$ is an estimator of the ideal policy gradient $\nabla J_i(\boldsymbol{\theta}_i^t)$ in (2). In this paper, we consider the GPOMDP estimator [26], given by

$$\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t) = \sum_{h=0}^{H-1} \left(\sum_{k=0}^{h} \nabla \log \pi_{\boldsymbol{\theta}_i^t}(a_i^k|s_i^k) \right) \left((\gamma)^h r_i^h - b_i^h \right), \tag{4}$$

where b_i^h is a baseline constant to reduce variance.

After the local parameter update step, each agent i sends $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ to its neighbors. Then, it aggregates all received neighboring messages and its own local parameter, as

$$\boldsymbol{\theta}_i^{t+1} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} \boldsymbol{\theta}_j^{t+\frac{1}{2}}. \tag{5}$$

where w_{ij} is (i, j)-th entry a weight matrix $W \in \mathbb{R}^{N \times N}$. The weight matrix W is doubly stochastic, and if and only if two agents are neither neighbors nor identical, its corresponding entry is 0 [42].

The updates of DP-PG are outlined in Algorithm 1.

Algorithm 1 Decentralized Parallel Policy Gradient (DP-PG)

```
Initialize: Parameter vector $\boldsymbol{\theta}_i^0 = \boldsymbol{\theta}^0, \forall i \in \mathcal{N}$
for $t = 0, 1, \ldots, T$ do
    for agents $i = 1, \ldots, N$ in parallel do
        Calculate $\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)$ according to (4).
        Update $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ according to (3).
        Send $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ to all neighbors.
        Update $\boldsymbol{\theta}_i^{t+1}$ according to (5).
    end for
end for
```

C. Convergence Analysis

This section analyzes the convergence of DP-PG. We denote the set of trajectories at iteration t as $\boldsymbol{\tau}^t = \{\tau_1^t, \tau_2^t, \ldots, \tau_N^t\}$. For the ease of subsequent theoretical analysis, we express the learning parameters in a compact form as

$$\Theta^t = [\boldsymbol{\theta}_1^t, \ldots, \boldsymbol{\theta}_N^t]^\top \in \mathbb{R}^{N \times D}. \tag{6}$$

We define the average learning parameter as $\bar{\boldsymbol{\theta}}^t = \frac{1}{N}\sum_{i=1}^N \boldsymbol{\theta}_i^t$.

Assumption 1: The immediate rewards r_i^h and the baseline constants b_i^h are bounded by positive constants $r_{\max}$ and $b_{\max}$, respectively. That is, $|r_i^h| \leq r_{\max}$ and $|b_i^h| \leq b_{\max}$ for any agent $i = 1, \ldots, N$ and $h = 0, \ldots, H-1$.

Assumption 2: For any $a_i \in \mathcal{A}$ and $s_i \in \mathcal{S}$, the log-density of the policy function $\pi_{\boldsymbol{\theta}_i}$ of any agent $i = 1, \ldots, N$ satisfies

$$||\nabla \log \pi_{\boldsymbol{\theta}_i}(a_i|s_i)|| \leq G, \tag{7}$$

$$\left|\frac{\partial^2}{\partial\theta_{i,n}\partial\theta_{i,m}} \log \pi_{\boldsymbol{\theta}_i}(a_i|s_i)\right| \leq M, \quad \forall n, m = 1, \cdots, D, \tag{8}$$

where G and M are positive constants, while $\theta_{i,n}$ and $\theta_{i,m}$ denote the n-th and m-th entries of $\boldsymbol{\theta}_i$, respectively.

Assumption 1 is standard and adopted by a number of papers, for example, [12], [17]. Assumption 2 implies bounded policy gradients and bounded partial derivatives, and holds true given a wide range of stochastic policies [17], [43].

Proposition 1: Under Assumptions 1 and 2, for any agent $i = 1, \ldots, N$ and $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_i' \in \mathbb{R}^D$, given the GPOMDP gradient estimator, it holds that

$$||\nabla J_i(\tau_i|\boldsymbol{\theta}_i)|| \leq C_g, \tag{9}$$

$$||\nabla J_i(\tau_i|\boldsymbol{\theta}_i) - \nabla J_i(\tau_i|\boldsymbol{\theta}_i')|| \leq L_g||\boldsymbol{\theta}_i - \boldsymbol{\theta}_i'||, \tag{10}$$

where $C_g = HG(r_{\max}+b_{\max})/(1-\gamma)$ and $L_g = HM(r_{\max}+b_{\max})/(1-\gamma)$.

The proof of Proposition 1 is similar to those in [12], [17], [43], and we leave it to Appendix D.

Proposition 2: Under Assumptions 1 and 2, for any agent $i = 1, \ldots, N$ and $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_i' \in \mathbb{R}^D$, it holds that

$$||\nabla J_i(\boldsymbol{\theta}_i) - \nabla J_i(\boldsymbol{\theta}_i')|| \leq L||\boldsymbol{\theta}_i - \boldsymbol{\theta}_i'||, \tag{11}$$

where $L = (H^2G^2 + HM)r_{\max})/(1-\gamma)$.

The proof of Proposition 2 is similar to that in [17], and we leave it to Appendix E.

Assumption 3: For any agent $i = 1, \ldots, N$, there exists a nonnegative constant ζ^2 such that

$$\mathbb{E}_{\tau_i}||\nabla J_i(\tau_i|\boldsymbol{\theta}_i) - \nabla J_i(\boldsymbol{\theta}_i)||^2 \leq \zeta^2. \tag{12}$$

for any $\tau_i \sim p(\cdot|\pi_{\boldsymbol{\theta}_i})$.

Assumption 4: For any agent $i = 1, \ldots, N$, there exists a nonnegative constant δ^2 such that

$$||\nabla J_i(\boldsymbol{\theta}) - \nabla J(\boldsymbol{\theta})|| \leq \delta^2. \tag{13}$$

Assumption 5: The stochastic policy gradients $\nabla J_i(\tau_i|\boldsymbol{\theta}_i)$ are independently sampled over iterations $t = 0, 1, \ldots$ and across agents $i \in \mathcal{N}$.

Assumption 6: The weight matrix W is doubly stochastic, namely, $\mathbf{1}^\top W = \mathbf{1}$ and $W\mathbf{1} = \mathbf{1}$. Meanwhile, $w_{ij} > 0$ if and only if agents i and j are neighbors or identical.

Assumption 3 and Assumption 4 are common in decentralized learning, and often referred as bounded inner variation and bounded outer variation, respectively. In the context of RL, Assumption 3 can be found in, for example, [12]. In parallel RL, since all instances refer to the same environment, Assumption 4 is easy to satisfy. Assumption 5 implies independent sampling, and is standard in analyzing stochastic algorithms. Finally, a weight matrix W satisfying Assumption 6 is widely used in decentralized learning algorithms, and can be constructed in various ways [42].

Lemma 1: Define the consensus error at iteration t as

$$\Delta^t = \frac{1}{N}\sum_{i \in \mathcal{N}} ||\boldsymbol{\theta}_i^t - \bar{\boldsymbol{\theta}}^t||. \tag{14}$$

Denote σ as the largest singular value of $W - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. Under Assumptions 1–6, with $\sigma^2 < \frac{1}{3}$, $\alpha^t = \alpha \leq \frac{1}{2\sqrt{3}L}$ and the GPOMDP gradient estimator, the consensus error of Algorithm 1 is bounded as

$$\mathbb{E}\Delta^t \leq \frac{8\alpha^2(\zeta^2+\delta^2)\sigma^2[1-(3\sigma^2)^t]}{1-3\sigma^2}, \tag{15}$$

where the expectation is taken over all random trajectories $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^t$.

Remark: Lemma 1 reveals that the consensus error can be bounded by a constant if t is sufficiently large and $\sigma < \frac{1}{\sqrt{3}}$. If we set $\alpha = \frac{1}{\sqrt{T}}$ with a sufficiently large T, then the consensus error vanish at the rate of $O(\frac{1}{T})$. Meanwhile, the requirement of $\sigma < \frac{1}{\sqrt{3}}$ means that the communication network should be well connected and the weight matrix should be appropriately generated. The proof can be found in Appendix B.

Theorem 1: Define the learning error at iteration T as

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2. \tag{16}$$

Under Assumptions 1–6, with $\alpha^t = \alpha = \frac{1}{\sqrt{T}}$ and $T \geq 12L^2$, the learning error of Algorithm 1 is bounded as

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \leq \frac{2\left[J(\boldsymbol{\theta}^*) - J(\bar{\boldsymbol{\theta}}^0)\right]}{\sqrt{T}} + \frac{2\zeta^2 L}{N\sqrt{T}} + \frac{L_g^2}{T}\sum_{t=1}^T \mathbb{E}\Delta^t, \tag{17}$$

where the expectation is taken over all random trajectories $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^t$.

Remark: The first and second terms at the right-hand side of (17) vanish at the rate of $O(\frac{1}{\sqrt{T}})$ when T increases. The

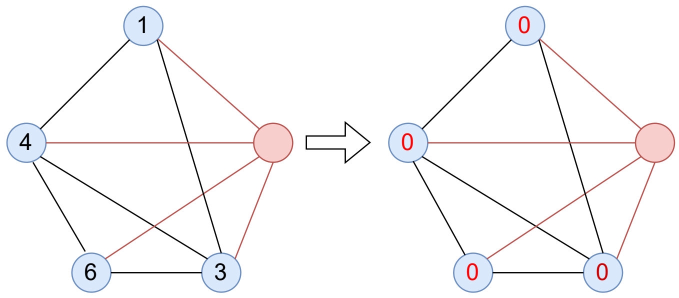


Fig. 2. A scalar example of Byzantine attacks. The red circle represents the Byzantine agent and the numbers denote the parameters held by the honest agents. Left: before attacks; Right: after attacks.

second term is proportional to $\frac{1}{N}$, showing the performance gain through collaborating among the agents. The last term vanishes at the rate of $O(\frac{1}{T})$, as we have discussed below Lemma 1. The proof can be found in Appendix A.

Summarizing Lemma 1 and Theorem 1, we can obtain the following corollary.

Corollary 1: Under the same conditions as those in Theorem 1, with $\sigma^2 < \frac{1}{3}$, the learning error of Algorithm 1 is bounded as

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \le \frac{2\big[J(\boldsymbol{\theta}^*) - J(\bar{\boldsymbol{\theta}}^0)\big]}{\sqrt{T}} + \frac{2\zeta^2 L}{N\sqrt{T}} + \frac{8(\zeta^2+\delta^2)L_g^2\sigma^2}{T(1-3\sigma^2)} - \frac{8(\zeta^2+\delta^2)L_g^2\sigma^2[3\sigma^2-(3\sigma^2)^T]}{T^2[1-3\sigma^2]^2}. \tag{18}$$

Note that the convergence analysis above is established for the Byzantine-free case, in which all agents are honest. Nevertheless, even one Byzantine agent can destroy the learning process via crafted attacks. Consider a scalar example as shown in Fig. 2. No matter what parameters the honest agents have, the Byzantine agent can send well-designed messages to the honest agents such that the weighted mean aggregations all yield 0. To address this issue, we devise a Byzantine-resilient variant of DP-RL in the next section.

IV. Byzantine-Resilient Decentralized Parallel Reinforcement Learning

A. Problem Statement

In the Byzantine case, the agents are divided into two sets: $\mathcal{H}$ as the set of honest agents and $\mathcal{B}$ as the set of Byzantine agents, with $\mathcal{H}\cup\mathcal{B}=\mathcal{N}$. The honest agents exactly follow the prescribed algorithm. However, the Byzantine agents can send arbitrarily wrong messages to their neighbors. In this case, it is impossible to maximize the expected long-term return averaged over all the agents as in (1). Instead, we resort to maximizing the expected long-term return averaged over the honest agents, given by

$$\max \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} J_i(\boldsymbol{\theta}). \tag{19}$$

B. Byzantine-Resilient Decentralized Parallel Policy Gradient (BRDP-PG)

To mitigate the influence of Byzantine attacks, we propose a Byzantine-resilient variant of DP-PG, called as BRDP-PG. Similar to DP-PG, BRDP-PG also has two steps at each iteration: local parameter update and neighboring message aggregation. At the local parameter update step, each honest agent i samples a trajectory τ_i^t and performs stochastic policy gradient to obtain $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ as in (3). A Byzantine agent can either follow (3) or not. At the neighboring message aggregation step, each honest agent i sends $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ to its neighbors, while a Byzantine agent sends arbitrarily wrong messages to its neighbors. Note that the wrong messages to different neighbors can be different. To describe the message transmissions over such a communication network, we denote the message sent by agent i as

$$\tilde{\boldsymbol{\theta}}_i^{t+\frac{1}{2}} = \begin{cases} \boldsymbol{\theta}_i^{t+\frac{1}{2}}, & \text{if } i\in\mathcal{H} \\ *, & \text{if } i\in\mathcal{B}, \end{cases} \tag{20}$$

where $*$ denotes an arbitrary D-dimensional vector.

Due to the presence of Byzantine attacks, using weighted mean to aggregate neighboring message is vulnerable. Therefore, in BRDP-PG, we instead use coordinate-wise trimmed-mean (CTM) for neighboring message aggregation. For each dimension $d=1,\cdots,D$ of the received neighboring messages, agent i removes q_i largest elements and q_i smallest elements, and then averages the rest and its own. Mathematically speaking, at each iteration t, each honest agent i divides its neighboring agents to three subsets at each dimension d, given by

$$\begin{aligned} \mathcal{N}_{i,d}^{t,+} &= \underset{\mathcal{X}:\{\mathcal{X}\subset\mathcal{N}_i,|\mathcal{X}|=q_i\}}{\arg\max} \sum_{j\in\mathcal{X}}\{\tilde{\boldsymbol{\theta}}_{j,d}^{t+\frac{1}{2}}\}, \\ \mathcal{N}_{i,d}^{t,-} &= \underset{\mathcal{X}:\{\mathcal{X}\subset\mathcal{N}_i,|\mathcal{X}|=q_i\}}{\arg\min} \sum_{j\in\mathcal{X}}\{\tilde{\boldsymbol{\theta}}_{j,d}^{t+\frac{1}{2}}\}, \\ \mathcal{N}_{i,d}^{t} &= \mathcal{N}_i \backslash \mathcal{N}_{i,d}^{t,+} \backslash \mathcal{N}_{i,d}^{t,-}. \end{aligned}$$

Then, agent i aggregates following

$$\begin{aligned} \boldsymbol{\theta}_{i,d}^{t+1} =& \mathcal{T}_{i,d}\Big(\{\tilde{\boldsymbol{\theta}}_j^{t+\frac{1}{2}}\}_{j\in\mathcal{N}_i\cup\{i\}}\Big) \\ :=& \frac{1}{|\mathcal{N}_i|-2q_i+1}\sum_{j\in\mathcal{N}_{i,d}^t\cup i}\tilde{\boldsymbol{\theta}}_{j,d}^{t+\frac{1}{2}}. \end{aligned} \tag{21}$$

We outline BRDP-PG in Algorithm 2.

C. Convergence Analysis

With slight abuse of notation, we write the learning parameters of the honest agents in a compact form as

$$\Theta^t = [\dots,\boldsymbol{\theta}_i^t,\dots]^\top, \tag{22}$$

where $i\in\mathcal{H}$ and $\Theta^t\in\mathbb{R}^{|\mathcal{H}|\times D}$. We also define the average learning parameter of the honest agents as $\bar{\boldsymbol{\theta}}^t = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{\theta}_i^t$.

We make the following assumptions on the network topology, the Byzantine agents, and the trimming parameters q_i.

Assumption 7: Denote Ω as the set of subgraphs of $\mathcal{G}$ by removing all Byzantine agents with their edges, as well as removing any additional q_i incoming edges at any honest agent $i\in\mathcal{H}$. For any subgraph $\mathcal{G}'\in\Omega$, there exists at least one agent i^* that has directed paths to all agents in $\mathcal{G}'$. The lengths of all paths are upper-bounded by D_Ω.

Algorithm 2 Byzantine-Resilient Decentralized Parallel Policy Gradient (BRDP-PG)

```
Initialize: Parameter vector $\boldsymbol{\theta}_i^0 = \boldsymbol{\theta}^0, \forall i \in \mathcal{N}$
for $t = 0, 1, \ldots, T$ do
    for agents $i = 1, \ldots, N$ in parallel do
        if $i \in \mathcal{H}$ then
            Calculate $\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)$ according to (4).
            Update $\boldsymbol{\theta}_i^{t+\frac{1}{2}}$ according to (3).
            Send $\tilde{\boldsymbol{\theta}}_i^{t+\frac{1}{2}} = \boldsymbol{\theta}_i^{t+\frac{1}{2}}$ to all neighbors.
        else
            Send $\tilde{\boldsymbol{\theta}}_i^{t+\frac{1}{2}} = *$ to all neighbors.
        end if
        if $i \in \mathcal{H}$ then
            for $d = 1, \ldots, D$ do
                Update $\boldsymbol{\theta}_{i,d}^{t+1}$ according to (21).
            end for
        end if
    end for
end for
```

Assumption 8: For any honest agent $i \in \mathcal{H}$, $|\mathcal{B}_i| \leq q_i < \frac{|\mathcal{N}_i|}{3}$, where $\mathcal{B}_i$ is the set of its Byzantine neighbors.

Assumption 7 requires the network topology remains connected after any honest agent $i \in \mathcal{H}$ removes all incoming edges from the Byzantine agents, as well as q_i additional incoming edges from the honest agents. We term D_Ω as the *network diameter*. Assumption 8 further requires the numbers of Byzantine agents are limited and the trimming parameters q_i are properly chosen. These assumptions are standard in analyzing Byzantine-resilient algorithms with CTM [25].

Lemma 2: Define the consensus error at iteration t as

$$\Delta^t = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} ||\boldsymbol{\theta}_i^t - \bar{\boldsymbol{\theta}}^t||^2. \tag{23}$$

Under Assumptions 1–5 and 7–8, with α^t satisfying

$$1 \leq \frac{\alpha^{t-1}}{\alpha^t} \leq \frac{2}{1+\eta_1}, \tag{24}$$

for some $\eta_1 \in (0,1)$, the consensus error of Algorithm 2 is bounded as

$$\mathbb{E}\Delta^t \leq \frac{16|\mathcal{H}|^2 C_g^2 \mu}{1-\mu}(\alpha^t)^2, \tag{25}$$

where the expectation is taken over all random trajectories $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^t$. while $\mu \in (0,1)$ increases monotonically as the network diameter D_Ω increases.

Remark: Lemma 2 reveals that the consensus error can be sufficiently small if the step size α^t is sufficiently small. Meanwhile, the network diameter D_Ω affects the convergence speed, implying that the trimming parameter q_i of any honest agent $i \in \mathcal{H}$ should be as close to $|\mathcal{B}_i|$ as possible. The proof can be found in Appendix F.

Theorem 2: Define the learning error at iteration T as

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2. \tag{26}$$

Under Assumptions 1–5 and 7–8, with $\alpha^t = \alpha = \frac{1}{\sqrt{T}}$ and $T \geq 12L^2$, the learning error of Algorithm 2 is bounded as

$$\begin{aligned} &\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \qquad (27) \\ \leq &\frac{2}{\sqrt{T}}[J(\boldsymbol{\theta}^*) - J(\bar{\boldsymbol{\theta}}^0)] + \frac{2\zeta^2 L}{|\mathcal{H}|\sqrt{T}} \\ &+ \frac{32|\mathcal{H}|^2 L_g^2 C_g^2 \mu}{T(1-\mu)} + \frac{96 P_{\mathcal{T}} |\mathcal{H}|^2 C_g^2 \mu}{1-\mu} + 16 P_{\mathcal{T}}(\zeta^2 + \delta^2). \end{aligned}$$

where the expectation is taken over all random trajectories $\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^t$, $\mu \in (0,1)$ and

$$P_{\mathcal{T}} := \left(\frac{|\mathcal{H}|}{\min_{i\in\mathcal{H}}\{|\mathcal{N}_i| - 2q_i + 1\}} - 1\right)^2.$$

Remark: The first term at the right-hand side of (27) remains the same as the one in (17), and vanishes at the rate of $O(\frac{1}{\sqrt{T}})$ when T increases. The second term now is proportional to $\frac{1}{|\mathcal{H}|}$, instead of $\frac{1}{N}$. The third, fourth and fifth terms come from the accumulation of the consensus errors. Although the third term vanishes at the rate of $O(\frac{1}{T})$, the fourth and fifth term are nonzero unless $P_{\mathcal{T}}$ is 0, which is almost impossible when Byzantine agents are present. The proof can be found in Appendix C.

V. Numerical Experiments

In this section, we conduct numerical experiments to show the data efficiency of DP-PG and the Byzantine-robustness of BRDP-PG. The code is available online[1].

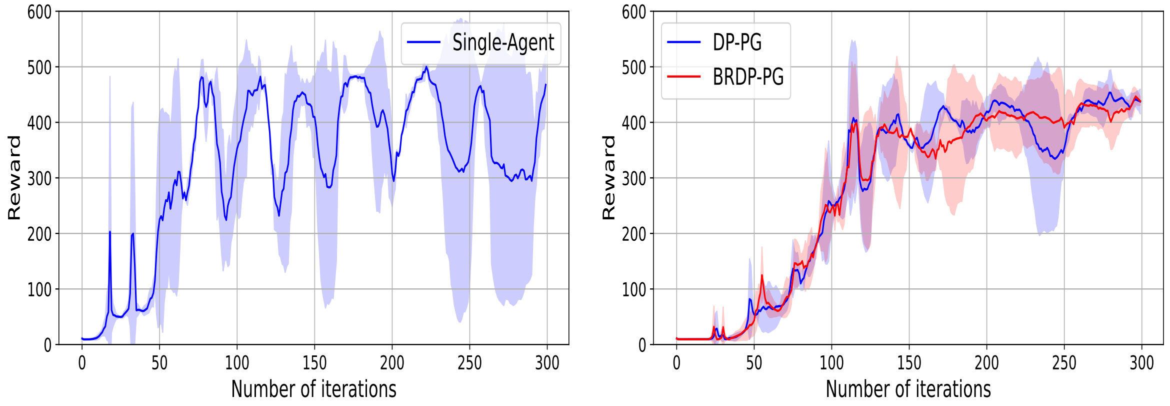
A. Experimental Settings

We consider two environments for numerical experiments, CartPole-V1 [44] and HalfCheetah-V2 [45].

- CartPole-V1: A pendulum stands upright on a cart initially. The goal is to keep the pendulum upright by applying forces in the left and right directions on the cart.
- HalfCheetah-V2: There is a 2-dimensional robot consisting of 9 links and 8 joints. The goal is to train the robot running to the right as fast as possible.

We assign each agent with a neural network consisting of two hidden layers, in which Relu and Tahn are adopted as activation functions for CartPole-V1 and HalfCheetah-V2, respectively. The numbers of hidden units for CartPole-V1 and HalfCheetah-V2 are (16,16) and (64,64), respectively. We optimize the learning parameters with stochastic gradient descent (SGD) and set the step size as 0.1 for both environments. Since using only one policy gradient sample suffers from large variance, a batch of policy gradient samples are used in practice. We set the batch size as 64 and 96 for CartPole-V1 and HalfCheetah-V2, respectively. The episode lengths H are maximally 500 and 1000 for CartPole-V1 and HalfCheetah-V2, respectively. Hyperparameters of these two environments are listed in Table I.

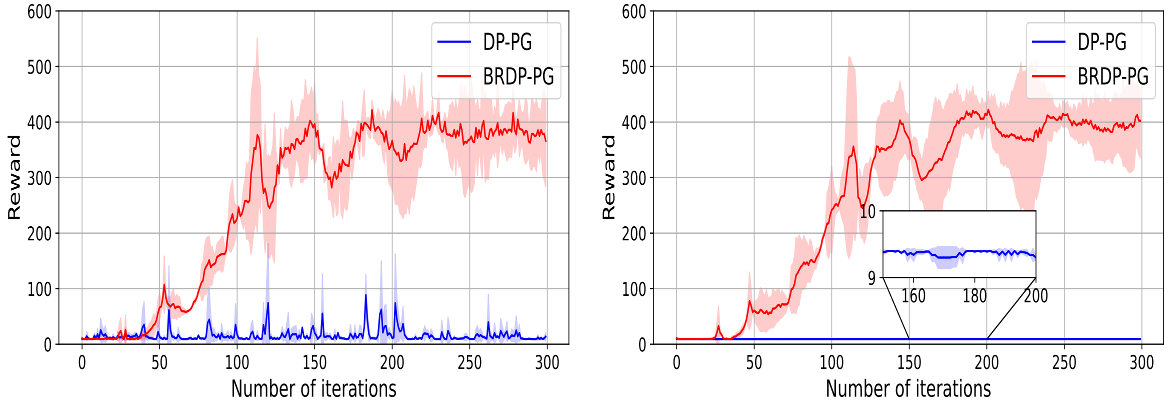
[1] https://github.com/TNNLS-Anonymous/BRDPPG



(a) $N = 1$

(b) $N = 10$

Fig. 3. The experimental results of CartPole-V1 with $N = 1$ and $N = 10$ for the Byzantine-free case.



(a) $N = 10$ with Gaussian attack.

(b) $N = 10$ with sign-flipping attack.

Fig. 4. The experimental results of CartPole-V1 with $N = 10$ for the Byzantine case.

TABLE I
PARAMETERS OF NEURAL NETWORKS AND TRAINING ALGORITHMS

	CartPole-V1	HalfCheetah-V2
Number of Hidden Units	(16,16)	(64,64)
Activation Function	ReLu	Tahn
Output Activation	Tahn	Tahn
Step Size	0.1	0.1
Batch Size	64	96
Maximum Episode Length	500	1000

1) Byzantine-free Case: To demonstrate the data efficiency of DP-PG, we set the number of agents as $N = 1$ and $N = 10$, respectively. For $N = 10$, the communication network is an Erdos-Renyi graph, in which each pair of agents are connected with probability 0.8. The weight matrix W used in DP-PG is constructed according to the Metropolis-Hastings rule [46].

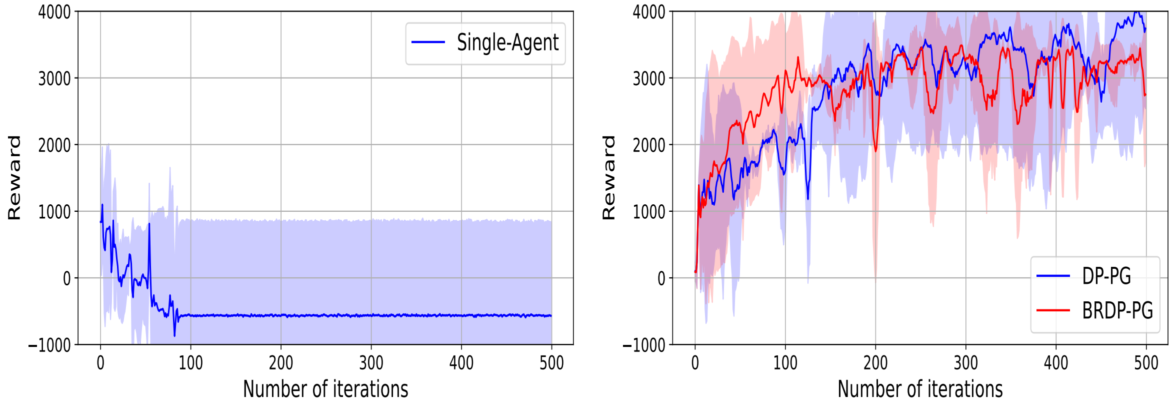
2) Byzantine Case: To evaluate the Byzantine-robustness of BRDP-PG, we use the Erdos-Renyi graph with $N = 10$ agents generated above, randomly set $|\mathcal{B}| = 1$ agent as Byzantine, but guarantee that the honest agents are connected. We launch two Byzantine attacks: Gaussian and sign-flipping [33]. For each honest agent $i \in \mathcal{H}$, we set $q_i = 1$.

- Gaussian attack: The elements of the messages sent from Byzantine agents are generated following the Gaussian distribution, whose mean and standard variance are 0 and 1, respectively.
- Sign-flipping attack: The Byzantine agents multiply their policy parameters with a negative constant $\nu = -2$, such that $\tilde{\boldsymbol{\theta}}_i^{t+\frac{1}{2}} = \nu \boldsymbol{\theta}_i^{t+\frac{1}{2}}$ for each Byzantine agent $i \in \mathcal{B}$.

We let all honest agents interact with their reset environments and use the averaged immediate reward as the performance metric. For each experimental setting, we repeat the numerical experiment for five times and report the averages.

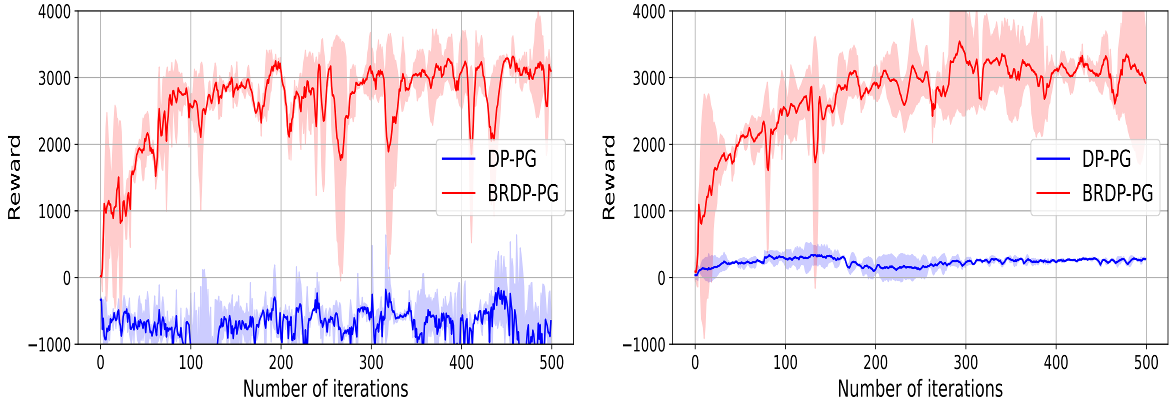
B. Experimental Results

1) CartPole-V1: Fig. 3 depicts the experimental results of CartPole-V1 for the Byzantine-free case. When $N = 1$, Fig. 3(a) shows that the single-agent policy gradient algorithm can quickly reach a high reward, but the reward is very unstable and the variance is quite large. In contrast, when $N = 10$, DP-PG steadily reaches a reward higher than 400, and the variance



(a) $N = 1$ (b) $N = 10$

Fig. 5. The experimental results of HalfCheetah-V2 with $N = 1$ and $N = 10$ for the Byzantine-free case.



(a) $N = 10$ with Gaussian attack. (b) $N = 10$ with sign-flipping attack.

Fig. 6. The experimental results of HalfCheetah-V2 with $N = 10$ for the Byzantine case.

is small, as shown in Fig. 3(b). This comparison demonstrates the effectiveness of DP-PG in improving data efficiency and hence stabilizing the training process. Fig. 3(b) also shows the performance of BRDP-PG. BRDP-PG is close to DP-PG in terms of both reward and variance, implying that it can also be applied in the Byzantine-free case.

Fig. 4 demonstrates the performance of DP-PG and BRDP-PG for the Byzantine case, when $N = 10$. DP-PG completely fails under the Gaussian attack. The sign-flipping attack appears to be less strong, but the corresponding reward of DP-PG is still low. In contrast, BRDP-PG is robust to both attacks.

2) HalfCheetah-V2: In Fig. 5, we show the experimental results of HalfCheetah-V2 for the Byzantine-free case. When $N = 1$, as demonstrated in Fig. 5(a), the single-agent policy gradient algorithm fails, revealing the negative impact brought by the small number of trajectories. Nevertheless, both DP-PG and BRDP-PG work well when $N = 10$; see Fig. 5(b).

When $N = 10$ and under the Byzantine attacks, the rewards and variances of DP-PG and BRDP-PG are shown in Fig. 6. The conclusions are consistent with those for Fig. 4. DP-PG fails under both attacks, and is even worse than the single-agent policy gradient algorithm under the Gaussian attack. On the other hand, BRDP-PG demonstrate satisfactory robustness to both Gaussian and sign-flipping attacks.

VI. Conclusions

This paper investigates decentralized methods for parallel RL. We first propose decentralized parallel policy gradient (DP-PG), where multiple agents exchange learning parameters via a peer-to-peer network for policy training. To deal with potential Byzantine attacks, we further develop Byzantine-resilient decentralized parallel policy gradient (BRDP-PG), which adopts coordinate trimmed-mean (CTM) to aggregate neighboring messages. We analyze the convergence properties of the two algorithms, revealing the data efficiency of DP-PG and Byzantine-robustness of BRDP-PG. Finally, we conduct numerical experiments on CartPole-V1 and HalfCheetah-V2 to validate the effectiveness of our proposed algorithms.

In CTM, each honest agent $i \in \mathcal{H}$ discards $2q_i$ receiving messages at each coordinate with $q_i \geq |\mathcal{B}_i|$, which inevitably leads to loss of useful information. In the future work, we will investigate the developments and applications of other robust aggregation rules.

APPENDIX A
PROOF OF THEOREM 1

According to Proposition 2 and $J(\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} J_i(\boldsymbol{\theta})$, we have

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}^t}[J(\bar{\boldsymbol{\theta}}^{t+1})] \\
\geq&\mathbb{E}_{\boldsymbol{\tau}^t}[J(\bar{\boldsymbol{\theta}}^t)+\langle\nabla J(\bar{\boldsymbol{\theta}}^t),\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t\rangle-\frac{L}{2}||\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t||^2] \\
=&J(\bar{\boldsymbol{\theta}}^t)+\underbrace{\mathbb{E}_{\boldsymbol{\tau}^t}[\langle\nabla J(\bar{\boldsymbol{\theta}}^t),\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t\rangle]}_{\textcircled{1}}-\frac{L}{2}\underbrace{\mathbb{E}_{\boldsymbol{\tau}^t}[||\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t||^2]}_{\textcircled{2}}.
\end{aligned}\tag{28}$$

With the equality $\langle\boldsymbol{x},\boldsymbol{y}\rangle=\frac{1}{2}||\boldsymbol{x}||^2+\frac{1}{2}||\boldsymbol{y}||^2-\frac{1}{2}||\boldsymbol{x}-\boldsymbol{y}||^2$, we can rewrite term ① in (28) as:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}^t}\left[\langle\nabla J(\bar{\boldsymbol{\theta}}^t),\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t\rangle\right] \\
=&\alpha^t\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left\langle\nabla J(\bar{\boldsymbol{\theta}}^t),\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right\rangle\right] \\
=&\alpha^t\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left\langle\nabla J(\bar{\boldsymbol{\theta}}^t),\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right\rangle\right] \\
=&\frac{\alpha^t}{2}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2-\frac{\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
&+\frac{\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right].
\end{aligned}\tag{29}$$

Further, term ② in (28) is bounded by

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}^t}\left[||\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t||^2\right] \\
=&\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\alpha^t\cdot\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
=&(\alpha^t)^2\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
=&(\alpha^t)^2\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right.\right.\right. \\
&\left.\left.\left.+\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\nabla J(\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
\leq&2(\alpha^t)^2\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
&+2(\alpha^t)^2\mathbb{E}_{\boldsymbol{\tau}^t}\left[||\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\nabla J(\bar{\boldsymbol{\theta}}^t)||^2\right] \\
\overset{(i)}{\leq}&2(\alpha^t)^2\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
&+\frac{2(\alpha^t)^2\zeta^2}{N},
\end{aligned}\tag{30}$$

where (i) uses Assumption 3.

With (28) rearranged as

$$\begin{aligned}
&\frac{L}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left[||\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t||^2\right]+\mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1})\right]-J(\bar{\boldsymbol{\theta}}^t) \\
\geq&\mathbb{E}_{\boldsymbol{\tau}^t}\left[\langle\nabla J(\bar{\boldsymbol{\theta}}^t),\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t\rangle\right],
\end{aligned}\tag{31}$$

substituting (29) and (30) into (31) yields

$$\begin{aligned}
&\frac{(\alpha^t)^2\zeta^2L}{N}+\mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1})\right]-J(\bar{\boldsymbol{\theta}}^t) \\
&+(\alpha^t)^2L\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
\geq&\frac{\alpha^t}{2}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2-\frac{\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\
&+\frac{\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right].
\end{aligned}\tag{32}$$

Further rearranging (32) yields

$$\begin{aligned}
&\frac{\alpha^t}{2}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \\
\leq&\ \mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1})\right]-J(\bar{\boldsymbol{\theta}}^t)+\frac{(\alpha^t)^2\zeta^2L}{N} \\
&+\frac{\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2 \\
&+\frac{2(\alpha^t)^2L-\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left|\left|\nabla J(\bar{\boldsymbol{\theta}}^t)-\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)+\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2.
\end{aligned}\tag{33}$$

When $\alpha^t \leq \frac{1}{2L}$, $\frac{2(\alpha^t)^2L-\alpha^t}{2} \leq 0$ and we have

$$\begin{aligned}
\frac{\alpha^t}{2}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \leq&\ \mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1})\right]-J(\bar{\boldsymbol{\theta}}^t)+\frac{(\alpha^t)^2\zeta^2L}{N} \\
&+\frac{\alpha^t}{2}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right].
\end{aligned}\tag{34}$$

Thus, we have

$$\begin{aligned}
||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \leq&\ \frac{2}{\alpha^t}\mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1})-J(\bar{\boldsymbol{\theta}}^t)\right]+\frac{2\alpha^t\zeta^2L}{N} \\
&\underbrace{+\,\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t)\right|\right|^2\right]}_{\textcircled{3}}.
\end{aligned}\tag{35}$$

To bound term ③ in (35), we observe that

$$\begin{aligned}
&\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1}-\bar{\boldsymbol{\theta}}^t) \\
=&\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{\alpha^tN}\sum_{i\in\mathcal{N}}\left[\sum_{j\in\mathcal{N}_i\cup\{i\}}w_{ij}\big(\boldsymbol{\theta}_j^t+\nabla J_j(\tau_j^t|\boldsymbol{\theta}_j^t)\big)-\bar{\boldsymbol{\theta}}^t\right] \\
=&\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{N}\sum_{i\in\mathcal{N}}\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t) \\
&\underbrace{+\frac{1}{\alpha^tN}\sum_{i\in\mathcal{N}}\left[\bar{\boldsymbol{\theta}}^t-\sum_{j\in\mathcal{N}_i\cup\{i\}}w_{ij}\boldsymbol{\theta}_j^t\right]}_{:=0} \\
=&\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t)-\frac{1}{N}\sum_{i\in\mathcal{N}}\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t).
\end{aligned}\tag{36}$$

Taking expectation on (36) yields

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t) - \frac{1}{N}\sum_{i\in\mathcal{N}}\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)\right|\right|^2\right] \qquad (37)\\
&= \mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\frac{1}{N}\sum_{i\in\mathcal{N}}\Big(\nabla J_i(\tau_i^t|\bar{\boldsymbol{\theta}}^t) - \nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)\Big)\right|\right|^2\right]\\
&\le \frac{1}{N}\sum_{i\in\mathcal{N}}\mathbb{E}_{\tau_i^t}\left[\left|\left|\nabla J_i(\tau_i^t|\bar{\boldsymbol{\theta}}^t) - \nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)\right|\right|^2\right]\\
&\overset{(ii)}{\le} \frac{L_g^2}{N}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\sum_{i\in\mathcal{N}}||\bar{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_i^t||^2\right],
\end{aligned}$$

where (ii) uses Proposition 1.

Substituting (37) into (35), we have

$$\begin{aligned}
&||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \qquad (38)\\
\le&\frac{2}{\alpha^t}\mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1}) - J(\bar{\boldsymbol{\theta}}^t)\right] + \frac{2\alpha^t\zeta^2L}{N}\\
&+ \mathbb{E}_{\boldsymbol{\tau}^t}\left[\left|\left|\nabla J(\boldsymbol{\tau}^t|\bar{\boldsymbol{\theta}}^t) - \frac{1}{\alpha^t}(\bar{\boldsymbol{\theta}}^{t+1} - \bar{\boldsymbol{\theta}}^t)\right|\right|^2\right]\\
\le&\frac{2}{\alpha^t}\mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1}) - J(\bar{\boldsymbol{\theta}}^t)\right] + \frac{L_g^2}{N}\mathbb{E}_{\boldsymbol{\tau}^t}\left[\sum_{i\in\mathcal{N}}||\bar{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_i^t||^2\right]\\
&+ \frac{2\alpha^t\zeta^2L}{N}\\
=&\frac{2}{\alpha^t}\mathbb{E}_{\boldsymbol{\tau}^t}\left[J(\bar{\boldsymbol{\theta}}^{t+1}) - J(\bar{\boldsymbol{\theta}}^t)\right] + L_g^2\mathbb{E}_{\boldsymbol{\tau}^t}\Delta^t + \frac{2\alpha^t\zeta^2L}{N}.
\end{aligned}$$

With $\alpha^t = \alpha \le \frac{1}{2L}$, taking expectation on (38) and averaging over $t = 1,\dots,T$, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla J(\bar{\boldsymbol{\theta}}^t)||^2 \qquad (39)\\
\le&\frac{2\mathbb{E}\left[J(\bar{\boldsymbol{\theta}}^{T+1}) - J(\bar{\boldsymbol{\theta}}^0)\right]}{\alpha T} + \frac{L_g^2}{T}\sum_{t=1}^{T}\mathbb{E}\Delta^t + \frac{2\alpha\zeta^2L}{N}\\
\le&\frac{2\left[J(\boldsymbol{\theta}^*) - J(\bar{\boldsymbol{\theta}}^0)\right]}{\alpha T} + \frac{L_g^2}{T}\sum_{t=1}^{T}\mathbb{E}\Delta^t + \frac{2\alpha\zeta^2L}{N},
\end{aligned}$$

which completes the proof.

Appendix B Proof of Lemma 1

According to Assumption 6, W is doubly stochastic. Using this property, we have

$$\begin{aligned}
\Delta^{t+1} =&\frac{1}{N}\left|\left|\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\Theta^{t+1}\right|\right|^2 \qquad (40)\\
=&\frac{1}{N}\left|\left|\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)W\Theta^{t+\frac{1}{2}}\right|\right|^2\\
=&\frac{1}{N}\left|\left|\left(W - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right)\left(\Theta^{t+\frac{1}{2}} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\Theta^{t+\frac{1}{2}}\right)\right|\right|^2\\
\le&\left|\left|W - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\right|\right|^2 \cdot \frac{1}{N}\left|\left|\Theta^{t+\frac{1}{2}} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top\Theta^{t+\frac{1}{2}}\right|\right|^2\\
\le&\sigma^2\frac{1}{N}\sum_{i\in\mathcal{N}}||\boldsymbol{\theta}_i^{t+\frac{1}{2}} - \bar{\boldsymbol{\theta}}^{t+\frac{1}{2}}||^2,
\end{aligned}$$

where σ is the largest singular value of $W - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$. Taking expectation over $\boldsymbol{\tau}^t$ and using Lemma 2 in [41], when $\alpha \le \frac{1}{2\sqrt{3}L}$ we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\tau}^t}\Delta^{t+1} &\le \sigma^2\frac{1}{N}\sum_{i\in\mathcal{N}}\mathbb{E}_{\boldsymbol{\tau}^t}\left[||\boldsymbol{\theta}_i^{t+\frac{1}{2}} - \bar{\boldsymbol{\theta}}^{t+\frac{1}{2}}||^2\right] \qquad (41)\\
&\le \sigma^2\left(\frac{3}{N}\sum_{i\in\mathcal{N}}||\boldsymbol{\theta}_i^t - \bar{\boldsymbol{\theta}}^t||^2 + 8\alpha^2(\zeta^2 + \delta^2)\right)\\
&\le 3\sigma^2\Delta^t + 8\sigma^2\alpha^2(\zeta^2 + \delta^2).
\end{aligned}$$

Using telescopic cancellation on (41) from 0 to t, when $\sigma^2 \le \frac{1}{3}$ we have

$$\begin{aligned}
\mathbb{E}\Delta^t &\le (3\sigma^2)^t\Delta^0 + \frac{8\alpha^2(\zeta^2 + \delta^2)\sigma^2[1 - (3\sigma^2)^t]}{1 - 3\sigma^2} \qquad (42)\\
&= \frac{8\alpha^2(\zeta^2 + \delta^2)\sigma^2[1 - (3\sigma^2)^t]}{1 - 3\sigma^2},
\end{aligned}$$

where the last equality holds since $\Delta^0 = 0$ by hypothesis.

Appendix C Proof of Theorem 2

The first part of the proof is similar to that of Theorem 1. However, term ③ in (35) should be changed to

$$\begin{aligned}
&\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) - \frac{1}{\alpha^t}(\check{\boldsymbol{\theta}}^{t+1} - \check{\boldsymbol{\theta}}^t) \qquad (43)\\
=&\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) - \frac{1}{\alpha^t|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left[\mathcal{T}_i\Big(\{\tilde{\boldsymbol{\theta}}_j^{t+\frac{1}{2}}\}_{j\in\mathcal{N}_i\cup\{i\}}\Big) - \check{\boldsymbol{\theta}}^t\right]\\
=&\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) + \frac{1}{\alpha^t|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left[\check{\boldsymbol{\theta}}^t - \mathcal{T}_i\Big(\{\tilde{\boldsymbol{\theta}}_j^{t+\frac{1}{2}}\}_{j\in\mathcal{N}_i\cup\{i\}}\Big)\right]\\
=&\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) + \frac{1}{\alpha^t|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left[\check{\boldsymbol{\theta}}^t + \frac{\alpha^t}{|\mathcal{H}|}\sum_{j\in\mathcal{H}}\nabla J_j(\tau_j^t|\boldsymbol{\theta}_j^t)\right.\\
&\left.- \mathcal{T}_i\Big(\{\tilde{\boldsymbol{\theta}}_j^{t+\frac{1}{2}}\}_{j\in\mathcal{N}_i\cup\{i\}}\Big) - \frac{\alpha^t}{|\mathcal{H}|}\sum_{j\in\mathcal{H}}\nabla J_j(\tau_j^t|\boldsymbol{\theta}_j^t)\right]\\
=&\underbrace{\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t) +}_{①}\\
&\underbrace{\frac{1}{\alpha^t|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left(\check{\boldsymbol{\theta}}^t + \frac{\alpha^t}{|\mathcal{H}|}\sum_{j\in\mathcal{H}}\nabla J_j(\tau_j^t|\boldsymbol{\theta}_j^t) - \mathcal{T}_i\Big(\{\tilde{\boldsymbol{\theta}}_j^{t+\frac{1}{2}}\}_{j\in\mathcal{N}_i\cup\{i\}}\Big)\right)}_{②}
\end{aligned}$$

where $\mathcal{T}_i(\cdot)$ denotes the CTM operation.

Denote the ℓ_2 norms of terms ① and ② in (43) as T_1, and T_2, respectively. Then, we establish their upper bounds as follows.

Upper bound of T_1. For T_1, it holds that

$$\begin{aligned} T_1 &= \left|\left|\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) - \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)\right|\right|^2 \\ &= \left|\left|\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \Big(\nabla J_i(\tau_i^t|\check{\boldsymbol{\theta}}^t) - \nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)\Big)\right|\right|^2 \\ &\leq \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \left|\left|\nabla J_i(\tau_i^t|\check{\boldsymbol{\theta}}^t) - \nabla J_i(\tau_i^t|\boldsymbol{\theta}_i^t)\right|\right|^2 \\ &\overset{(i)}{\leq} \frac{L_g^2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} ||\check{\boldsymbol{\theta}}^t - \boldsymbol{\theta}_i^t||^2, \end{aligned} \tag{44}$$

where (i) uses Proposition 1.

Upper bound of T_2. According to [25], CTM in (21) intrinsically yields a certain linear combination of honest messages, given by

$$\boldsymbol{\theta}_{i,d}^{t+1} = \sum_{j\in\mathcal{H}_i\cup\{i\}} y_{j,d}^t \boldsymbol{\theta}_{j,d}^{t+\frac{1}{2}}, \tag{45}$$

where $\mathcal{H}_i$ is the set of honest neighbors of agent i, while the coefficients $y_{j,d}^t \in [0,1]$ and satisfy $\sum_{j\in\mathcal{H}_i\cup\{i\}} y_{j,d}^t = 1$. We also denote $\mathcal{B}_i = \mathcal{N}_i\backslash\mathcal{H}_i$ as the set of Byzantine neighbors of agent i. For the ease of subsequent theoretical analysis, we express the update above in a compact form as

$$\check{\Theta}_d^{t+1} = Y^t(d)\check{\Theta}_d^{t+\frac{1}{2}}, \tag{46}$$

where $\check{\Theta}_d^{t+1}$ and $\check{\Theta}_d^{t+\frac{1}{2}}$ respectively stack $\boldsymbol{\theta}_{i,d}^{t+1}$ and $\boldsymbol{\theta}_{j,d}^{t+\frac{1}{2}}$ to $|\mathcal{H}|\times 1$ vectors, and $Y^t(d) \in \mathbb{R}^{|\mathcal{H}|\times|\mathcal{H}|}$ a row stochastic matrix. Defining $\nabla J(\check{\boldsymbol{\tau}}^t|\check{\Theta}^t)_d := [\nabla J(\tau_1^t|\boldsymbol{\theta}_1^t)_d; \ldots; \nabla J(\tau_{|\mathcal{H}|}^t|\boldsymbol{\theta}_{|\mathcal{H}|}^t)_d] \in \mathbb{R}^{|\mathcal{H}|}$, we have

$$\check{\Theta}_d^{t+1} = Y^t(d)[\check{\Theta}_d^t + \alpha^t\nabla J(\check{\boldsymbol{\tau}}^t|\check{\Theta}^t)_d]. \tag{47}$$

Therefore, we have

$$\begin{aligned} T_2 &= \left|\left|\frac{1}{\alpha^t|\mathcal{H}|}\sum_{i\in\mathcal{H}} \Big(\check{\boldsymbol{\theta}}^{t+\frac{1}{2}} - \mathcal{T}_i\Big(\{\tilde{\boldsymbol{\theta}}_j^{t+\frac{1}{2}}\}_{j\in\mathcal{N}_i\cup\{i\}}\Big)\Big)\right|\right|^2 \\ &= \sum_{d=1}^{D}\left[\frac{1}{\alpha^t|\mathcal{H}|}\mathbf{1}^\top\Big(Y^t(d)\check{\Theta}_d^{t+\frac{1}{2}} - \frac{1}{|\mathcal{H}|}\mathbf{1}\mathbf{1}^\top\check{\Theta}_d^{t+\frac{1}{2}}\Big)\right]^2 \\ &= \frac{1}{(\alpha^t)^2}\sum_{d=1}^{D}\left[\frac{1}{|\mathcal{H}|}(\mathbf{1}^\top Y^t(d) - \mathbf{1}^\top)\cdot\Big(\check{\Theta}_d^{t+\frac{1}{2}} - \frac{1}{|\mathcal{H}|}\mathbf{1}\mathbf{1}^\top\check{\Theta}_d^{t+\frac{1}{2}}\Big)\right]^2 \\ &\leq \frac{1}{(\alpha^t)^2}\sum_{d=1}^{D}||\frac{1}{|\mathcal{H}|}(\mathbf{1}^\top Y^t(d) - \mathbf{1}^\top)||^2\left|\left|\check{\Theta}_d^{t+\frac{1}{2}} - \frac{1}{|\mathcal{H}|}\mathbf{1}\mathbf{1}^\top\check{\Theta}_d^{t+\frac{1}{2}}\right|\right|^2 \\ &\overset{(ii)}{\leq} \frac{P_{\mathcal{T}}}{(\alpha^t)^2|\mathcal{H}|}\left|\left|\check{\Theta}^{t+\frac{1}{2}} - \frac{1}{|\mathcal{H}|}\mathbf{1}\mathbf{1}^\top\check{\Theta}^{t+\frac{1}{2}}\right|\right|_F^2 \\ &= \frac{P_{\mathcal{T}}}{(\alpha^t)^2|\mathcal{H}|}\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^{t+\frac{1}{2}} - \check{\boldsymbol{\theta}}^{t+\frac{1}{2}}||^2, \end{aligned} \tag{48}$$

where $P_{\mathcal{T}} := (\frac{|\mathcal{H}|}{\min_{i\in\mathcal{H}}\{|\mathcal{N}_i|-2q_i+1\}} - 1)^2$ and (ii) follows the derivation in [47].

According to the upper bounds (44) and (48), we have

$$\begin{aligned} &\mathbb{E}_{\check{\boldsymbol{\tau}}^t}\left[\left|\left|\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) - \frac{1}{\alpha^t}(\check{\boldsymbol{\theta}}^{t+1} - \check{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\ \leq& 2\mathbb{E}_{\check{\boldsymbol{\tau}}^t}T_1 + 2\mathbb{E}_{\check{\boldsymbol{\tau}}^t}T_2 \\ \leq& \frac{2L_g^2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2 + \frac{2P_{\mathcal{T}}}{(\alpha^t)^2|\mathcal{H}|}\mathbb{E}_{\check{\boldsymbol{\tau}}^t}\left[\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^{t+\frac{1}{2}} - \check{\boldsymbol{\theta}}^{t+\frac{1}{2}}||^2\right]. \end{aligned} \tag{49}$$

According Lemma 2 in [41], we have

$$\begin{aligned} &\mathbb{E}_{\check{\boldsymbol{\tau}}^t}\left[\left|\left|\nabla J(\check{\boldsymbol{\tau}}^t|\check{\boldsymbol{\theta}}^t) - \frac{1}{\alpha^t}(\check{\boldsymbol{\theta}}^{t+1} - \check{\boldsymbol{\theta}}^t)\right|\right|^2\right] \\ \leq& \frac{2L_g^2}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2 \\ &+ \frac{2P_{\mathcal{T}}}{(\alpha^t)^2}\cdot\left[\frac{3}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\left|\left|\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t\right|\right|^2 + 8(\alpha^t)^2(\zeta^2+\delta^2)\right] \\ =& \left(\frac{2L_g^2}{|\mathcal{H}|} + \frac{6P_{\mathcal{T}}}{(\alpha^t)^2|\mathcal{H}|}\right)\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2 + 16P_{\mathcal{T}}(\zeta^2+\delta^2), \end{aligned} \tag{50}$$

when $\alpha^t \leq \frac{1}{2\sqrt{3}L}$.

Rewriting (35) and substituting (50) to it yield

$$\begin{aligned} ||\nabla J(\check{\boldsymbol{\theta}}^t)||^2 \leq& \frac{2}{\alpha^t}\mathbb{E}_{\check{\boldsymbol{\tau}}^t}[J(\check{\boldsymbol{\theta}}^{t+1}) - J(\check{\boldsymbol{\theta}}^t)] \\ &+ \left(2L_g^2 + \frac{6P_{\mathcal{T}}}{(\alpha^t)^2}\right)\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2 \\ &+ 16P_{\mathcal{T}}(\zeta^2+\delta^2) + \frac{2\alpha^t\zeta^2 L}{|\mathcal{H}|}. \end{aligned} \tag{51}$$

Taking expectation on (51) and averaging over $t = 1, \ldots, T$, we have

$$\begin{aligned} &\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla J(\check{\boldsymbol{\theta}}^t)||^2 \\ \leq& \frac{1}{T}\sum_{t=1}^{T}\frac{2}{\alpha^t}\mathbb{E}[J(\check{\boldsymbol{\theta}}^{t+1}) - J(\check{\boldsymbol{\theta}}^t)] \\ &+ \frac{1}{T}\sum_{t=1}^{T}\left(2L_g^2 + \frac{6P_{\mathcal{T}}}{(\alpha^t)^2}\right)\mathbb{E}\left[\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2\right] \\ &+ 16P_{\mathcal{T}}(\zeta^2+\delta^2) + \frac{2\zeta^2 L}{|\mathcal{H}|}\frac{1}{T}\sum_{t=1}^{T}\alpha^t. \end{aligned} \tag{52}$$

Applying Lemma 2 to (52) with $\alpha^t = \alpha = \frac{1}{\sqrt{T}}$ and $T \geq$

$12L^2$ yields

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}||\nabla J(\check{\boldsymbol{\theta}}^t)||^2 \qquad (53)\\
\leq&\frac{2}{\sqrt{T}}\sum_{t=1}^{T}\mathbb{E}[J(\check{\boldsymbol{\theta}}^{t+1})-J(\check{\boldsymbol{\theta}}^{t})]+16P_{\mathcal{T}}(\zeta^2+\delta^2)+\frac{2\zeta^2 L}{|\mathcal{H}|\sqrt{T}}\\
&+\frac{1}{T}\sum_{t=1}^{T}(2L_g^2+6TP_{\mathcal{T}})\bigg(\frac{16|\mathcal{H}|^2C_g^2\mu}{T(1-\mu)}\bigg)\\
\leq&\frac{2}{\sqrt{T}}\sum_{t=1}^{T}\mathbb{E}[J(\check{\boldsymbol{\theta}}^{t+1})-J(\check{\boldsymbol{\theta}}^{t})]+\frac{32|\mathcal{H}|^2C_g^2L_g^2\mu}{T(1-\mu)}\\
&+\frac{96P_{\mathcal{T}}|\mathcal{H}|^2C_g^2\mu}{1-\mu}+16P_{\mathcal{T}}(\zeta^2+\delta^2)+\frac{2\zeta^2 L}{|\mathcal{H}|\sqrt{T}}\\
\leq&\frac{2\mathbb{E}[J(\check{\boldsymbol{\theta}}^{T+1})-J(\check{\boldsymbol{\theta}}^{0})]}{\sqrt{T}}+\frac{32|\mathcal{H}|^2C_g^2L_g^2\mu}{T(1-\mu)}+\frac{96P_{\mathcal{T}}|\mathcal{H}|^2C_g^2\mu}{1-\mu}\\
&+16P_{\mathcal{T}}(\zeta^2+\delta^2)+\frac{2\zeta^2 L}{|\mathcal{H}|\sqrt{T}}\\
\leq&\frac{2J(\check{\boldsymbol{\theta}}^{*})-J(\check{\boldsymbol{\theta}}^{0})}{\sqrt{T}}+\frac{32|\mathcal{H}|^2C_g^2L_g^2\mu}{T(1-\mu)}+\frac{96P_{\mathcal{T}}|\mathcal{H}|^2C_g^2\mu}{1-\mu}\\
&+16P_{\mathcal{T}}(\zeta^2+\delta^2)+\frac{2\zeta^2 L}{|\mathcal{H}|\sqrt{T}},
\end{aligned}
$$

which completes the proof.

Appendix D
Proof of Proposition 1

According to the definition of the GPOMDP estimator in (4), we have

$$
\begin{aligned}
&||\nabla J_i(\tau_i|\boldsymbol{\theta}_i)|| \qquad (54)\\
=&\bigg|\bigg|\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\bigg)\big((\gamma)^h r_i^h-b_i^h\big)\bigg|\bigg|\\
\leq&\sum_{h=0}^{H-1}\bigg|\bigg|\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\bigg)\big((\gamma)^h r_i^h-b_i^h\big)\bigg|\bigg|\\
\leq&(r_{\max}+b_{\max})\sum_{h=0}^{H-1}(\gamma)^h\sum_{k=0}^{h}||\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)||\\
\overset{(i)}{\leq}&(r_{\max}+b_{\max})\sum_{h=0}^{H-1}(\gamma)^h(h+1)G\\
\leq&HG(r_{\max}+b_{\max})\sum_{h=0}^{H-1}(\gamma)^h\\
\overset{(ii)}{\leq}&\frac{HG(r_{\max}+b_{\max})}{1-\gamma},
\end{aligned}
$$

where (i) uses the fact that $\sum_{h=0}^{H-1}(\gamma)^h=\frac{(1-(\gamma)^H)}{1-\gamma}\leq\frac{1}{1-\gamma}$ and (ii) uses Assumption 2. Similarly, we have

$$
\begin{aligned}
&||\nabla J_i(\tau_i|\boldsymbol{\theta}_i)-\nabla J(\tau_i|\boldsymbol{\theta}_i')|| \qquad (55)\\
=&\bigg|\bigg|\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\bigg)\big((\gamma)^h r_i^h-b_i^h\big)-\\
&\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i'}\log\pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)\bigg)\big((\gamma)^h r_i^h-b_i^h\big)\bigg|\bigg|\\
=&\bigg|\bigg|\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)-\nabla_{\boldsymbol{\theta}_i'}\log\pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)\bigg)\\
&\cdot\big((\gamma)^h r_i^h-b_i^h\big)\bigg|\bigg|\\
\leq&\sum_{h=0}^{H-1}\bigg|\bigg|\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)-\nabla_{\boldsymbol{\theta}_i'}\log\pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)\bigg)\\
&\cdot\big((\gamma)^h r_i^h-b_i^h\big)\bigg|\bigg|\\
\leq&(r_{\max}+b_{\max})\\
&\cdot\sum_{h=0}^{H-1}(\gamma)^h\sum_{k=0}^{h}||\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a^k|s^k)-\nabla_{\boldsymbol{\theta}_i'}\log\pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)||\\
\overset{(i)}{\leq}&(r_{\max}+b_{\max})\sum_{h=0}^{H-1}(\gamma)^h(h+1)M||\boldsymbol{\theta}_i-\boldsymbol{\theta}_i'||\\
\leq&HM(r_{\max}+b_{\max})\sum_{h=0}^{H-1}(\gamma)^h||\boldsymbol{\theta}_i-\boldsymbol{\theta}_i'||\\
\overset{(ii)}{\leq}&\frac{HM(r_{\max}+b_{\max})}{1-\gamma}||\boldsymbol{\theta}_i-\boldsymbol{\theta}_i'||,
\end{aligned}
$$

where (i) uses the fact that $\sum_{h=0}^{H-1}(\gamma)^h=\frac{(1-(\gamma)^H)}{1-\gamma}\leq\frac{1}{1-\gamma}$ and (ii) uses Assumption 2.

Appendix E
Proof of Proposition 2

For any agent $i=1,\ldots,N$ and $\boldsymbol{\theta}_i$, $\boldsymbol{\theta}_i'\in\mathbb{R}^d$, according to (2) and Lemma 3 in [17], we have

$$
\begin{aligned}
&||\nabla J_i(\boldsymbol{\theta}_i)-\nabla J_i(\boldsymbol{\theta}_i')|| \qquad (56)\\
=&\bigg|\bigg|\underbrace{\mathbb{E}_{\tau_i\sim p(\tau_i|\pi_{\boldsymbol{\theta}_i})}\bigg[\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\bigg)(\gamma)^h r_i^h\bigg]}_{①}\\
&\underbrace{-\mathbb{E}_{\tau_i\sim p(\tau_i|\pi_{\boldsymbol{\theta}_i'})}\bigg[\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i'}\log\pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)\bigg)(\gamma)^h r_i^h\bigg]}_{①}\\
&\underbrace{+\mathbb{E}_{\tau_i\sim p(\tau_i|\pi_{\boldsymbol{\theta}_i'})}\bigg[\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\bigg)(\gamma)^h r_i^h]}_{②}\\
&\underbrace{-\mathbb{E}_{\tau_i\sim p(\tau_i|\pi_{\boldsymbol{\theta}_i'})}\bigg[\sum_{h=0}^{H-1}\bigg(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i'}\log\pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)\bigg)(\gamma)^h r_i^h\bigg]}_{②}\bigg|\bigg|.
\end{aligned}
$$

For term ① in (56), we bound its norm as

$$\begin{aligned}
&\Big\|\Big\|\mathbb{E}_{\tau_i\sim p(\tau_i|\pi_{\boldsymbol{\theta}_i})}\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\\
&\quad-\mathbb{E}_{\tau_i\sim p(\tau_i|\pi_{\boldsymbol{\theta}'_i})}\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\Big\|\Big\|\\
=&\Big\|\Big\|\int p(\tau_i|\pi_{\boldsymbol{\theta}_i})\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\\
&\quad-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\mathrm{d}\tau_i\Big\|\Big\|\\
=&\Big\|\Big\|\int\big(p(\tau_i|\pi_{\boldsymbol{\theta}_i})-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\big)\\
&\quad\cdot\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\mathrm{d}\tau_i\Big\|\Big\|\\
\le&\int\big|p(\tau_i|\pi_{\boldsymbol{\theta}_i})-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\big|\\
&\quad\cdot\Big\|\Big\|\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big\|\Big\|\mathrm{d}\tau_i\\
\le&\int\big|p(\tau_i|\pi_{\boldsymbol{\theta}_i})-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\big|\\
&\quad\cdot\Big(\sum_{h=0}^{H-1}(\gamma)^h r_i^h\sum_{k=0}^{h}\big|\big|\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\big|\big|\Big)\mathrm{d}\tau_i\\
\le&\int\big|p(\tau_i|\pi_{\boldsymbol{\theta}_i})-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\big|\cdot\Big(\sum_{h=0}^{H-1}(h+1)G(\gamma)^h r_i^h\Big)\mathrm{d}\tau_i\\
\le&\frac{HGr_{\max}}{1-\gamma}\int\underbrace{\big|p(\tau_i|\pi_{\boldsymbol{\theta}_i})-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\big|}_{\textcircled{3}}\mathrm{d}\tau_i.
\end{aligned}\tag{57}$$

The last inequality holds due to Assumption 2. For term ③ in (57), we rewrite it as

$$\begin{aligned}
&\big|p(\tau_i|\pi_{\boldsymbol{\theta}_i})-p(\tau_i|\pi_{\boldsymbol{\theta}'_i})\big|\\
=&\Big|\rho(s_i^0)\pi_{\boldsymbol{\theta}_i}(a_i^0|s_i^0)\prod_{h=1}^{H-1}p(s_i^h|s_i^{h-1},a_i^{h-1})\pi_{\boldsymbol{\theta}_i}(a_i^h|s_i^h)\\
&\quad-\rho(s_i^0)\pi_{\boldsymbol{\theta}'_i}(a_i^0|s_i^0)\prod_{h=1}^{H-1}p(s_i^h|s_i^{h-1},a_i^{h-1})\pi_{\boldsymbol{\theta}'_i}(a_i^h|s_i^h)\Big|\\
=&\rho(s_i^0)\prod_{h=1}^{H-1}p(s_i^h|s_i^{h-1},a_i^{h-1})\\
&\quad\cdot\Big|\prod_{h=0}^{H-1}\pi_{\boldsymbol{\theta}_i}(a_i^h|s_i^h)-\prod_{h=0}^{H-1}\pi_{\boldsymbol{\theta}'_i}(a_i^h|s_i^h)\Big|\\
=&\rho(s_i^0)\prod_{h=1}^{H-1}p(s_i^h|s_i^{h-1},a_i^{h-1})\underbrace{\Big|(\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i)^\top\nabla\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|}_{\textcircled{4}},
\end{aligned}\tag{58}$$

where the last equality uses the mean-value theorem, while $\hat{\boldsymbol{\theta}}:=(1-c)\boldsymbol{\theta}_i+c\boldsymbol{\theta}'_i$ is a certain convex combination of $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_i$ with $c\in[0,1]$. Further, term ④ in (58) satisfies

$$\begin{aligned}
&\Big|(\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i)^\top\nabla\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|\\
=&\Big|(\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i)^\top\nabla\log\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|\\
=&\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|(\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i)^\top\nabla\log\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|\\
\le&\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|\Big|\sum_{h=0}^{H-1}\nabla\log\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\Big|\Big|\,||\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i||\\
\le&HG||\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i||\prod_{h=0}^{H-1}\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h),
\end{aligned}\tag{59}$$

where the last inequality uses Assumption 2. Plugging (58) and (59) into (57) yields

$$\begin{aligned}
&\Big\|\Big\|\mathbb{E}_{\tau_i\sim p(\cdot|\pi_{\boldsymbol{\theta}_i})}\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\\
&\quad-\mathbb{E}_{\tau_i\sim p(\cdot|\pi_{\boldsymbol{\theta}'_i})}\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}'_i}\log\pi_{\boldsymbol{\theta}'_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\Big\|\Big\|\\
\le&\frac{HGr_{\max}}{1-\gamma}\int\rho(s_i^0)\pi_{\hat{\boldsymbol{\theta}}}(a_i^0|s_i^0)\prod_{h=1}^{H-1}p(s_i^h|s_i^{h-1},a_i^{h-1})\pi_{\hat{\boldsymbol{\theta}}}(a_i^h|s_i^h)\\
&\quad\cdot HG||\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i||\mathrm{d}\tau_i\\
=&\frac{H^2G^2r_{\max}}{1-\gamma}\int p(\tau_i|\hat{\boldsymbol{\theta}})||\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i||\mathrm{d}\tau_i\\
=&\frac{H^2G^2r_{\max}}{1-\gamma}||\boldsymbol{\theta}_i-\boldsymbol{\theta}'_i||.
\end{aligned}\tag{60}$$

For term ② in (56), we bound its norm as

$$\begin{aligned}
&\Big\|\Big\|\mathbb{E}_{\tau_i\sim p(\cdot|\boldsymbol{\theta}'_i)}\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\\
&\quad-\mathbb{E}_{\tau_i\sim p(\cdot|\boldsymbol{\theta}'_i)}\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}'_i}\log\pi_{\boldsymbol{\theta}'_i}(a_i^k|s_i^k)\Big)(\gamma)^h r_i^h\Big]\Big\|\Big\|\\
\le&\int p(\tau_i|\boldsymbol{\theta}'_i)\Big\|\Big\|\Big[\sum_{h=0}^{H-1}\Big(\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)\\
&\quad-\nabla_{\boldsymbol{\theta}'_i}\log\pi_{\boldsymbol{\theta}'_i}(a_i^k|s_i^k)\Big)\cdot(\gamma)^h r_i^h\Big]\Big\|\Big\|\mathrm{d}\tau_i\\
\le&r_{\max}\int p(\tau_i|\boldsymbol{\theta}'_i)\sum_{h=0}^{H-1}(\gamma)^h\\
&\quad\cdot\Big\|\Big\|\sum_{k=0}^{h}\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k)-\nabla_{\boldsymbol{\theta}'_i}\log\pi_{\boldsymbol{\theta}'_i}(a_i^k|s_i^k)\Big\|\Big\|\mathrm{d}\tau_i
\end{aligned}$$

$$
\begin{aligned}
&\le r_{\max} \int p(\tau_i|\boldsymbol{\theta}_i') \sum_{h=0}^{H-1} (\gamma)^h \\
&\quad \cdot \sum_{k=0}^{h} ||\nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i^k|s_i^k) - \nabla_{\boldsymbol{\theta}_i'} \log \pi_{\boldsymbol{\theta}_i'}(a_i^k|s_i^k)||\mathrm{d}\tau_i \\
&\overset{(i)}{\le} r_{\max} \int p(\tau_i|\boldsymbol{\theta}_i') \sum_{h=0}^{H-1} (\gamma)^h (h+1)M||\boldsymbol{\theta}_i - \boldsymbol{\theta}_i'||\mathrm{d}\tau_i \\
&\le HMr_{\max} \int p(\tau_i|\boldsymbol{\theta}_i') \sum_{h=0}^{H-1} (\gamma)^h ||\boldsymbol{\theta}_i - \boldsymbol{\theta}_i'||\mathrm{d}\tau_i \\
&\le \frac{HMr_{\max}}{1-\gamma} ||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||, \qquad (61)
\end{aligned}
$$

where (i) follows Assumption 2.

Combining (60) and (61) with (56) yields

$$
\begin{aligned}
&||\nabla J_i(\boldsymbol{\theta}_i) - \nabla J_i(\boldsymbol{\theta}_i')|| \qquad (62) \\
\le& \frac{H^2G^2 r_{\max}}{1-\gamma} ||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2|| \\
&+ \frac{HMr_{\max}}{1-\gamma} ||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2|| \\
=& \frac{(H^2G^2 + HM)r_{\max}}{1-\gamma} ||\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2||,
\end{aligned}
$$

which completes the proof.

APPENDIX F PROOF OF LEMMA 2

According to [25], CTM in (21) intrinsically yields a certain linear combination of honest messages, given by

$$
\boldsymbol{\theta}_{i,d}^{t+1} = \sum_{j \in \mathcal{H}_i \cup \{i\}} y_{j,d}^t \boldsymbol{\theta}_{j,d}^{t+\frac{1}{2}}, \qquad (63)
$$

where $\mathcal{H}_i$ is the set of honest neighbors of agent i, while the coefficients $y_{j,d}^t \in [0,1]$ and satisfy $\sum_{j \in \mathcal{H}_i \cup \{i\}} y_{j,d}^t = 1$. We also denote $\mathcal{B}_i = \mathcal{N}_i \backslash \mathcal{H}_i$ as the set of Byzantine neighbors of agent i. For the ease of subsequent theoretical analysis, we express the update above in a compact form as

$$
\check{\Theta}_d^{t+1} = Y^t(d) \check{\Theta}_d^{t+\frac{1}{2}}, \qquad (64)
$$

where $\check{\Theta}_d^{t+1}$ and $\check{\Theta}_d^{t+\frac{1}{2}}$ respectively stack $\boldsymbol{\theta}_{i,d}^{t+1}$ and $\boldsymbol{\theta}_{j,d}^{t+\frac{1}{2}}$ to $|\mathcal{H}| \times 1$ vectors, and $Y^t(d) \in \mathbb{R}^{|\mathcal{H}| \times |\mathcal{H}|}$ a row stochastic matrix. Defining $\nabla J(\check{\boldsymbol{\tau}}^t|\check{\Theta}^t)_d := [\nabla J(\tau_1^t|\boldsymbol{\theta}_1^t)_d; \ldots; \nabla J(\tau_{|\mathcal{H}|}^t|\boldsymbol{\theta}_{|\mathcal{H}|}^t)_d]$ $\in \mathbb{R}^{|\mathcal{H}|}$, we have

$$
\check{\Theta}_d^{t+1} = Y^t(d)[\check{\Theta}_d^t + \alpha^t \nabla J(\check{\boldsymbol{\tau}}^t|\check{\Theta}^t)_d]. \qquad (65)
$$

We can expand (65) to

$$
\begin{aligned}
\check{\Theta}_d^{t+1} =& Y^t(d) Y^{t-1}(d) \cdots Y^0(d) \check{\Theta}_d^0 \qquad (66) \\
&+ \sum_{v=0}^{t} \alpha^v Y^t(d) Y^{t-1}(d) \cdots Y^v(d) \nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d \\
=& \bar{Y}_0^t(d) \check{\Theta}_{[d]}^0 + \sum_{v=0}^{t} \alpha^v \bar{Y}_v^t(d) \nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d,
\end{aligned}
$$

where $\bar{Y}_a^b(d) := Y^b(d) Y^{b-1}(d) \cdots Y^a(d)$.

The limit of $\bar{Y}_v^t(d)$ has been established by Lemma 1 in [25]. We restate it as follows.

Lemma 3: If the row stochastic matrices $Y^t(d)$ are constructed as described in [25], the limit of $\bar{Y}_v^t(d)$ is given by

$$
\begin{aligned}
\lim_{t \to \infty} \bar{Y}_v^t(d) = \mathbf{1}\boldsymbol{p}_v^T(d), \qquad (67) \\
||\bar{Y}_v^t(d) - \mathbf{1}\boldsymbol{p}_v^\top(d)||^2 \le 4|\mathcal{H}|^2 \mu^{t-v+1},
\end{aligned}
$$

where $\boldsymbol{p}_v(d)$ is a stochastic vector whose elements are within $[0, \frac{1}{|\mathcal{N}_i|-2q_i+1}]$ and $\mu \in (0,1)$ increases monotonically as the network diameter $D_{\mathcal{G}}$ increases.

Motivated by Lemma 3 in [25], we define an auxiliary variable $\boldsymbol{\omega}$ as $\boldsymbol{\omega}_d^{t+1} = \boldsymbol{p}_0^\top(d)\Theta_d^0 + \sum_{v=0}^{t} \alpha^v \boldsymbol{p}_v^\top(d) \nabla J(\boldsymbol{\tau}^v|\Theta^v)_d$. Then, the consensus error can be bounded as

$$
\begin{aligned}
&\mathbb{E}\left[\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} ||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2\right] \qquad (68) \\
=& \mathbb{E}\left[\frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} ||\boldsymbol{\theta}_i^t - \boldsymbol{\omega}^t||^2 - ||\boldsymbol{\omega}^t - \check{\boldsymbol{\theta}}^t||^2\right] \\
\le& \mathbb{E}\left[\frac{1}{|\mathcal{H}|} ||\check{\Theta}^t - \hat{\Theta}^t||_F^2\right],
\end{aligned}
$$

where $\hat{\Theta}^t = \mathbf{1}(\boldsymbol{\omega}^t)^\top$. With (66), we have

$$
\begin{aligned}
&||\check{\Theta}^t - \hat{\Theta}^t||_F^2 \qquad (69) \\
=& \sum_{d=1}^{D} ||\check{\Theta}_{[d]}^t - \hat{\Theta}_d^t||^2 \\
\le& 2 \sum_{d=1}^{D} \underbrace{||(\bar{Y}_0^{t-1}(d) - \mathbf{1}\boldsymbol{p}_0^\top(d)) \check{\Theta}_{[d]}^0||^2}_{①} \\
&+ 2 \underbrace{\sum_{d=1}^{D} \sum_{v=0}^{t-1} \left|\left| \alpha^v (\bar{Y}_v^{t-1}(d) - \mathbf{1}\boldsymbol{p}_v^\top(d)) \nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_{[d]} \right|\right|^2}_{②}.
\end{aligned}
$$

Observe that $\bar{Y}_v^{t-1}(d)$ and $\mathbf{1}\boldsymbol{p}_v^\top(d)$ are both row stochastic, meaning that $(\bar{Y}_v^{t-1} - \mathbf{1}\boldsymbol{p}_v^\top)\mathbf{1} = 0$. Therefore, term ① in (69) is bounded as

$$
\begin{aligned}
&||(\bar{Y}_0^{t-1}(d) - \mathbf{1}\boldsymbol{p}_0^\top(d)) \check{\Theta}^0||^2 \qquad (70) \\
=& ||(\bar{Y}_0^{t-1}(d) - \mathbf{1}\boldsymbol{p}_0^\top(d))(\check{\Theta}_d^0 - \boldsymbol{\omega}_d^0 \mathbf{1})||^2 \\
\le& ||\bar{Y}_0^{t-1}(d) - \mathbf{1}\boldsymbol{p}_0^\top(d)||^2 \cdot \underbrace{||\check{\Theta}_d^0 - \boldsymbol{\omega}_d^0 \mathbf{1}||^2}_{=0} = 0,
\end{aligned}
$$

since $\check{\Theta}^0 = \hat{\Theta}^0 = \mathbf{1}(\boldsymbol{\omega}^0)^\top$.

As for term ② in (69), we have

$$\begin{aligned}
&\sum_{d=1}^{D} ||\alpha^v(\bar{Y}_v^{t-1}(d) - \mathbf{1}\boldsymbol{p}_v^\top(d))\nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d||^2 \qquad (71)\\
\leq &\sum_{d=1}^{D} ||\bar{Y}_v^{t-1}(d) - \mathbf{1}\boldsymbol{p}_v^\top(d)||^2 \cdot ||\alpha^v \nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d||^2\\
\leq &\sum_{d=1}^{D} 4|\mathcal{H}|^2\mu^{t-v}||\alpha^v \nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d||^2\\
= &4|\mathcal{H}|^2\mu^{t-v}(\alpha^v)^2\sum_{d=1}^{D} ||\nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d||^2\\
\overset{(i)}{\leq} &4|\mathcal{H}|^3C_g^2\mu^{t-v}(\alpha^v)^2,
\end{aligned}$$

where (i) uses Proposition 1. With (71), we have

$$\begin{aligned}
&\sum_{d=1}^{D}\sum_{v=0}^{t-1} \left|\left|\alpha^v(\bar{Y}_v^{t-1}(d) - \mathbf{1}\boldsymbol{p}_v^\top(d))\nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)_d\right|\right|^2 \qquad (72)\\
\leq &\sum_{v=0}^{t-1} 4|\mathcal{H}|^3C_g^2\mu^{t-v}(\alpha^v)^2\\
= &4|\mathcal{H}|^3C_g^2\sum_{v=0}^{t-1}\mu^{t-v}(\alpha^v)^2.
\end{aligned}$$

Now we introduce the following lemma whose proof can be found in [25].

Lemma 4: If the step size α^t satisfies

$$1 \leq \frac{\alpha^{t-1}}{\alpha^t} \leq \frac{2}{1+\eta_1}, \qquad (73)$$

and the iterates x^t satisfy

$$x^{t+1} \leq \eta_1 x^t + \eta_2(\alpha^t)^2 \quad \text{and} \quad x^0 \leq \eta_2(\alpha^0)^2, \qquad (74)$$

where $\eta_1 \in (0,1)$ and $\eta_2 \geq 0$, then x^k has an upper bound

$$x^t \leq \frac{2\eta_2}{1-\eta_1}(\alpha^t)^2 \qquad (75)$$

To bound (72), we define an auxiliary variable

$$x_1^t := \sum_{v=0}^{t-1}\mu^{t-v}(\alpha^v)^2, \qquad (76)$$

which satisfies $x^{t+1} = \mu x^t + \mu(\alpha^t)^2$. Using Lemma 4, we have

$$\sum_{v=0}^{t-1}\mu^{t-v}(\alpha^v)^2 = x^t \leq \frac{2\mu}{1-\mu}(\alpha^t)^2. \qquad (77)$$

Substituting (77) back to (72) yields

$$\begin{aligned}
&\left|\left|\sum_{v=0}^{t-1}\alpha^v(\bar{Y}_v^{t-1} - \mathbf{1}\boldsymbol{p}_v^\top)\nabla J(\check{\boldsymbol{\tau}}^v|\check{\Theta}^v)\right|\right|_F^2 \qquad (78)\\
\leq &\frac{8|\mathcal{H}|^3C_g^2\mu}{1-\mu}(\alpha^t)^2.
\end{aligned}$$

Thus, (69) can be bounded as

$$||\check{\Theta}^t - \hat{\Theta}^t||_F^2 \leq \frac{16|\mathcal{H}|^3C_g^2\mu}{1-\mu}(\alpha^t)^2, \qquad (79)$$

and consequently

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}||\boldsymbol{\theta}_i^t - \check{\boldsymbol{\theta}}^t||^2\right] \qquad (80)\\
\leq &\mathbb{E}\left[\frac{1}{|\mathcal{H}|}||\check{\Theta}^t - \hat{\Theta}^t||_F^2\right]\\
\leq &\frac{16|\mathcal{H}|^2C_g^2\mu}{1-\mu}(\alpha^t)^2.
\end{aligned}$$

References

[1] B. Lian, V. S. Donge, W. Xue, F. L. Lewis, and A. Davoudi, "Distributed minmax strategy for multiplayer games: Stability, robustness, and algorithms," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 3265–3277, 2024.

[2] M. Adibi and J. van der Woude, "Secondary frequency control of microgrids: An online reinforcement learning approach," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4824–4831, 2022.

[3] Z. Ni and S. Paul, "A multistage game in smart grid security: A reinforcement learning solution," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2684–2695, 2019.

[4] Y. Wu, S. Liao, X. Liu, Z. Li, and R. Lu, "Deep reinforcement learning on autonomous driving policy with auxiliary critic network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 7, pp. 3680–3690, 2023.

[5] C. Huang, R. Zhang, M. Ouyang, P. Wei, J. Lin, J. Su, and L. Lin, "Deductive reinforcement learning for visual autonomous urban driving navigation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5379–5391, 2021.

[6] A. S. Leong, A. Ramaswamy, D. E. Quevedo, H. Karl, and L. Shi, "Deep reinforcement learning for wireless sensor scheduling in cyber–physical systems," *Automatica*, vol. 113, p. 108759, 2020.

[7] T. Zhou, M. Chen, and J. Zou, "Reinforcement learning based data fusion method for multi-sensors," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 6, pp. 1489–1497, 2020.

[8] W. Zhu, X. Guo, D. Owaki, K. Kutsuzawa, and M. Hayashibe, "A survey of sim-to-real transfer techniques applied to reinforcement learning for bioinspired robots," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 7, pp. 3444–3459, 2023.

[9] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[10] Y. Li and D. Schuurmans, "Mapreduce for parallel reinforcement learning," in *European Conference on Recent Advances in Reinforcement Learning*, 2011, pp. 309–320.

[11] W. Caarls and E. Schuitema, "Parallel online temporal difference learning for motor control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 7, pp. 1457–1468, 2016.

[12] X. Fan, Y. Ma, Z. Dai, W. Jing, C. Tan, and B. K. H. Low, "Fault-tolerant federated reinforcement learning with theoretical guarantee," *Advances in Neural Information Processing Systems*, pp. 1007–1021, 2021.

[13] J. Xu, J. Wang, J. Rao, Y. Zhong, S. Wu, and Q. Sun, "Parallel cross entropy policy gradient adaptive dynamic programming for optimal tracking control of discrete-time nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–13, 2024.

[14] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[15] T. Liu, B. Tian, Y. Ai, L. Li, D. Cao, and F.-Y. Wang, "Parallel reinforcement learning: A framework and case study," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 4, pp. 827–835, 2018.

[16] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen *et al.*, "Massively parallel methods for deep reinforcement learning," *arXiv preprint arXiv:1507.04296*, 2015.

[17] T. Chen, K. Zhang, G. B. Giannakis, and T. Başar, "Communication-efficient policy gradient methods for distributed reinforcement learning," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 2, pp. 917–929, 2021.

[18] J. Chen, J. Sun, and G. Wang, "From unmanned systems to autonomous intelligent systems," *Engineering*, vol. 12, no. 5, pp. 16–19, 2022.

[19] X. Wang, J. Sun, G. Wang, F. Allgöwer, and J. Chen, "Data-driven control of distributed event-triggered network systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 2, pp. 351–364, 2023.

[20] A. Mathkar and V. S. Borkar, "Distributed reinforcement learning via gossip," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1465–1470, 2017.

[21] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Finite-sample analysis for decentralized batch multiagent reinforcement learning with networked agents," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5925–5940, 2021.

[22] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*, 2018, pp. 5650–5659.

[23] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.

[24] C. Xie, O. Koyejo, and I. Gupta, "Phocas: Dimensional Byzantine-resilient stochastic gradient descent," *arXiv preprint arXiv:1805.09682*, 2018.

[25] Z. Wu, H. Shen, T. Chen, and Q. Ling, "Byzantine-resilient decentralized policy evaluation with linear function approximation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3839–3853, 2021.

[26] J. Baxter and P. L. Bartlett, "Infinite-horizon policy-gradient estimation," *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, 2001.

[27] M. Grounds and D. Kudenko, "Parallel reinforcement learning with linear function approximation," in *International Joint Conference on Autonomous Agents and Multi-agent Systems*, 2007, pp. 1–3.

[28] Z. Chen and N. Li, "An optimal control-based distributed reinforcement learning framework for a class of non-convex objective functionals of the multi-agent network," *IEEE/CAA Journal of Automatica Sinica*, pp. 1–13, 2022.

[29] X. Sha, J. Zhang, K. You, K. Zhang, and T. Başar, "Fully asynchronous policy evaluation in distributed reinforcement learning over networks," *Automatica*, vol. 136, p. 110092, 2022.

[30] L. Huang, M. Fu, A. Rao, A. A. Irissappane, J. Zhang, and C. Xu, "A distributional perspective on multiagent cooperation with deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 3, pp. 4246–4259, 2024.

[31] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*, 2018, pp. 5872–5881.

[32] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning," in *International Conference on Machine Learning*, 2019, pp. 1626–1635.

[33] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 1544–1551.

[34] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for Byzantine robust optimization," in *International Conference on Machine Learning*, 2021, pp. 5311–5319.

[35] L. Su and N. H. Vaidya, "Byzantine-resilient multiagent optimization," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2227–2233, 2021.

[36] J. Peng, W. Li, and Q. Ling, "Byzantine-robust decentralized stochastic optimization over static and time-varying networks," *Signal Processing*, vol. 183, p. 108020, 2021.

[37] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust decentralized learning via self-centered clipping," *arXiv preprint arXiv:2202.01545*, 2022.

[38] N. Gupta and N. H. Vaidya, "Byzantine fault-tolerance in peer-to-peer distributed gradient-descent," *arXiv preprint arXiv:2101.12316*, 2021.

[39] S. Guo, T. Zhang, H. Yu, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Byzantine-resilient decentralized stochastic gradient descent," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 6, pp. 4096–4106, 2022.

[40] J. Xu and S.-L. Huang, "Byzantine-resilient decentralized collaborative learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 5253–5257.

[41] Z. Wu, T. Chen, and Q. Ling, "Byzantine-resilient decentralized stochastic optimization with robust aggregation rules," *IEEE Transactions on Signal Processing*, vol. 71, pp. 3179–3195, 2023.

[42] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *IEEE Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005, pp. 1653–1664.

[43] M. Papini, D. Binaghi, G. Canonaco, M. Pirotta, and M. Restelli, "Stochastic variance-reduced policy gradient," in *International Conference on Machine Learning*, 2018, pp. 4026–4035.

[44] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE transactions on Systems, Man, and Cybernetics*, no. 5, pp. 834–846, 1983.

[45] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[46] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[47] R. Wang, Y. Liu, and Q. Ling, "Byzantine-resilient resource allocation over decentralized networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4711–4726, 2022.