

# Adversarial attacks in consensus-based multi-agent reinforcement learning

Martin Figura, Krishna Chaitanya Kosaraju, and Vijay Gupta

**Abstract**—Recently, many cooperative distributed multi-agent reinforcement learning (MARL) algorithms have been proposed in the literature. In this work, we study the effect of adversarial attacks on a network that employs a consensus-based MARL algorithm. We show that an adversarial agent can persuade all the other agents in the network to implement policies that optimize an objective that it desires. In this sense, the standard consensus-based MARL algorithms are fragile to attacks.

## I. INTRODUCTION

Multi-agent reinforcement learning (MARL) is a branch of reinforcement learning (RL) [1], where multiple decision-makers learn a policy that is optimal in the context of competitive, cooperative, or mixed objectives [2], [3]. A recent success story of MARL in popular parlance include its performance in the video game Starcraft II by training an agent which outperforms the world’s best human players [4]. In this paper, we focus on MARL for cooperative agents. Potential applications in this stream have been proposed for long for diverse fields such as sensor networks [5], robotics [6], and traffic control [7].

Early formulations of MARL assumed that the agents share a common reward and focused on decentralized decision-making. The sample efficiency in the training of a multi-agent network in this setting is significantly improved by establishing communication between agents [8]. Nonetheless, the centralized reward is often infeasible due to overwhelming communication requirements and complex network topology, which motivated the development of cooperative distributed MARL with decentralized knowledge of rewards. In this setting, each agent has a local utility function and views only its own reward. The problem is still cooperative in the sense that the agents wish to maximize the sum of all utility functions. In this problem, the agents must communicate not only to improve the sample efficiency but also to become aware of the other agents’ performance. Only by propagating information over the entire network, the agents can achieve a common objective, e.g., to maximize team-average returns. The ability to learn a policy that maximizes the common objective in a partially observable environment can be facilitated by consensus algorithms as presented in [9], whereby the agent’s rewards (and possibly actions) remain unknown to the rest of the network and must not be directly communicated between agents to ensure their privacy.

Consensus algorithms are generally devised for distributed systems to find agreement on signal values over networks [10]. These algorithms find applications in many fields including sensor networks [11], coordination of vehicles [12], or even blockchain [13]. In practice, consensus algorithms must be robust to faults that arise from relatively frequent occurrences of interrupted communication links or corrupted signals [14]. Therefore, the convergence of resilient consensus algorithms was rigorously studied under different considerations for the nature of adversarial attacks [15], graph topology [16], [17], or frequency of communication [18]. These research efforts naturally complement studies of the influence of adversarial attacks on network performance. A simple yet powerful result from the analysis of linear consensus [11] states that the topology of a consensus matrix determines the limiting value for the consensus updates. In the presence of a single *malicious* agent, which does not apply consensus updates, the limiting value coincides with the adversary’s value.

In the consensus MARL algorithm in [9, Algorithm 2], every agent estimates the team-average reward and value function using linear approximations and exchanges parameters with other agents through a consensus protocol. Interestingly, this scheme guarantees the asymptotic convergence to the team-average optimal policy even with simultaneous actor, critic, and consensus updates over time-varying communication graphs. Furthermore, the algorithm retains the convergence property even with sparse data transmission for strongly connected graphs [19].

In this paper, we study the effects of adversarial attacks on the consensus MARL algorithm [9, Algorithm 2] with discounted rewards in the objective function. The attacks we consider are different from the commonly studied data poisoning attacks in ML or RL, which seek to understand if changing the data or rewards by an external agent can degrade the performance of RL algorithms [20]. Here, we consider a MARL setting where a participating agent itself is malicious. Specifically, we ask whether a single adversarial agent can either prevent convergence of the algorithm, or even worse, lead the other agents to optimize a utility function that it chooses. We show that the answer to this question is in the affirmative by designing a suitable attack and analyzing the convergence of the algorithm under it.

Specifically, we take under the scope networks with a single malicious agent, i.e., with an adversary that can compromise the consensus and critic updates and transmits the same signal values to its neighbors. We show that when the malicious agent greedily attempts to maximize its own well-

M. Figura, K. C. Kosaraju, and V. Gupta are with the Department of Electrical Engineering at the University of Notre Dame, Notre Dame, IN, 46556 USA, {mfigura, kkosaraj, vgupta2}@nd.edu.

defined objective function, all other agents in the network end up maximizing the adversary's objective function as well. We provide a proof of asymptotic convergence analogous to [9]. Our study motivates the development of resilient consensus MARL algorithms.

The paper is structured as follows. We provide a background of the networked Markov decision process along with the agents' objectives in Section 2. In Section 3, we state all assumptions in a compact form, present the consensus MARL algorithm, and provide the convergence analysis. Section 4 is dedicated to numerical simulations.

*Notation:* We let  $\mathbf{1}$  denote the vector of ones. The operator  $\otimes$  represents the Kronecker product. The cardinality of a set  $\mathcal{C}$  is denoted by  $|\mathcal{C}|$ .

## II. BACKGROUND

### A. Networked Markov decision process

We consider a networked Markov decision process (MDP) given as a tuple  $(\mathcal{S}, \{\mathcal{A}^i\}_{i \in \mathcal{N}}, \mathcal{P}, \{\mathcal{R}^i\}_{i \in \mathcal{N}}, \mathcal{G})$ , where  $\mathcal{N} = \{1, \dots, N\}$ ,  $\mathcal{S}$  is a set of states,  $\mathcal{P}$  is a set of transitional probabilities,  $\gamma \in [0, 1)$  is a discount factor,  $\mathcal{G}$  represents a graph, and  $\mathcal{A}^i$  and  $\mathcal{R}^i$  are a set of actions and rewards of agent  $i$ , respectively. The graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  is defined by a set of vertices  $\mathcal{N}$  associated with the agents in the network and a set of edges  $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ . The global state and action are denoted by  $s$  and  $a$ , respectively,  $s'$  denotes the state at the next time step, and the superscript  $i$  denotes a signal of agent  $i$ . We let  $r^i(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}_i \subset \mathbb{R}$  denote the individual reward of subsystem  $i$ ,  $p(s'|s, a) : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{P} \subset \mathbb{R}$  the joint transitional probability, and  $\pi^i(a^i|s) : \mathcal{S} \times \mathcal{A}_i \rightarrow (0, 1)$  the policy of subsystem  $i$ . The overall network policy can be written as a stacked vector of individual policies,  $\pi(a|s) = [\pi^1(a^1|s)^T, \dots, \pi^N(a^N|s)^T]^T$ . In case we need to explicitly specify a signal value at time  $t$ , we use subscript  $t$ , i.e.,  $r_{t+1}^i(s_t, a_t, s_{t+1})$ . An important consideration in this work is that agent  $i$ ,  $i \in \mathcal{N}$ , receives the private reward  $r_{t+1}^i$  along with the observation of the global state  $s_t$  and action  $a_t$  at each step in training.

We let  $\pi(a|s; \theta) = [\pi^1(a^1|s; \theta^1)^T, \dots, \pi^N(a^N|s; \theta^N)^T]^T$  denote the network policy parameterized by  $\theta$ , where  $\pi^i(a^i|s; \theta^i)$  is a parameterized policy of agent  $i$ . Further, we distinguish between reward signals by making the following definitions:

- 1) average individual reward at  $(s, a)$ :

$$\hat{r}^i(s, a) = \sum_{s'} p(s'|s, a) r^i(s, a, s')$$

- 2) average individual reward under global policy  $\pi(a|s; \theta)$ :

$$\hat{r}_\theta^i(s) = \sum_a \pi(a|s; \theta) \hat{r}^i(s, a)$$

- 3) average individual reward at all state-action pairs  $(s, a) \in \mathcal{S} \times \mathcal{A}$ :

$$\hat{R}^i = [\hat{r}^i(s, a), s \in \mathcal{S}, a \in \mathcal{A}]^T \in \mathbb{R}^{|\mathcal{S}| \cdot |\mathcal{A}|}$$

- 4) average individual reward under global policy  $\pi(a|s; \theta)$  at all states  $s \in \mathcal{S}$ :

$$\hat{R}_\theta^i = [\hat{r}_\theta^i(s), s \in \mathcal{S}]^T \in \mathbb{R}^{|\mathcal{S}|}$$

We also define team-average rewards  $r(s, a, s') = \frac{1}{N} \sum_{i=1}^N r^i(s, a, s')$ ,  $\hat{r}(s, a) = \frac{1}{N} \sum_{i=1}^N \hat{r}^i(s, a)$ ,  $\hat{r}_\theta(s) = \frac{1}{N} \sum_{i=1}^N \hat{r}_\theta^i(s)$ ,  $\hat{R}_\theta = \frac{1}{N} \sum_{i=1}^N \hat{R}_\theta^i \in \mathbb{R}^{|\mathcal{S}|}$ , and  $\hat{R} = \frac{1}{N} \sum_{i=1}^N \hat{R}^i$ . Furthermore, we define the estimated network reward function at  $(s, a)$  as  $\bar{r}(s, a; \lambda)$ , where  $\lambda$  are the function parameters.

### B. Objective

We let  $\mathcal{N}^+$  and  $\mathcal{N}^-$  denote the set of cooperative agents and adversaries, respectively, and note that  $\mathcal{N} = \mathcal{N}^+ \cup \mathcal{N}^-$ . The objective of agents  $i \in \mathcal{N}^+$  is to maximize a team-average objective function given as

$$\max_\theta J^+(\theta) = \max_\theta \mathbb{E} \left[ \sum_{t=0}^{\infty} \frac{1}{N} \sum_{i=1}^N \gamma^t r_{t+1}^i | s_0 = s \right]. \quad (1)$$

The cooperative agents are unaware of the presence of an adversarial agent that seeks to maximize a different objective function. We define the objective function for  $i \in \mathcal{N}^-$  as

$$\max_\theta J^-(\theta) = \max_\theta \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1}^i | s_0 = s \right]. \quad (2)$$

It is important to note that the adversarial agent can compromise the rewards  $r_{t+1}^i$ ,  $i \in \mathcal{N}^-$ , to incentivize its malicious behavior. Furthermore, once the agents establish communication the adversary can spread false information about the performance of the entire network embedded in the compromised rewards  $r_{t+1}^i$ . This may eventually lead to incentivizing a bad behavior in the cooperative agents as well, regardless of whether the maximized objective is (1) or (2). In Section III, we show that the entire network maximizes the adversarial agents' objective in (2) when the adversarial agent ignores signals transmitted by the cooperative agents.

## III. MULTI-AGENT ACTOR-CRITIC ALGORITHM UNDER ADVERSARIAL ATTACKS

In this section, we present assumptions on the network and signals, define the consensus MARL algorithm, state main theorems concerning the convergence of the actor and critic, and prove that the adversarial agent persuades the remaining agents in the network to maximize its individual objective in (2) despite their initial agreement to maximize the team objective in (1).

We formally define true action-value functions of the cooperative agents and the adversary under policy  $\pi(a|s)$  in the respective order as follows

$$Q_\pi^i(s, a) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \frac{1}{N} \gamma^t \sum_{k \in \mathcal{N}} r_{t+1}^k \right], i \in \mathcal{N}^+$$

$$Q_\pi^i(s, a) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r_{t+1}^i \right], i \in \mathcal{N}^-$$

where  $j \in \mathcal{N}^-$ . Further, we define the true state value functions as

$$V_\pi^i(s) = \sum_a \pi(a|s) Q_\pi^i(s, a), \quad i \in \mathcal{N}.$$

We will approximate  $V_\pi^i(s)$  using a parameterized state-value function  $V(s; v^i)$ .

**Remark.** It is required that all agents use the same basis functions (or neural networks) so that their parameter values  $v^i$  can eventually converge to a consensus value. This limitation can be overcome by considering gossip-based algorithms [21]. However, the convergence analysis for gossip-based algorithms is rather challenging.

#### A. Assumptions

In this subsection, we state assumptions needed for the convergence of the consensus MARL algorithm which we introduce later in the next section. The assumptions are similar to [9].

**Assumption 1.** The policy  $\pi^i(a^i|s; \theta^i) > 0$  for any  $i \in \mathcal{N}$ ,  $\theta^i \in \Theta^i$ ,  $s \in \mathcal{S}$ ,  $a^i \in \mathcal{A}^i$ . Also,  $\pi^i(a^i|s; \theta^i)$  is continuously differentiable with respect to  $\theta^i$ . For any  $\theta \in \Theta$ , we let  $P_\theta(s_{t+1}|s_t) = \sum_{a_t \in \mathcal{A}} P(s_{t+1}|s_t, a_t) \pi(a|s; \theta)$  denote the transition matrix of the Markov chain  $\{s_t\}_{t \geq 0}$  induced by policy  $\pi(a|s; \theta)$ . The Markov chain  $\{s_t\}_{t \geq 0}$  is irreducible and aperiodic under any  $\pi(a|s; \theta)$ .

**Assumption 2.** The update of the policy parameter  $\theta_t^i$  includes a local projection operator,  $\Gamma_i : \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{m_i}$ , that projects any  $\theta_t^i$  onto a compact set  $\Theta^i$ . Also, we assume that  $\Theta = \prod_i \Theta^i$  is large enough to include at least one local minimum of  $J(\theta)$ .

**Assumption 3.** The instantaneous reward  $r_{t+1}^i(s_t, a_t, s_{t+1})$  is uniformly bounded for any  $i \in \mathcal{N}$  and  $t \geq 0$ .

**Assumption 4.** The sequence of random matrices  $\{C_t\}_{t \geq 0} \subseteq \mathbb{R}^{N \times N} \subseteq \mathbb{R}^{N \times N}$  satisfies

- 1)  $C_t$  is row stochastic, i.e.,  $C_t \mathbf{1} = \mathbf{1}$ , and  $c_t(i, j) = 1$  for  $i = j \in \mathcal{N}^-$ . There exists a constant  $\eta \in (0, 1)$  such that, for any  $c_t(i, j) > 0$ , we have  $c_t(i, j) \geq \eta$ .
- 2)  $C_t$  respects the communication graph  $\mathcal{G}_t$ , i.e.,  $c_t(i, j) = 0$  if  $(i, j) \notin \mathcal{E}_t$ .
- 3) The spectral norm of  $\mathbb{E}[C_t^T(I - \mathbf{1}\mathbf{1}^T/N)C_t]$  belongs to  $[0, 1)$ .
- 4) Given the  $\sigma$ -algebra generated by the random variables before time  $t$ ,  $C_t$  is conditionally independent of  $r_{t+1}^i$  for any  $i \in \mathcal{N}$ .

**Assumption 5.** For each agent  $i$ , the state-value function and the team reward function are both parameterized by linear functions, i.e.,  $V(s; v) = v^T \varphi(s)$  and  $\bar{r}(s, a; \lambda) = \lambda^T f(s, a)$ , where  $\varphi(s) = [\varphi_1(s), \dots, \varphi_L(s)] \in \mathbb{R}^L$  and  $f(s, a) = [f_1(s, a), \dots, f_M(s, a)] \in \mathbb{R}^M$  are the features associated with  $s$  and  $(s, a)$ , respectively. The feature vectors  $\varphi(s)$  and  $f(s, a)$  are uniformly bounded for any  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}$ . Furthermore, let the feature matrix  $\Phi \in \mathbb{R}^{|\mathcal{S}| \times L}$  have  $[\varphi_l(s), s \in \mathcal{S}]^T$  as its  $l$ -th column for any  $l \in [L]$ , and the

feature matrix  $F \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{A}| \times M}$  have  $[f_m(s, a), s \in \mathcal{S}, a \in \mathcal{A}]^T$  as its  $m$ -th column for any  $m \in [M]$ . Both  $\Phi$  and  $F$  have full column rank.

**Assumption 6.** The step sizes  $\alpha_{v,t}$  and  $\alpha_{\theta,t}$  satisfy  $\sum_t \alpha_{v,t} = \infty$ ,  $\sum_t \alpha_{\theta,t} = \infty$ ,  $\sum_t \alpha_{v,t}^2 + \alpha_{\theta,t}^2 < \infty$ ,  $\alpha_{\theta,t} = o(\alpha_{v,t})$ , and  $\lim_{t \rightarrow \infty} \alpha_{v,t+1} \alpha_{v,t}^{-1} = 1$ .

**Assumption 7.** The set  $\mathcal{N}^-$  contains exactly one element, i.e., there is one malicious agent with a well-defined objective specified in (2).

We note that Assumption 4.3 is satisfied when the communication graph  $\mathcal{G}$  is connected in the mean. To simplify the convergence analysis in Section III.C, we assume that there is only one adversary that is learning using compromised rewards and does not perform consensus updates. The latter leads to unbalanced consensus updates in the entire network. We note that more general adversarial attacks are possible, e.g., there may be multiple adversarial agents in the network that may perform arbitrary parameter updates. Nonetheless, the narrow scope of attacks presented in this work is sufficient to demonstrate the fragility of the vanilla consensus MARL algorithm.

#### B. MARL algorithm

In this subsection, we introduce the consensus MARL algorithm. We let  $\Delta^i$  denote an estimated network TD error estimated by agent  $i$ . We noted earlier in Section III that every agent maintains parameters  $v^i$  which describe the network value function approximation  $V(s, v^i)$ . Further, we recall that the rewards  $r^i(s, a, s')$  remain private but the agents are allowed to estimate the network reward function. Intuitively, estimating the network reward function is a necessary step since the agents try to maximize the team-average objective in (1). We let  $d_\theta(s)$  denote a stationary distribution of the Markov chain  $\{s_t\}_{t \geq 0}$  under policy  $\pi(a|s; \theta)$ . If the rewards were mutually observable among the agents, they would minimize the weighted mean square error

$$\arg \min_{\lambda} \sum_{s \in \mathcal{S}, a \in \mathcal{A}} d_\theta(s) \pi(a|s; \theta) [\bar{r}(s, a; \lambda) - \hat{r}(s, a)]^2. \quad (3)$$

The optimization problem in (3) can be recast into a distributed optimization problem, which has the same stationary points, given as follows

$$\arg \min_{\lambda} \frac{1}{N} \sum_{i=1}^N \sum_{s \in \mathcal{S}, a \in \mathcal{A}} d_\theta(s) \pi(a|s; \theta) [\bar{r} - \hat{r}^i]^2. \quad (4)$$

since  $\frac{1}{N} \sum_i (\bar{r}(s, a; \lambda) - \hat{r}^i(s, a)) = \bar{r}(s, a; \lambda) - \hat{r}(s, a)$ . Hence, the agents can individually perform gradient steps to update the parameters  $\lambda^i$  based on their true rewards  $r^i(s, a, s')$ . By communicating via a consensus protocol, they further gain information about the encoded rewards of the other agents. The estimation and communication of the network reward function parameters  $\lambda^i$  and network value function parameters  $v^i$  provide the agents with the ability to update their policy to benefit the team. The consensus actor-critic algorithm, a version of [9, Algorithm 2] with

---

**Algorithm 1:** Consensus MARL algorithm

---

**Initialize parameters**  $\theta_0^i, \lambda_0^i, \tilde{\lambda}_0^i, v_0^i, \tilde{v}_0^i, \forall i \in \mathcal{N}$ ;  
**Initialize**  $s_0, \{\alpha_{v,t}\}_{t \geq 0}, \{\alpha_{\theta,t}\}_{t \geq 0}, t \leftarrow 0$ ;  
**Repeat until convergence**  
**for**  $i \in \mathcal{N}$  **do**  
  **Observe** state  $s_{t+1}$ , action  $a_t$ , and reward  $r_{t+1}^i$ ;  
  **Update**  
     $\tilde{\lambda}_t^i \leftarrow \lambda_t^i + \alpha_{v,t}(r_{t+1}^i - \bar{r}_{t+1}(\lambda_t^i)) \nabla_{\lambda} \bar{r}_{t+1}(\lambda_t^i)$ ;  
     $\delta_t^i \leftarrow r_{t+1}^i + \gamma V(s_{t+1}; v_t^i) - V(s_t; v_t^i)$ ;  
     $\Delta_t^i \leftarrow \bar{r}_{t+1}(\lambda_t^i) + \gamma V(s_{t+1}; v_t^i) - V(s_t; v_t^i)$ ;  
  **Update critic**  $\tilde{v}_t^i \leftarrow v_t^i + \alpha_{v,t} \delta_t^i \nabla_v V(s_t; v_t^i)$ ;  
  **Update actor**  $\theta_{t+1}^i \leftarrow \theta_t^i + \alpha_{\theta,t} \Delta_t^i \psi_t^i$ ;  
  **Send**  $\tilde{\lambda}_t^i, \tilde{v}_t^i$  to the neighbors over  $\mathcal{G}_t$ ;  
  **Take** action  $a_{t+1}^i \sim \pi^i(a_{t+1}^i | s_{t+1}; \theta_{t+1}^i)$ ;  
**end**  
**for**  $i \in \mathcal{N}$  **do**  
  **Consensus step**  
     $\lambda_{t+1}^i = \sum_{j \in \mathcal{N}} c_t(i, j) \tilde{\lambda}_t^j, v_{t+1}^i = \sum_j c_t(i, j) \tilde{v}_t^j$ ;  
**end**  
**Update** iteration counter  $t \leftarrow t + 1$

---

discounted returns is given in Algorithm 1. We note that the algorithm is the same for all agents but the adversary omits the consensus step. Furthermore, the action taken by the full network  $a_t$  can be assumed unobservable if the estimated rewards are independent of the actions, i.e.,  $\bar{r}(s, a; \lambda^i) = \bar{r}(s; \lambda^i)$ . In the following subsection, we provide the convergence analysis for Algorithm 1.

**Remark.** The scope of this work can be easily extended to [9, Algorithm 1], where agents approximate state-action value function parameters. For such an algorithm, the global action  $a_t$  must be observable by all agents in the network.

### C. Convergence analysis

In this subsection, we proceed with the convergence analysis. First, we show that the critic  $V(s; v^i)$  and network reward function  $\bar{r}(s, a; \lambda^i)$  converge to a fixed point for all  $i \in \mathcal{N}$  while the policy  $\pi(a|s; \theta)$  remains fixed. Then, we prove the full convergence of the actor updates that occur on a slower timescale. We write the stationary distribution of Markov chain  $d_\theta(s)$  for all states  $s \in \mathcal{S}$  as a matrix  $D_\theta^s = \text{diag}[d_\theta(s), s \in \mathcal{S}]$ . Similarly, we write the distribution of all state-action pairs  $(s, a)$  as a matrix  $D_\theta^{s,a} = \text{diag}[d_\theta(s) \cdot \pi(a|s; \theta), s \in \mathcal{S}, a \in \mathcal{A}]$ . For brevity, we use shorthands  $\varphi_t = \varphi(s_t)$  and  $f_t = f(s_t, a_t)$ . Finally, we define the consensus value  $\langle z_t \rangle = \frac{1}{N} \sum_i z_t^i$  and the disagreement vector  $z_{\perp,t} = z_t - \mathbf{1} \otimes \langle z_t \rangle$ .

**Theorem 1.** Under assumptions 1 and 3-7, for any policy  $\pi(a|s; \theta)$ , with the updates of  $\{v_t^i\}$  in Algorithm 1, we have  $\lim_t v_t^i = v_\theta$  and  $\lim_t \lambda_t^i = \lambda_\theta$  for  $i \in \mathcal{N}$  almost surely. Furthermore,  $v_\theta$  and  $\lambda_\theta$  are the unique solutions to

$$F^T D_\theta^{s,a} (\hat{R}^j - F \lambda_\theta) = 0 \quad (5)$$

$$\Phi^T D_\theta^s (\hat{R}_\theta^j + \gamma P_\theta \Phi v_\theta - \Phi v_\theta) = 0, \quad (6)$$

where  $j \in \mathcal{N}^-$ .

*Proof.* We let  $z_t = [(z_t^1)^T, \dots, (z_t^N)^T]^T \in \mathbb{R}^{(M+L)N}$ , where  $z_t^i = [(\lambda_t^i)^T, (v_t^i)^T]^T$ . Furthermore, we define  $b_t = r_{t+1} \otimes [f_t^T \ \phi_t^T]^T$  and  $A_t = I \otimes A_t'$ , where  $A_t' \begin{bmatrix} -f_t f_t^T & 0 \\ 0 & \phi_t(\gamma \phi_{t+1} - \phi_t)^T \end{bmatrix}$ . We let  $\mathcal{F}_t^z = \{z_0, Y_\tau, \tau \leq t\}$  denote a filtration where  $Y_\tau = \{s_\tau, a_\tau, r_\tau, C_{\tau-1}\}$  is a collection of random variables. The iterations of Algorithm 1 can be written in a compact form as follows

$$\begin{aligned} z_{t+1} &= (C_t \otimes I)(z_t + \alpha_{v,t}(A_t z_t + b_t)) \\ &= (C_t \otimes I)[z_t + \alpha_{v,t}(h(z_t, Y_t) + M_{t+1})] \end{aligned}$$

where  $h(z_t, Y_t) = \mathbb{E}(A_t z_t + b_t | \mathcal{F}_t^z)$  and  $M_t = A_t z_t + b_t - \mathbb{E}(A_t z_t + b_t | \mathcal{F}_t^z)$ . To prove Theorem 1, we need to show that

- 1) **Lemma 1:** the parameters  $\lambda_t$  and  $v_t$  remain bounded for all  $t \geq 0$ ,
- 2) **Lemma 2:** the adversary's parameters asymptotically converge, i.e.,  $\lambda_t^j \rightarrow \lambda_\theta$  and  $v_t^j \rightarrow v_\theta, j \in \mathcal{N}^-$ ,
- 3) **Lemma 3:** the agents' parameters asymptotically converge to the consensus value  $\langle \lambda_t \rangle$  and  $\langle v_t \rangle$ .

We take advantage of the rich convergence analysis in [9] to prove the lemmas.

**Lemma 1.** Under assumptions 1 and 3-6, the sequence  $\{z_t\}$  satisfies  $\sup_t \|z_t\| < \infty$  almost surely.

*Proof.* The proof is given in [9, Appendix C]. The only difference in our work is that in the absence of the consensus step the updates of  $z_t^i, i \in \mathcal{N}$ , asymptotically follow the ODE  $\dot{z}_t^i = \bar{A}_t' z_t^i + \bar{b}_t^i$  where

$$\bar{A}_t' = \begin{bmatrix} -F^T D_\theta^{s,a} F & 0 \\ 0 & \Phi^T D_\theta^s (\gamma P_\theta - I) \Phi \end{bmatrix} \quad (7)$$

$$\bar{b}_t^i = [(F^T D_\theta^{s,a} \hat{R}^i)^T \ (\Phi^T D_\theta^s \hat{R}_\theta^i)^T]^T. \quad (8)$$

The discount factor satisfies  $\gamma \in [0, 1)$  and the stochastic matrix  $P_\theta$  has positive eigenvalues that are less than or equal to 1. Therefore, the matrix  $\Phi^T D_\theta^s (\gamma P_\theta - I) \Phi$  has eigenvalues with strictly negative real parts, which implies that the ODE  $\dot{z}_t^i = \bar{A}_t' z_t^i + \bar{b}_t^i$  has an asymptotically stable equilibrium. Hence,  $\sup_t \|z_t\| < \infty$  almost surely.  $\square$

**Lemma 2.** Under assumptions 1, 3, and 5-7,  $\lim_{t \rightarrow \infty} z_t^j = z_\theta, j \in \mathcal{N}^-$ , almost surely. Furthermore,  $z_\theta = [\lambda_\theta^T, v_\theta^T]^T$  is a unique solution to (5) and (6).

*Proof.* We recall that the adversarial agent does not perform the consensus step. Using the findings in Lemma 1, we can immediately conclude that  $\dot{z}_t^j = \bar{A}_t' z_t^j + \bar{b}_t^j$  is the limiting ODE, with  $\bar{A}_t'$  given in (7) and  $\bar{b}_t^j = [(F^T D_\theta^{s,a} \hat{R}^j)^T \ (\Phi^T D_\theta^s \hat{R}_\theta^j)^T]^T$ . The ODE has a unique asymptotically stable equilibrium  $z_\theta = [\lambda_\theta^T, v_\theta^T]^T$  that satisfies (5) and (6).  $\square$

**Lemma 3** (Appendix B.4, Step 1 in [9]). Under assumptions 1 and 3-7, the disagreement vector  $z_{\perp,t}$  satisfies  $\lim_{t \rightarrow \infty} z_{\perp,t} = 0$  almost surely.

To complete the proof of Theorem 1, we recall that

- $\lim_{t \rightarrow \infty} (z_t^j - z_\theta) = 0$  for  $j \in \mathcal{N}^-$  a.s. (Lemma 2)
- $\lim_{t \rightarrow \infty} (z_t^i - \langle z_t \rangle) = 0$  for  $i \in \mathcal{N}$  a.s. (Lemma 3).

Therefore,  $\lim_{t \rightarrow \infty} (\langle z_t \rangle - z_\theta) = 0$  almost surely where  $z_\theta = [\lambda_\theta^T, v_\theta^T]^T$  satisfies (5) and (6).  $\square$

Having proved the critic and network reward convergence under a fixed policy  $\pi(a|s; \theta)$ , we proceed to make a statement about the convergence of the actor updates on the slower timescale.

**Theorem 2.** [9, Theorem 4.10] Under assumptions 1-7, the policy parameter  $\theta_i^i$  converges almost surely to a point in the set of asymptotically stable equilibria of

$$\dot{\theta}^i = \hat{\Gamma}^i [\mathbb{E}_{s_t \sim d_\theta, a_t \sim \pi_\theta} (\Delta_{t,\theta} \cdot \psi_{t,\theta}^i)] \quad \text{for } i \in \mathcal{N}, \quad (9)$$

where  $\mathbb{E}_{s_t \sim d_\theta, a_t \sim \pi_\theta} (\Delta_{t,\theta} \cdot \psi_{t,\theta}^i) = \mathbb{E}_{s_t \sim d_\theta, a_t \sim \pi_\theta} [(f_t^T \lambda_\theta + \gamma \varphi_{t+1} v_\theta - \varphi_t v_\theta) \nabla_{\theta^i} \log \pi^i(a_t^i | s_t; \theta^i)]$ .

We note that the policy  $\pi(a|s; \theta)$  converges to an equilibrium where the estimated network TD error  $\Delta_{t,\theta}$  is equal to zero. Since  $\Delta_{t,\theta}$  is a function of the parameterized network reward function  $\bar{r}(s, a; \lambda)$  and value function  $V(s; v)$ , the policy  $\pi(a|s; \theta)$  does not converge to the true optimal policy. The error between  $\pi(a|s; \theta)$  and the true optimal policy that maximizes (2) can be reduced by selecting appropriate models for  $\bar{r}(s, a; \lambda)$  and  $V(s; v)$ .

In the next section, we present an example in which a group of agents employs Algorithm 1 to learn an optimal policy and is subject to a malicious attack from a single adversarial agent.

#### IV. NUMERICAL SIMULATIONS

In this section, we assess the performance of Algorithm 1 through numerical simulations using nonlinear function approximation. We also compare our results against the de-

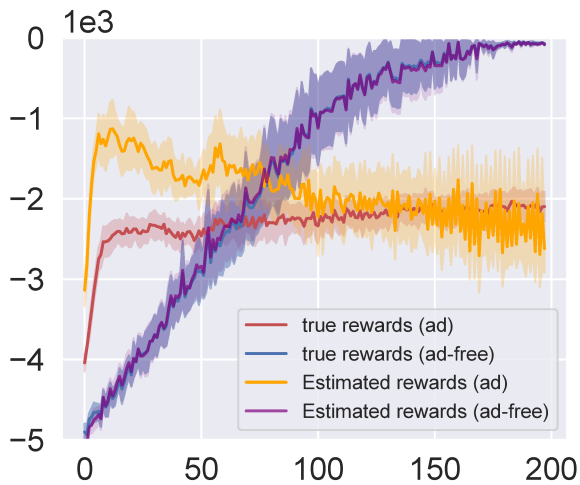


Fig. 1. Cumulative team-average rewards per episode for the adversary-free and attacked network. The comparison shows that the former performs significantly better

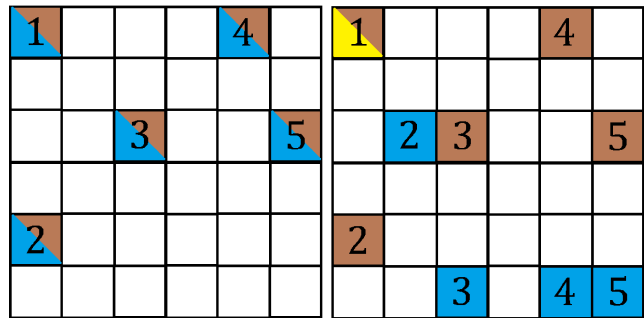


Fig. 2. Final network states in a simulation with no adversary (left) and with an adversary (right) after training for 200 episodes. Blue, yellow, and brown cells correspond to the cooperative agents', adversary's, and desired positions, respectively. All agents reach their desired positions when the network is adversary-free, whereas only the adversary finds its desired position when it attacks the network.

centralized actor-critic algorithm in [9]. The code for these experiments can be found in [22].

We consider a network of agents  $\mathcal{N} = \{1, 2, 3, 4, 5\}$  in a grid-world of dimension  $(6 \times 6)$ . The position of agent  $i$  is described by the tuple  $(x_i, y_i) \in S^i$ , where  $S^i = [0, \dots, 5]^2$ . We note that the tuples  $(0, 0)$  and  $(5, 5)$  correspond to the top-left and bottom-right corners of the grid, respectively. The state of the grid-world is given as  $s = [(x^i, y^i), i \in \mathcal{N}] \in \mathcal{S}$  where  $\mathcal{S} = \mathcal{S}^1 \times \dots \times \mathcal{S}^5$ . The cardinality of  $\mathcal{S}$  is  $|\mathcal{S}| = 36^5 (\approx 60.5 \text{ million states})$ . Agent  $i$ ,  $i \in \mathcal{N}$ , takes actions from the set  $\mathcal{A}^i = \{0 : Left, 1 : Right, 2 : Up, 3 : Down, 4 : Stay\}$ . If an action is to bring the agent to an infeasible state, then the agent remains in the same state. The set of actions of the network is given as  $\mathcal{A} = \mathcal{A}^1 \times \dots \times \mathcal{A}^5$ , whose cardinality is  $|\mathcal{A}| = 5^5$ .

The goal of each cooperative agent,  $i \in \mathcal{N}^+ = \{2, 3, 4, 5\}$ , is to maximize the objective in (1), whereas the adversary  $i \in \mathcal{N}^- = \{1\}$  attempts to maximize (2). The rewards of agent  $i$ ,  $i \in \mathcal{N}$ , are given as follows

$$r^i(s^i) = -|x^i - x_{des}^i| - |y^i - y_{des}^i| - q^i,$$

where  $q^i$  denotes the number of neighboring agents that agent  $i$  collides at the current time step. For simplicity, we consider that the communication graph  $\mathcal{G}$  is fully connected and the consensus matrix  $C$  in the adversary-free scenario has elements  $c(i, j) = 1/5$  for  $i \in \mathcal{N}^+$ ,  $j \in \mathcal{N}$ .

In both scenarios, we trained the agents for 200 episodes. Final states of a simulation of the grid-world after training are shown in Fig. 2. The agents' positions in the grid-world were randomly initialized in each training episode that was set to terminate after 1000 steps or when the agents have reached their desired positions. The actor  $\pi^i(a^i | s, \theta^i)$ , critic  $V(s, v^i)$ , and global reward functions  $\bar{r}(s, \lambda^i)$  were approximated using artificial neural networks with two hidden layers. In Fig. 1, we compare the true cumulative team-average returns and cumulative estimated rewards obtained by the network in each episode. The estimated reward function converged in both scenarios but the convergence rate was slower in the presence of the adversary. The accumulated rewards per

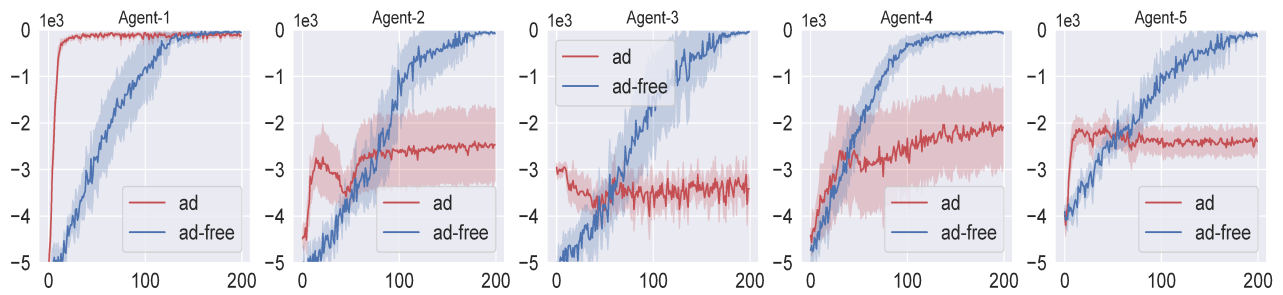


Fig. 3. True cumulative rewards per episode obtained by each agent in the network. Blue and red color depict the performance of adversary-free network and the attacked network, respectively. The adversary quickly learns an optimal policy because it acts greedily with respect to the rest of the network.

episode of each agent are depicted in Fig. 3. We can see that all agents learn a near-optimal policy when there is no adversary in the network. The adversary indeed harms the network, i.e., it learns a near-optimal policy but the remaining agents in the network perform poorly compared to the adversary-free scenario.

## V. CONCLUSION AND FUTURE WORK

In this paper, we showed that the general consensus MARL algorithm originally proposed in [9] is not robust to adversarial attacks. We studied a well-defined malicious attack whereby a single adversarial agent attempts to compromise the objective function of a network of agents. We showed in the analysis that the network policy upon convergence locally maximizes the adversary’s objective function under this specific malicious attack. Our work naturally raises the question of whether we can develop consensus-based MARL algorithms that are resilient to general adversarial attacks. Such attacks may include compromised rewards, arbitrary parameter updates, and arbitrary changes in the policy. There are many results on resilient consensus algorithms in the literature but it is unclear if the theoretical analysis can carry over to RL algorithms. The unique challenge for resilient consensus MARL is to provide robustness for the functions jointly estimated by the network of agents while the rewards remain private.

## REFERENCES

- [1] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning*, vol. 135. MIT press Cambridge, 1998.
- [2] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” 2019.
- [3] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews*, 38 (2), 2008, 2008.
- [4] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.
- [5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.
- [6] E. Yang and D. Gu, “Multiagent reinforcement learning for multi-robot systems: A survey,” tech. rep., tech. rep, 2004.
- [7] L. Kuyer, S. Whiteson, B. Bakker, and N. Vlassis, “Multiagent reinforcement learning for urban traffic control using coordination graphs,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 656–671, Springer, 2008.
- [8] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, “Learning to communicate with deep multi-agent reinforcement learning,” in *Advances in neural information processing systems*, pp. 2137–2145, 2016.
- [9] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, “Fully decentralized multi-agent reinforcement learning with networked agents,” *arXiv preprint arXiv:1802.08757*, 2018.
- [10] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [11] R. Olfati-Saber and J. S. Shamma, “Consensus filters for sensor networks and distributed sensor fusion,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 6698–6703, IEEE, 2005.
- [12] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” *IEEE Control systems magazine*, vol. 27, no. 2, pp. 71–82, 2007.
- [13] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, and C. Qijun, “A review on consensus algorithm of blockchain,” in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2567–2572, IEEE, 2017.
- [14] M. J. Fischer, “The consensus problem in unreliable distributed systems (a brief survey),” in *International conference on fundamentals of computation theory*, pp. 127–140, Springer, 1983.
- [15] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, “Resilient asymptotic consensus in robust networks,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.
- [16] D. Saldana, A. Prorok, S. Sundaram, M. F. Campos, and V. Kumar, “Resilient consensus for time-varying networks of dynamic agents,” in *2017 American control conference (ACC)*, pp. 252–258, IEEE, 2017.
- [17] S. Sundaram and C. N. Hadjicostis, “Finite-time distributed consensus in graphs with time-invariant topologies,” in *2007 American Control Conference*, pp. 711–716, IEEE, 2007.
- [18] D. Ding, Z. Wang, D. W. Ho, and G. Wei, “Observer-based event-triggering consensus control for multiagent systems with lossy sensors and cyber-attacks,” *IEEE transactions on cybernetics*, vol. 47, no. 8, pp. 1936–1947, 2016.
- [19] Y. Lin, K. Zhang, Z. Yang, Z. Wang, T. Başar, R. Sandhu, and J. Liu, “A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5562–5567, IEEE, 2019.
- [20] Y. Ma, X. Zhang, W. Sun, and J. Zhu, “Policy poisoning in batch reinforcement learning and control,” in *Advances in Neural Information Processing Systems*, pp. 14570–14580, 2019.
- [21] A. Mathkar and V. S. Borkar, “Distributed reinforcement learning via gossip,” *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1465–1470, 2016.
- [22] K. C. Kosaraju, M. Figura, and V. Gupta, “Adversarial - multi-agent reinforcement learning (adv-marl).” <https://github.com/asokraj/adv-marl>, 2020.