What is the Solution for State-Adversarial Multi-Agent Reinforcement Learning?

Songyang Han[1], **Sanbao Su**[1], **Sihong He**[1], **Shuo Han**[2], **Haizhao Yang**[3], **Fei Miao**[1]

[1] University of Connecticut, [2] University of Illinois at Chicago, [3] University of Marryland, College Park
{songyang.han, sanbao.su, sihong.he, fei.miao}@uconn.edu, hanshuo@uic.edu, hzyang@umd.edu

Abstract

Various types of Multi-Agent Reinforcement Learning (MARL) methods have been developed, assuming that agents' policies are based on true states. Recent works have improved the robustness of MARL under uncertainties from the reward, transition probability, or other partners' policies. However, in real-world multi-agent systems, state estimations may be perturbed by sensor measurement noise or even adversaries. Agents' policies trained with only true state information will deviate from optimal solutions when facing adversarial state perturbations during execution. MARL under adversarial state perturbations has limited study. Hence, in this work, we propose a State-Adversarial Markov Game (SAMG) and make the first attempt to study the fundamental properties of MARL under state uncertainties. We prove that the optimal agent policy and the robust Nash equilibrium do not always exist for an SAMG. Instead, we define the solution concept, robust agent policy, of the proposed SAMG under adversarial state perturbations, where agents want to maximize the worst-case expected state value. We then design a gradient descent ascent-based robust MARL algorithm to learn the robust policies for the MARL agents. Our experiments show that adversarial state perturbations decrease agents' rewards for several baselines from the existing literature, while our algorithm outperforms baselines with state perturbations and significantly improves the robustness of the MARL policies under state uncertainties.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has been studied and successfully applied to solve problems such as multi-robot coordination (Hüttenrauch and Šošić 2017), multi-agent communication (Sukhbaatar, Szlam, and Fergus 2016), sequential social dilemmas (Leibo and Zambaldi 2017), resource management (Pretorius, Cameron et al. 2020), etc. The non-stationary training environment issue is addressed by using an actor-critic framework in (Lowe et al. 2017; Foerster and Farquhar 2018). However, Deep Reinforcement Learning (DRL) policy is highly vulnerable under adversarial state perturbation attacks (Behzadan and Munir 2017; Pattanaik and Tang 2017; Xiao, Pan et al. 2019; Huang et al. 2017; Lin et al. 2017). Even a slight state perturbation can lead the DRL policy to a completely different action (Huang et al. 2017; Lin et al. 2017). This is a significant issue, especially for safety-critical systems like autonomous vehicles, e.g. adding paint in an adversarial way to the surface of a road can confuse an autonomous vehicle's lane-following policy and cause potential accidents (Kurakin, Goodfellow, and Bengio 2016). Therefore, it is important to have a robust policy: a well-trained agent should behave well under adversarial state perturbations.

Additionally, in real-world applications, agents may not have the perfect state information due to sensor measurement noise, poor weather, or illumination conditions (Liu et al. 2021; Kothandaraman, Chandra, and Manocha 2021). For agents trained in simulation with true state information, the policy may have poor performance in practice, which is usually known as sim-to-real gaps (Jiang et al. 2021; Sallab et al. 2017; Han et al. 2022). Perturbed or adversarial observation issues have been investigated in single-agent DRL to narrow the gap between the simulation and the real world (Pinto, Davidson, and Sukthankar 2017; Zhang et al. 2020a) or to defend against adversarial attacks (Xiao, Pan et al. 2019; Pattanaik and Tang 2017; Pattanaik et al. 2018; Mandlekar et al. 2017). The adversarial state perturbation problem cannot be characterized by the existing research in the partially observable Markov decision process (POMDP) or Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek, Amato et al. 2016; Lerer et al. 2020), as the conditional observation probability cannot capture the worst-case state uncertainty under adversarial attacks. Adversarial perturbations have a much more powerful impact on an agent's policy than random noise (Kos and Song 2017; Pattanaik et al. 2018). However, it remains challenging to deal with adversarial state perturbations in MARL, as the problem is more complicated considering the interaction among all the agents and the adversaries. Therefore, it is crucial to study the fundamental properties of MARL under adversarial state perturbations.

In this work, we make the first attempt to study the challenges and fundamental properties of robust MARL under adversarial state perturbations. We formulate a State-Adversarial Markov Game (SAMG) to study the properties and solution concepts of MARL under adversarial state perturbations. We prove that the optimal agent policy or robust Nash equilibrium does not always exist under adversarial state perturbations. Instead, we propose a new solution concept, the robust agent policy, and prove that the robust agent policy exists for finite state and finite action

spaces. We design an algorithm, called Robust Multi-Agent Reinforcement Learning (RMARL), to train robust policies of all agents with adversarial state perturbations. The algorithm uses Gradient Descent Ascent (GDA) optimizer (Lin, Jin, and Jordan 2020) to update each agent's policy network and each adversary's policy network. Our experiment results show that the proposed RMARL algorithm improves the robustness of the agents' policies compared with the existing MARL literature. To the best of our knowledge, we propose the first mathematical formulation and algorithm that considers state perturbations in MARL with both theoretical and empirical justifications.

In summary, the main contributions of this work are:

- We formulate a State-Adversarial Markov Game (SAMG) to study the fundamental properties of MARL under adversarial state perturbations. We prove that the classical solution concept such as optimal agent policy or robust Nash equilibrium does not always exist.
- Instead, we define a new solution concept, robust agent policy, to maximize the worst-case expected state value. We prove the existence of the robust agent policy for an SAMG with finite state and finite action spaces. We propose a Robust Multi-Agent Reinforcement Learning (RMARL) algorithm to solve the challenge of training robust policies under adversarial state perturbations.
- We empirically evaluate our proposed RMARL algorithm in experiments. Our algorithm outperforms baselines with random or adversarial state perturbations and improves agent policies' robustness under state uncertainties.

The rest of this paper is organized as follows. We introduce related work in Section 2. In Section 3, we define the problem formulation of the SAMG. The solution concepts are studied for agents and adversaries of the SAMG in Section 4. In Section 5, we propose an algorithm for finding a robust policy for the MARL agents. The experiments are shown in Section 6. In Appendix, we provide detailed proof, implementation details, and discussions about future work.

2 Related Work

Multi-Agent Reinforcement Learning (MARL) The MARL has a long history in the AI field (Hu, Wellman et al. 1998; Busoniu, Babuska, and De Schutter 2008). Recent works have been investigated to encourage the collaboration of the agents by assigning rewards appropriately, such as a value decomposition network (Sunehag, Lever et al. 2018; Rashid et al. 2020; Su, Adams, and Beling 2021), subtracting a counterfactual baseline (Foerster and Farquhar 2018), or an implicit method (Zhou et al. 2020). Multi-Agent Deep Deterministic Policy Gradient (MADDPG) proposes a centralized Q-function to alleviate the problem caused by the non-stationary environment (Lowe et al. 2017). The scalability issue of MARL can be alleviated by adding attention to the critic (Iqbal and Sha 2019), using neighbor information (Qu et al. 2020), or using V-learning (Jin et al. 2021). The "team stochastic game" (Muniraj, Vamvoudakis, and Farhood 2018) splits the MARL agents into two teams to compete. However, during training, all methods assume that agents get the true state value. None of the recent MARL advances specifies how to deal with perturbed state values by malicious adversaries.

Robust Reinforcement Learning Most existing robust MARL works focus on uncertainties in reward, transition dynamics, and training partners' policies, while our work focus on uncertainties in the state. Robust reinforcement learning can be traced back to (Morimoto and Doya 2005) in the single-agent setting. Combined with deep learning techniques, the robust MARL is recently studied considering different types of uncertainties in reward (Zhang et al. 2020b), transition dynamics (Zhang et al. 2020b; Sinha, O'Kelly et al. 2020; Hu et al. 2020; Yu et al. 2021b), training partner's type (Shen and How 2021), and training partners' policies (Li et al. 2019; van der Heiden et al. 2020; Sun, Kim, and How 2021). The work in (Zhang et al. 2020b) considers the robust equilibrium of multi-agents with reward uncertainties where agents can get true state information at each stage. The work in (Shen and How 2021) considers uncertain training partner's type (e.g. adversary, neutral, or teammate) to the protagonist in two-player scenarios. The M3DDPG algorithm extends the MADDPG to get a robust policy for the worst situation by assuming all the training partners are adversaries (Li et al. 2019). However, none of the above MARL works consider the state perturbations.

There are some works (Mandlekar et al. 2017; Pattanaik et al. 2018; Zhang et al. 2020a, 2021) considering the robust policy under adversarial state perturbations in single-agent reinforcement learning. Though the work (Lin et al. 2020) studies state perturbation, only one single agent's state observation can be perturbed in their MARL. In this work, we focus on the more challenging multi-agent cases where each agent's state observation can be perturbed by an adversary.

3 State-Adversarial Markov Game (SAMG)

We formulate a State-Adversarial Markov Game (SAMG) with n agents in the agent set $\mathcal{N} = \{1, ..., n\}$. Each agent i is associated with an action $a^i \in \mathcal{A}^i$. The global joint action is $a = (a^1, ..., a^n) \in \mathcal{A}$, $\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$. The global joint state is $s \in \mathcal{S}$. All agents share a stage-wise reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The total expected return is $\sum_{t=0}^{\infty} \gamma^t r_{t+1}(s_t, a_t)$ where γ is a discount factor. We consider that each agent is associated with an adversary as shown in Fig. 1. Each adversary decides a perturbed state $\rho^i \in \mathcal{S}$ for the corresponding agent as the agent's perturbed knowledge or observation about the global state. We denote the joint perturbed state as $\rho := [\rho^i]_{i \in \mathcal{N}}$. We consider the admissible perturbed state as a task-specific "neighboring" state of s, e.g. the bounded sensor measurement errors, to model the real-world challenges of getting accurate states for multi-agent systems like connected and autonomous vehicles and multi-robots systems (Liu et al. 2021; Kothandaraman, Chandra, and Manocha 2021). To analyze a realistic problem, the power of the state perturbation should also be limited (Everett, Lütjens, and How 2021; Zhang et al. 2020a). We define an admissible perturbed state set $\mathcal{P}_s$ to restrict the perturbed state only to be within a predefined subset of states such that $\rho \in \mathcal{P}_s$:

Definition 1 (**Admissible Perturbed State Set**). *We consider the set of admissible perturbed state for agent i at state s as $\mathcal{P}_s^i \subseteq \mathcal{S}$. Denote the joint admissible perturbed state set at state s as $\mathcal{P}_s := \mathcal{P}_s^1 \times \cdots \times \mathcal{P}_s^n$.*

Note that the true state is included in the admissible perturbed state set, i.e., $s \in \mathcal{P}_s^i$ for any $i \in \mathcal{N}$. For example, consider a 2-agent 3-state system with $\mathcal{S} = \{s_1, s_2, s_3\}$. When the current true state is s_1 for both agents, adversary 1 perturbs agent 1's state observation within $\mathcal{P}_{s_1}^1 = \{s_1, s_2\}$; adversary 2 perturbs agent 2's state observation within $\mathcal{P}_{s_1}^2 = \{s_1, s_3\}$.

The state perturbation reflects the state uncertainty from the perspective of each agent, but it does not change the true state of multi-agent systems. The state transition function is $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is a probability simplex denoting the set of all possible probability measures on $\mathcal{S}$. The state still transits from the true state to the next state. Each agent is associated with a policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ to choose an action $a^i \in \mathcal{A}^i$ given the perturbed state ρ^i. Note that the input of π^i is the perturbed state ρ^i. The perturbed state affects each agent's action. The set $\Delta(\mathcal{A}^i)$ includes all possible probability measures on $\mathcal{A}^i$. We use $\pi = (\pi^1, \pi^2, ..., \pi^n)$ to denote the joint agent policy.

The adversary policy, i.e. the state perturbation policy, associated with agent i is $\chi^i(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{P}_s^i)$, where the input of χ^i is the true state $s \in \mathcal{S}$. The power of the adversary is limited by the admissible perturbed state set $\mathcal{P}_s^i$. We denote the joint adversary policy as $\chi = (\chi^1, \chi^2, ..., \chi^n)$. The agents want to find a policy π to maximize their total expected return while adversaries want to find a policy χ to minimize the agents' total expected return.

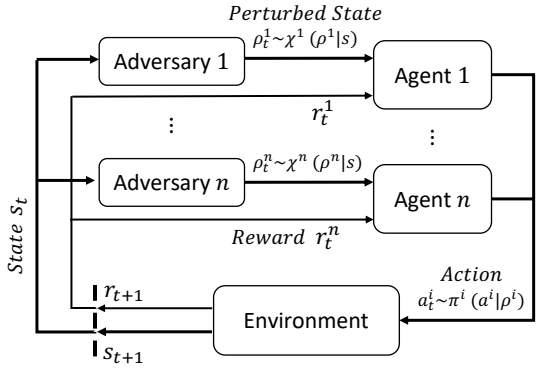

Figure 1: Multi-agent reinforcement learning under adversarial state perturbations. Each agent is associated with an adversary to perturb its knowledge or observation of the true state. Agents want to find a policy π to maximize their total expected return while adversaries want to find a policy χ to minimize agents' total expected return.

Our SAMG problem cannot be solved by the existing work for single-agent reinforcement learning with adversarial state perturbations (Mandlekar et al. 2017; Pattanaik et al. 2018; Zhang et al. 2020a, 2021). Each agent's action in SAMG is selected based on its own perturbed state observation and the state knowledge of each agent can be different after adversarial perturbations, so the SAMG problem cannot be solved by the above single-agent RL where the agent has only one state observation at each stage.

Our SAMG problem cannot be solved by the existing work in the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek, Amato et al. 2016; Lerer et al. 2020). In contrast, the policy in SAMG needs to be robust under a set of admissible perturbed states. The adversary aims to find the worst-case state perturbation policy χ to minimize the MARL agents' total expected return, but the Dec-POMDP cannot characterize the worst-case state perturbations. Moreover, all the agents cannot get the true state s in Dec-POMDP, while in an SAMG, the true state s is known by the adversaries. Adversaries can take the true state information and use it to select state perturbations for the MARL agents. More details about the connection between Dec-POMDP and SAMG are included in Appendix A.

Moreover, our SAMG problem cannot be solved by robust Markov games considering the uncertainties from reward (Zhang et al. 2020b), transition dynamics (Zhang et al. 2020b; Hu et al. 2020; Sinha, O'Kelly et al. 2020; Yu et al. 2021b), training partner's policies (Li et al. 2019; van der Heiden et al. 2020), because the MARL agents do not know the true state information after adversarial perturbations. Hence, we define the corresponding solution concepts of the policies for the adversaries and the agents, and analyze the properties of the policies for the SAMG in the next section.

4 Solution Concepts

In this section, we study the solution concepts of the SAMG. We formally define the optimal adversary policy, optimal agent policy, and robust Nash equilibrium. We then show that under the optimal adversary policy, the optimal agent policy or robust Nash equilibrium does not always exist. Instead, we propose a new objective, the worst-case expected state value, and prove that there exists a robust agent policy to maximize it.

We first introduce the widely used state value function concept for our proposed SAMG as follows:

$$V_{\pi,\chi}(s) = \mathbb{E}_{a_t \sim \pi(\cdot|\rho_t), \rho_t \sim \chi(\cdot|s_t)} \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}(s_t, a_t) | s_0 = s \right], \tag{1}$$

where γ is the discount factor.

4.1 Optimal Adversary Policy

For a fixed agent policy π, the optimal (worst-case) adversary $\chi^*(\pi)$ aims to minimize the agents' total expected return, that is to say,

$$V_{\pi,\chi^*(\pi)}(s) = \min_{\chi} V_{\pi,\chi}(s). \tag{2}$$

The following proposition shows the existence of the optimal adversary for an SAMG.

Proposition 1 (**Existence of Optimal Adversary Policy**). *Given an SAMG $G = (\mathcal{N}, \mathcal{S}, \mathcal{A}, r, \mathcal{P}_s, p, \gamma)$, for any fixed agent policy π, there exists an optimal adversary policy χ^* such that $V_{\pi,\chi^*(\pi)}(s) \leq V_{\pi,\chi}(s)$ for any $s \in \mathcal{S}$ and any χ.*

Proof. See proof in Appendix B. □

The key process of the proof is constructing an MDP for the adversary where the adversary gets the negative of the agent reward. Since for an MDP with finite state and finite action spaces, there always exists an optimal policy [Theorem 6.2.10 in (Puterman 2014)], the optimal adversary policy χ^* of the corresponding SAMG always exists as well.

4.2 Optimal Agent Policy

The optimal adversary policy is very powerful and it can easily corrupt the MARL agents' policies through state perturbations. We first define the optimal agent policy as follows:

Definition 2 (Optimal Agent Policy). *Under the optimal adversary policy χ^*, an agent policy π^* is an optimal policy if $V_{\pi^*,\chi^*(\pi^*)}(s) \geq V_{\pi,\chi^*(\pi)}(s)$ for any π and all $s \in \mathcal{S}$.*

In the following theorem, we show that the optimal agent policy π^* does not always exist for an SAMG under the optimal state perturbation adversary.

Theorem 1 (Non-existence of Optimal Agent Policy). *Under the optimal adversary policy χ^*, an optimal agent policy π^* does not always exist for an SAMG such that $V_{\pi^*,\chi^*(\pi^*)}(s) \geq V_{\pi,\chi^*(\pi)}(s)$ for any π and all $s \in \mathcal{S}$.*

Proof. See proof in Appendix B. □

The proof is done by constructing a counterexample where there is no optimal policy for the agents. An optimal agent policy π^* should maximize the state value for all states. However, under the adversarial state perturbations, sometimes agents have to make trade-offs between different state values and no agent policy can maximize all the state values.

4.3 Robust Nash Equilibrium

Then we look at the widely used Nash equilibrium concept in MARL for an SAMG. A Nash equilibrium is used to describe policies where no agent wants to deviate unilaterally. If an agent deviates from a Nash equilibrium, its total expected return won't increase. In a robust Nash equilibrium, we consider each agent is associated with an adversary that tries to minimize its total expected return.

Definition 3 (Robust Nash Equilibrium). *For an SAMG, the policy (π^*, χ^*) is a robust Nash equilibrium if for all $s \in \mathcal{S}$ and all $i \in \mathcal{N}$ and all π^i and χ^i, it holds that*

$$\begin{aligned} V_{\pi^i,\pi^{-i*},\chi^{i*},\chi^{-i*}}(s) &\leq V_{\pi^{i*},\pi^{-i*},\chi^{i*},\chi^{-i*}}(s) \\ &\leq V_{\pi^{i*},\pi^{-i*},\chi^{i},\chi^{-i*}}(s), \end{aligned} \tag{3}$$

where π^{-i} and χ^{-i} denotes the agent policies and adversary policies of all the other agents except agent i, respectively.

Definition 3 shows that π^* is in a robust Nash equilibrium if each agent's policy is a robust best response to the other agents' policies under adversarial state perturbations. When agent i is calculating its robust best response, it assumes a worst-case perspective of the state perturbations.

Theorem 2 (Non-existence of Robust Nash Equilibrium). *For an SAMG with finite state and finite action spaces, the robust Nash equilibrium defined in Definition 3 does not always exist.*

Proof. See proof in Appendix C. □

The proof is done by constructing a counterexample. For any state $s \in \mathcal{S}$, there exists a stage-wise equilibrium among the agents and adversaries (See details in Theorem 6 in Appendix C). However, there may not exist an equilibrium holding for all states, because the agents do not know the true state under adversarial state perturbations. The stage-wise equilibrium in one state may conflict with the stage-wise equilibrium in a different state. Sometimes agents have to make trade-offs between different states.

4.4 Robust Agent Policy

The optimal agent policy and robust Nash equilibrium concepts do not always exist in our SAMG problem according to the non-existence analysis. Instead, we propose a new objective, the worst-case expected state value, for the SAMG:

Definition 4 (Worst-case Expected State Value). *The worst-case expected state value under the optimal state perturbation adversary is:*

$$\mathbb{E}_{s_0 \sim \Pr(s_0)} \left[V_{\pi,\chi^*(\pi)}(s_0) \right] \tag{4}$$

where $\Pr(s_0)$ is the probability distribution of the initial state.

Even though we cannot maximize all the state values, we can use the probability of each state as an indicator of how important each state is. In an SAMG, we have to make trade-offs between different states. Therefore, we consider a weighted sum of all the state values according to the initial state distribution in the worst-case expected state value. The agent policy to maximize the worst-case expected state value is called a robust agent policy.

Definition 5 (Robust Agent Policy). *An agent policy π^* that maximizes the worst-case expected state value is called a robust agent policy:*

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{s_0 \sim \Pr(s_0)} \left[V_{\pi,\chi^*(\pi)}(s_0) \right]. \tag{5}$$

The following proposition shows finding a robust agent policy is equivalent to solving a maximin problem.

Proposition 2. *Finding an agent policy π to maximize the worst-case expected state value under the optimal adversary for π is equivalent to the maximin problem: $\max_\pi \min_\chi \sum_{s_0} \Pr(s_0) V_{\pi,\chi}(s_0)$.*

Proof. According to the Proposition 1, for any fixed agent policy π, there exists an optimal adversary policy χ^* such that $V_{\pi,\chi^*(\pi)}(s_0) = \min_\chi V_{\pi,\chi}(s_0)$ for any $s_0 \in \mathcal{S}$. Thus,

$$\begin{aligned} &\max_\pi \mathbb{E}_{s_0 \sim \Pr(s_0)} \left[V_{\pi,\chi^*(\pi)}(s_0) \right] \\ &= \max_\pi \mathbb{E}_{s_0 \sim \Pr(s_0)} \left[\min_\chi V_{\pi,\chi}(s_0) \right] \quad \text{(Eq. (2))} \\ &= \max_\pi \sum_{s_0} \Pr(s_0) \min_\chi V_{\pi,\chi}(s_0) \quad \text{(Definition of Expectation)} \\ &= \max_\pi \min_\chi \sum_{s_0} \Pr(s_0) V_{\pi,\chi}(s_0), \quad \text{(Proposition 1)} \end{aligned} \tag{6}$$

□

In the following theorem, we show the existence of robust agent policy for finite state and finite action spaces.

Theorem 3 (**Existence of Robust Agent Policy**). *For an SAMG with finite state and finite action spaces, there exists a robust agent policy π to maximize the worst-case expected state value defined in Definition 4.*

Proof. See proof in Appendix C. □

The proof is based on the Weierstrass M-test (Rudin et al. 1976), uniform limit theorem (Rudin et al. 1976), and the extreme value theorem. Different from the definitions of the optimal agent policy and robust Nash equilibrium, the worst-case expected state value objective does not require the optimality condition to hold for any state s. Agents won't get stuck in trade-offs between different states, therefore, we can find a robust agent policy to maximize the worst-case expected state value for the SAMG problem.

5 Robust MARL Algorithm

In general, it is challenging to develop an algorithm to compute an optimal or equilibrium policy for MARL with uncertainties (Zhang et al. 2020b, 2021). It is practical to use neural networks as function approximations in MARL in case the joint state and action spaces grow exponentially with the total number of agents. In order to deal with massive or even continuous state-action space, we design an actor-critic algorithm based on the recent MARL advances to approximate a robust agent policy under adversarial state perturbations. Our algorithm adopts centralized training and decentralized execution paradigm following the popular framework in (Lowe et al. 2017). During training, there is a centralized critic $Q(s, a)$ that records the total expected return given the global state s and global action a. The connection between $Q(s, a)$ and $V(s)$ is that for any $i \in \mathcal{N}, s \in \mathcal{S}, a \in \mathcal{A}$,

$$Q(s,a) = r(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s,a)V(s'). \tag{7}$$

Each agent's state input for the actor is perturbed by an adversary $\chi^i(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{P}_s^i)$. During execution, each agent i selects action a^i based on the perturbed state $\rho^i \in \mathcal{S}$ using a trained policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$. We want to find a policy π^i for each agent to maximize the worst-case expected state value in Definition 4 under adversarial state perturbations.

As shown in Alg. 1, our algorithm has a centralized critic network Q for training. Each agent has one actor network π^i and one adversary network χ^i. The critic Q takes in the true global state and global action during the training process. It returns a Q-value denoting the total expected return given s and a. The state transition experience is represented by (s, a, r, s') where s' is the next state. It is stored in a replay buffer $\mathcal{D}$ for the critic network's training. We apply "replay buffer" and "target network" techniques (Mnih, Kavukcuoglu et al. 2015). The critic network is trained with a mini-batch gradient descent optimizer in line 16. In line 18, we use Gradient Descent Ascent (GDA) optimizer (Lin, Jin, and Jordan 2020) to update parameters for each agent's actor network and adversary network for the maximin problem $\max_\pi \min_\chi \sum_{s_0} \Pr(s_0) V_{\pi,\chi}(s_0)$ in Proposition 2. A detailed introduction for the GDA optimizer is included in Appendix D.

Algorithm 1: Robust MARL Under Adversarial State Perturbations

1 Randomly initialize the critic network Q, the actor network π^i, and the adversary network χ^i for each agent;
2 Initialize target networks $Q', \pi^{i\prime}, \chi^{i\prime}$;
3 **for** *each episode* **do**
4 The initial state $s_0 \leftarrow$ sample from $\Pr(s_0)$;
5 Initialize a random process $\mathcal{X}$ for action exploration;
6 **for** *each time step* **do**
7 **for** *i=1 to n* **do**
8 $\rho^i \leftarrow$ sample from $\chi^i(\cdot|s)$;
9 $a^i \leftarrow$ sample from $\pi^i(\cdot|\rho^i) + \mathcal{X}$;
10 **end**
11 Execute actions $a = (a^1, ..., a^n)$;
12 Obtain the reward r and the next state s';
13 $\mathcal{D} \leftarrow \mathcal{D} \cup (s, a, r, s')$;
14 $s \leftarrow s'$;
15 $Q \leftarrow$ MGD_Optimizer$(Q, \mathcal{D}, Q', \pi', \chi')$;
16 /* Mini-batch gradient descent optimizer for critic. */
17 $\pi, \chi \leftarrow$ GDA_Optimizer(Q, π, χ);
18 /* Gradient descent ascent optimizer for policies. */
19 Update all target networks: $\theta^{i\prime} \leftarrow \tau\theta^i + (1-\tau)\theta^{i\prime}$.
20 **end**
21 **end**

6 Experiments

To show the effectiveness of our algorithm, we adopt the multi-agent particle environments developed in (Lowe et al. 2017) that have several agents and landmarks in a two-dimensional world. The host machine adopted in our experiments is a server configured with AMD Ryzen Threadripper 2990WX 32-core processors and four Quadro RTX 6000 GPUs. Our experiments are performed on Python 3.5.4, Gym 0.10.5, Numpy 1.14.5, Tensorflow 1.8.0, and CUDA 9.0. In our experiments, we consider the set of admissible perturbed state for agent i at state s as an ℓ_∞ norm ball around s: $\mathcal{P}_s^i := \{\rho^i \in \mathcal{S} : \|\rho^i - s\|_\infty \leq d\}$ where d is a radius denoting the perturbation budget. In implementation, the adversary network takes in the true state s and learns a state perturbation vector Δ^i and we project $s + \Delta^i$ to $\mathcal{P}_s^i$. Our code is available online on GitHub [1]. The environment used in experiments includes cooperative navigation (CN), exchange target (ET), keep-away (KA), physical deception (PD), and convert communication (CC). All hyperparameters used in experiments for RMARL and baselines are listed in Appendix D with more implementation details and experiment results.

[1] https://github.com/susanbao/RMARL_code

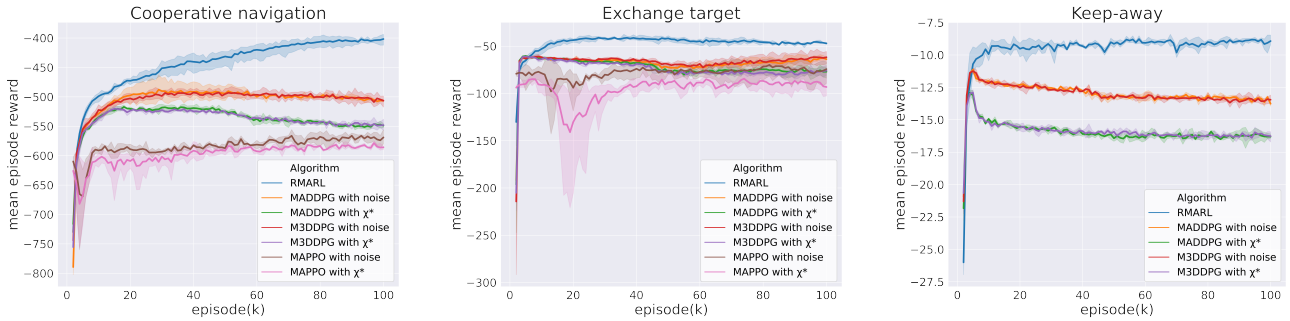

Figure 2: Our RMARL algorithm compared with baseline algorithms during the training process. Our RMARL algorithm gets higher mean episode rewards and is more robust to the state perturbations. All baselines are trained under random state perturbations or well-trained adversary policy χ^*. Because MAPPO only works in fully cooperative tasks, we only report its results in cooperative navigation and exchange target.

6.1 Baselines

We have in total 9 baselines in our experiment: MADDPG (MA) (Lowe et al. 2017), M3DDPG (M3) (Li et al. 2019), MAPPO (MP) (Yu et al. 2021a), MA/M3/MP with random state perturbations, MA/M3/MP with well-trained adversarial state perturbations. We introduce detail of these baselines in Appendix D. To test the robustness under the state uncertainty, we impose state noise to MADDPG, M3DDPG, and MAPPO produced by a truncated normal distribution $\mathcal{N}(0, \lambda, u, l)$ where λ is the uncertainty level, u and l are the upper and lower bounds to ensure noise compact. We use the truncated normal noise to simulate the adversaries selecting random state perturbations. While in our RMARL algorithm, agents are trained under adversaries that try to minimize the agents' total expected return. For each scenario, we save the well-trained adversaries χ^* in RMARL to represent the optimal state perturbation adversaries. We then use the well-trained adversaries to perturb the states for MADDPG, M3DDPG, and MAPPO to train and test their robustness under adversarial state perturbations. Because MAPPO only works in fully cooperative tasks, we only report its results in cooperative navigation and exchange target. For both training and testing, we report statistics that are averaged across 10 runs in each scenario and algorithm.

6.2 Comparison Results

Training Comparison Under different Perturbations We first compare our algorithm with baselines during the training process to show that our RMARL algorithm can outperform baselines to get higher mean episode rewards under different state perturbations. Note that our RMARL algorithm has a built-in adversary to perturb states, so we do not train it under random state perturbations. Comparing RMARL to other baselines with different state perturbations, the RMARL gets higher mean episode rewards. It shows our RMARL algorithm is more robust under different state perturbations. Comparing each baseline with random state perturbations to the same baseline with the well-trained adversary policy χ^*, we can see the adversary trained by the RMARL is more powerful than the random state perturbations. Because the adversary policy χ^* intentionally selects state perturbations to minimize agents' total expected return. The mean episode reward of the last 1000 episodes during training is shown in Table 1. Our RMARL algorithm achieves up to 58.46% higher mean episode rewards than the baselines under different state perturbations.

	CN	ET	KA
RM (ours)	**-401.7**	**-47.02**	**-8.93**
MA w/ $\mathcal{N}$	-506.48	-63.76	-13.76
M3 w/ $\mathcal{N}$	-506.54	-61.71	-13.45
MP w/ $\mathcal{N}$	-569.07	-94.28	-
MA w/ χ^*	-548.80	-77.01	-16.30
M3 w/ χ^*	-547.99	-75.87	-16.26
MP w/ χ^*	-585.83	-113.19	-

Table 1: Mean episode reward of the last 1000 episodes during the training. Our RMARL algorithm achieves up to 58.46% higher mean episode rewards than the baselines.

Training Comparison With More Agents We compare our RMARL algorithm with baselines in the cooperative navigation scenario with more agents added. The original cooperative navigation environment has 3 agents and the training results are shown in Fig. 2. We show the training results with 4 agents in Fig. 3. After increasing the total number of agents in the environment, our RMARL algorithm still gets higher mean episode rewards than baselines. We include the training results with more than 4 agents in Appendix D.

Training Comparison With Different Perturbation Budgets We compare our algorithm with baselines in the cooperative navigation scenario with different selections of d. We consider the set of admissible perturbed state for agent i at state s as an ℓ_∞ norm ball around s: $\mathcal{P}_s^i := \{\rho^i \in \mathcal{S} : \|\rho^i - s\|_\infty \leq d\}$ where d is a radius denoting the perturbation budget. As shown in Fig. 4, when d increases, adversaries get more freedom to perturb the state within a larger admissible perturbed state set. As d increases, adversaries get more powerful, and the agents' total expected return gets lower.

Testing Comparison in different Environments We then test the learned policies in different environments to show

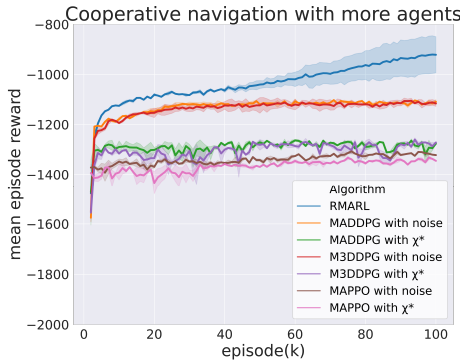

Figure 3: Our RMARL algorithm compared with baselines during the training process in the cooperative navigation scenario with more agents added. Our algorithm gets higher mean episode rewards in the environment with an increased agent number.

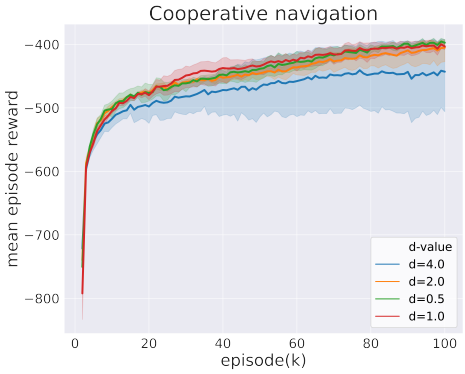

Figure 4: Our RMARL algorithm is trained in the cooperative navigation environment with different selections of d. When d increases, adversaries get more advantage/ freedom, and may further decrease agents' total expected return.

our RMARL policy is more robust under state perturbations. As shown in Table 2, the mean episode rewards are averaged across 2000 episodes and 10 test runs in each environment. We put all the well-trained agents using different algorithms into environments with injected random state perturbations. The original MADDPG, M3DDPG, and MAPPO's results are shown as a reference for no state perturbation scenario. The MADDPG, M3DDPG, and MAPPO get much lower mean episode rewards after injecting random state perturbations. It shows that state perturbations have a large impact on the MARL and an algorithm to handle state perturbations is in demand. As shown in Table 2, the RMARL policy achieves up to 59.68% higher mean episode rewards than baselines. It shows that the RMARL policy is more robust than baselines in environments with random state perturbations. Note that our RMARL's reward is slightly lower than MADDPG and M3DDPG because there is no uncertainty in the baselines' original settings. We also test the learned policy using different algorithms in environments with well-trained adversary policies χ^* to perturb states. The result is shown in Table 3. Our RMARL policy achieves up to 56.50% higher mean episode reward than the baselines with well-trained adversarial state perturbations. The result from the above two tests shows that our RMARL algorithm achieves higher robustness in different environments with state perturbations.

		CN	ET	KA	PD	CC
MA	μ	-388.59	-45.79	-8.8	3.03	3.53
	σ	60.72	23.5	5.07	0.67	14.72
M3	μ	-390.94	-39.55	-8.54	2.12	5.19
	σ	59.83	20.53	5.04	1.04	15.81
MP	μ	-381.70	-37.62	-	-	-
	σ	54.06	18.94	-	-	-
MA w/ $\mathcal{N}$	μ	-487.67	-55.79	-11.21	1.24	-16.88
	σ	72.28	26.78	6.82	**0.47**	0.46
M3 w/ $\mathcal{N}$	μ	-478.96	-54.40	-11.28	1.30	-16.75
	σ	70.27	26.64	6.71	0.58	**0.28**
MP w/ $\mathcal{N}$	μ	-523.83	-86.51	-	-	-
	σ	78.45	30.86	-	-	-
RM w/ $\mathcal{N}$	μ	**-437.42**	**-52.89**	**-9.89**	**1.98**	**-11.82**
	σ	**65.15**	**25.09**	**5.92**	0.93	0.79

Table 2: Mean episode reward of 2000 episodes during testing under random state perturbations. Our RMARL policy achieves up to 59.68% higher mean episode rewards than the baselines with random state perturbations.

		CN	ET	KA	PD	CC
MA w/χ^*	μ	-537.56	-71.65	-14.72	-0.95	-16.64
	σ	72.28	42.50	5.44	1.32	**0.20**
M3 w/χ^*	μ	-515.85	-70.68	-13.51	-0.70	-20.60
	σ	74.58	41.54	5.30	0.96	0.21
MP w/χ^*	μ	-572.39	-109.26	-	-	-
	σ	79.34	47.07	-	-	-
RM w/χ^*	μ	**-395.3**	**-47.53**	**-9.19**	**3.10**	**4.98**
	σ	**63.52**	**27.64**	**5.10**	**0.76**	15.54

Table 3: Mean episode rewards of 2000 episodes during testing under well-trained adversarial state perturbations. Our RMARL policy achieves up to 56.50% higher mean episode reward than the baselines with well-trained χ^*.

7 Conclusion

In this work, we propose a State-Adversarial Markov Game (SAMG) and study the fundamental properties of robust multi-agent reinforcement learning under adversarial state perturbations. We prove that the traditional solution concepts such as optimal agent policy and robust Nash equilibrium do not always exist for an SAMG. Instead, we propose a new solution concept, robust agent policy, to maximize the worst-case expected state value and prove the existence of a robust agent policy. This is the major theoretical contribution of our work. We also propose a Robust Multi-Agent Reinforcement Learning (RMARL) algorithm to find a robust policy for the MARL agents under adversarial state perturbations. We show in numerical experiments that the RMARL algorithm improves the robustness of the trained policies under random state perturbations or adversarial state perturbations. Some discussions and future work are listed in Appendix E.

References

Behzadan, V.; and Munir, A. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *MLDM*, 262–275. Springer.

Busoniu, L.; Babuska, R.; and De Schutter, B. 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst., Man, Cybern. Syst.*, 38(2): 156–172.

Debreu, G. 1952. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10): 886–893.

Everett, M.; Lütjens, B.; and How, J. P. 2021. Certifiable robustness to adversarial state uncertainty in deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.*

Fan, K. 1952. Fixed-point and minimax theorems in locally convex topological linear spaces. *Proceedings of the National Academy of Sciences of the United States of America*, 38(2): 121.

Fink, A. M. 1964. Equilibrium in a stochastic n-person game. *Journal of science of the hiroshima university, series ai (mathematics)*, 28(1): 89–93.

Foerster, J.; and Farquhar, G. 2018. Counterfactual multi-agent policy gradients. In *AAAI*.

Glicksberg, I. L. 1952. A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1): 170–174.

Guo, D.; Tang, L.; Zhang, X.; and Liang, Y.-C. 2020. Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning. *IEEE Trans. Veh. Technol.*, 69(11): 13124–13138.

Han, S.; Wang, H.; Su, S.; Shi, Y.; and Miao, F. 2022. Stable and Efficient Shapley Value-Based Reward Reallocation for Multi-Agent Reinforcement Learning of Autonomous Vehicles. *arXiv preprint arXiv:2203.06333*.

Hu, J.; Wellman, M. P.; et al. 1998. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, volume 98, 242–250. Citeseer.

Hu, Y.; Shao, K.; Li, D.; Jianye, H.; Liu, W.; Yang, Y.; Wang, J.; and Zhu, Z. 2020. Robust Multi-Agent Reinforcement Learning Driven by Correlated Equilibrium.

Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; and Abbeel, P. 2017. Adversarial attacks on neural network policies. *ICLR*.

Hüttenrauch, M.; and Šošić, A. 2017. Guided deep reinforcement learning for swarm systems. *arXiv preprint arXiv:1709.06011*.

Iqbal, S.; and Sha, F. 2019. Actor-attention-critic for multi-agent reinforcement learning. In *ICML*, 2961–2970. PMLR.

Iyengar, G. N. 2005. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280.

Jiang, Y.; Zhang, T.; Ho, D.; Bai, Y.; Liu, C. K.; Levine, S.; and Tan, J. 2021. Simgan: Hybrid simulator identification for domain adaptation via adversarial reinforcement learning. In *ICRA*, 2884–2890. IEEE.

Jin, C.; Liu, Q.; Wang, Y.; and Yu, T. 2021. V-Learning–A Simple, Efficient, Decentralized Algorithm for Multiagent RL. *arXiv preprint arXiv:2110.14555*.

Jin, C.; Netrapalli, P.; and Jordan, M. 2020. What is local optimality in nonconvex-nonconcave minimax optimization? In *ICML*, 4880–4889. PMLR.

Kardeş, E.; Ordóñez, F.; and Hall, R. W. 2011. Discounted robust stochastic games and an application to queueing control. *Operations research*, 59(2): 365–382.

Kos, J.; and Song, D. 2017. Delving into adversarial attacks on deep policies. *ICLR*.

Kothandaraman, D.; Chandra, R.; and Manocha, D. 2021. SS-SFDA: Self-supervised source-free domain adaptation for road segmentation in hazardous environments. In *ICCV*, 3049–3059.

Kreyszig, E. 1991. *Introductory functional analysis with applications*, volume 17. John Wiley & Sons.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Leibo, J. Z.; and Zambaldi, V. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *AAMAS*, 464–473.

Lerer, A.; Hu, H.; Foerster, J.; and Brown, N. 2020. Improving policies via search in cooperative partially observable games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7187–7194.

Li, S.; Wu, Y.; Cui, X.; Dong, H.; Fang, F.; and Russell, S. 2019. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *AAAI*, volume 33, 4213–4220.

Lin, J.; Dzeparoska, K.; Zhang, S. Q.; Leon-Garcia, A.; and Papernot, N. 2020. On the robustness of cooperative multi-agent reinforcement learning. In *2020 IEEE Security and Privacy Workshops (SPW)*, 62–68. IEEE.

Lin, T.; Jin, C.; and Jordan, M. 2020. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, 6083–6093. PMLR.

Lin, Y.-C.; Hong, Z.-W.; Liao, Y.-H.; Shih, M.-L.; Liu, M.-Y.; and Sun, M. 2017. Tactics of adversarial attack on deep reinforcement learning agents. *IJCAI*.

Liu, Z.; Cai, Y.; Wang, H.; Chen, L.; Gao, H.; Jia, Y.; and Li, Y. 2021. Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions. *IEEE Trans. Intell. Transp. Syst.*

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, 6379–6390.

Mandlekar, A.; Zhu, Y.; Garg, A.; Fei-Fei, L.; and Savarese, S. 2017. Adversarially robust policy learning: Active construction of physically-plausible perturbations. In *IROS*, 3932–3939. IEEE.

Mnih, V.; Kavukcuoglu, K.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Morimoto, J.; and Doya, K. 2005. Robust reinforcement learning. *Neural computation*, 17(2): 335–359.

Muniraj, D.; Vamvoudakis, K. G.; and Farhood, M. 2018. Enforcing signal temporal logic specifications in multi-agent adversarial environments: A deep q-learning approach. In *2018 IEEE Conference on Decision and Control (CDC)*, 4141–4146. IEEE.

Nash, J. 1951. Non-cooperative games. *Annals of mathematics*, 286–295.

Nilim, A.; and El Ghaoui, L. 2005. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5): 780–798.

Oliehoek, F. A.; Amato, C.; et al. 2016. *A concise introduction to decentralized POMDPs*, volume 1. Springer.

Pattanaik, A.; and Tang, Z. 2017. Robust deep reinforcement learning with adversarial attacks. *AAMAS*.

Pattanaik, A.; Tang, Z.; Liu, S.; Bommannan, G.; and Chowdhary, G. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. In *AAMAS*, 2040–2042.

Pinto, L.; Davidson, J.; and Sukthankar, R. 2017. Robust adversarial reinforcement learning. In *ICML*, 2817–2826. PMLR.

Pretorius, A.; Cameron, S.; et al. 2020. A game-theoretic analysis of networked system control for common-pool resource management using multi-agent reinforcement learning. In *NeurIPS*, volume 33, 9983–9994.

Puterman, M. L. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Qu, G.; Lin, Y.; Wierman, A.; and Li, N. 2020. Scalable Multi-Agent Reinforcement Learning for Networked Systems with Average Reward. In *NeurIPS*, volume 33, 2074–2086.

Rashid, T.; Farquhar, G.; Peng, B.; and Whiteson, S. 2020. Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *NeurIPS*.

Razaviyayn, M.; Huang, T.; Lu, S.; Nouiehed, M.; Sanjabi, M.; and Hong, M. 2020. Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances. *IEEE Signal Process. Mag.*, 37(5): 55–66.

Rudin, W.; et al. 1976. *Principles of mathematical analysis*, volume 3. McGraw-hill New York.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19): 70–76.

Shapley, L. S. 1953. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100.

Shen, M.; and How, J. P. 2021. Robust Opponent Modeling via Adversarial Ensemble Reinforcement Learning. In *ICAPS*, volume 31, 578–587.

Sinha, A.; O'Kelly, M.; et al. 2020. Formulazero: Distributionally robust online adaptation via offline population synthesis. In *ICML*, 8992–9004. PMLR.

Su, J.; Adams, S.; and Beling, P. 2021. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11352–11360.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning Multiagent Communication with Backpropagation. In *NeurIPS*.

Sun, C.; Kim, D.-K.; and How, J. P. 2021. ROMAX: Certifiably Robust Deep Multiagent Reinforcement Learning via Convex Relaxation. *arXiv preprint arXiv:2109.06795*.

Sunehag, P.; Lever, G.; et al. 2018. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *AAMAS*, 2085–2087.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

van der Heiden, T.; Salge, C.; Gavves, E.; and van Hoof, H. 2020. Robust Multi-Agent Reinforcement Learning with Social Empowerment for Coordination and Communication. *arXiv preprint arXiv:2012.08255*.

Xiao, C.; Pan, X.; et al. 2019. Characterizing attacks on deep reinforcement learning. *arXiv preprint arXiv:1907.09470*.

Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; and Wu, Y. 2021a. The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games. *arXiv preprint arXiv:2103.01955*.

Yu, J.; Gehring, C.; Schäfer, F.; and Anandkumar, A. 2021b. Robust Reinforcement Learning: A Constrained Game-theoretic Approach. In *Learning for Dynamics and Control*, 1242–1254. PMLR.

Zhang, H.; Chen, H.; Boning, D.; and Hsieh, C.-J. 2021. Robust reinforcement learning on state observations with learned optimal adversary. *arXiv preprint arXiv:2101.08452*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Liu, M.; Boning, D.; and Hsieh, C.-J. 2020a. Robust deep reinforcement learning against adversarial perturbations on state observations. *NeurIPS*, 33: 21024–21037.

Zhang, K.; Sun, T.; Tao, Y.; Genc, S.; Mallya, S.; and Basar, T. 2020b. Robust Multi-Agent Reinforcement Learning with Model Uncertainty. In *NeurIPS*.

Zhou, M.; Liu, Z.; Sui, P.; Li, Y.; and Chung, Y. Y. 2020. Learning Implicit Credit Assignment for Cooperative Multi-Agent Reinforcement Learning. In *NeurIPS*, volume 33, 11853–11864.

Appendix for "Robust Multi-Agent Reinforcement Learning Under Adversarial State Perturbations"

Numbering For the propositions and theorems that appeared in the main manuscript, we keep the same numbering in the Appendix.

A Comparison with Dec-POMDP and Markov Games

A.1 Comparison with Dec-POMDP

Our SAMG problem cannot be solved by the existing work in the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Lerer et al. 2020). In contrast, the policy in our problem needs to be robust under a set of admissible perturbed states. The adversary aims to find the worst-case state perturbation policy χ to minimize the MARL agents' total expected return. In the following proposition, we show that under certain additional conditions our proposed SAMG problem becomes a Dec-POMDP problem.

Proposition 3. *When the adversary policy χ is a fixed deterministic function, the SAMG problem becomes a Dec-POMDP (Lerer et al. 2020).*

Proof. When the adversary policy χ is a fixed deterministic function, an SAMG $(\mathcal{N}, \mathcal{S}, \mathcal{A}, r, \mathcal{P}_s, p, \gamma)$ becomes a Dec-POMDP $(\mathcal{N}, \mathcal{S}, \mathcal{A}, r, O, p, \gamma)$. The agent set $\mathcal{N} = \{1, ..., n\}$. The global joint state is $s \in \mathcal{S}$. Each agent i is associated with an action $a^i \in \mathcal{A}^i$. The global joint action is $a = (a^1, ..., a^n) \in \mathcal{A}$, $\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$. All agents share a stage-wise reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The state transition function is $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ is a probability simplex denoting the set of all possible probability measures on $\mathcal{S}$. The state transits from the true state to the next state. The discount factor is γ. When the adversary policy χ is a fixed deterministic function, the adversary policy follows $\chi^i(\rho^i|s) = 1$ selecting the perturbed state ρ^i for the true state s with probability 1. Let us use the notation $\chi^i(s) = \rho^i$ for this case. The observational function $o^i(s) = \chi^i(s) = \rho^i$, where o^i is the observation of agent i given the state s. □

Different from Dec-POMDP, the adversary policy χ is selected to minimize the agents' total expected return in our problem. Note that agents cannot get the true state s in Dec-POMDP, but in our problem, the true state s is known by the adversaries. Adversaries can take the true state information and use it to select the perturbed states.

A.2 Comparison with Markov Games

Under a special condition, when the adversary policy χ is a bijective mapping from $\mathcal{S}$ to $\mathcal{S}$, the SAMG problem becomes a Markov game as shown in the following proposition. This proposition shows the connection between a SAMG and a Markov game with a special type of state perturbation.

When χ is a bijective mapping from $\mathcal{S}$ to $\mathcal{S}$, the adversary policy follows $\chi(\rho|s) = 1$ selecting the perturbed state ρ for the true state s with probability 1. Let us use the notation $\chi(s) = \rho$ for this special case.

Proposition 4. *Specially, when the adversary policy χ is a bijective mapping from $\mathcal{S}$ to $\mathcal{S}$, the SAMG problem becomes a Markov game.*

Proof. When the adversary policy χ is a bijective mapping from $\mathcal{S}$ to $\mathcal{S}$, an SAMG problem $(\mathcal{N}, \mathcal{S}, \mathcal{A}, r, \mathcal{P}_s, p, \gamma)$ becomes a Markov game $(\mathcal{N}_{new}, \mathcal{S}_{new}, \mathcal{A}_{new}, r^i_{new}, p_{new}, \gamma)$ that is constructed as follows:

Taking $s_{new} = \rho = \chi(s)$ as the new state, the new global joint state set is $\mathcal{S}_{new} := \mathcal{S}$. The global joint action set $\mathcal{A}_{new} = \mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^n$ and the agent set $\mathcal{N}_{new} = \mathcal{N}$ stay the same.

We can construct a new reward function $r^i_{new} : \mathcal{S}_{new} \times \mathcal{A}_{new} \to \mathbb{R}$ for each agent i as

$$r^i_{new}(s_{new} = \chi(s), a_{new} = a) = r(s, a), \tag{8}$$

and a new state transition function $p_{new} : \mathcal{S}_{new} \times \mathcal{A}_{new} \to \Delta(\mathcal{S}_{new})$ defined as

$$p_{new}(\rho' = \chi(s')|\rho = \chi(s), a) = p(s'|s, a). \tag{9}$$

Each agent uses a policy $\pi^i_{new} : \mathcal{S}_{new} \to \Delta(\mathcal{A}^i)$ to choose an action based on the new state. Hence, the SAMG problem becomes a Markov game. □

When the adversary policy χ is a bijective mapping from $\mathcal{S}$ to $\mathcal{S}$, the new global joint state set $\mathcal{S}_{new}$ is a perturbation of $\mathcal{S}$ and each state gets a new "label" given by the adversary. In this case, the SAMG is reduced to a Markov game.

B Two-Agent Two-State Game

In this section, we analyze a small-scale game example with two agents $\mathcal{N} = \{1, 2\}$ and two states $\mathcal{S} = \{s_1, s_2\}$ as shown in Fig. 5. Each agent has two actions $\mathcal{A}^1 = \mathcal{A}^2 = \{a_1, a_2\}$. The transition probabilities are defined below.

$$\begin{aligned} &p(s' = s_1|s = s_1, a^1 \neq a^2) = 1, \\ &p(s' = s_2|s = s_1, a^1 = a^2) = 1, \\ &p(s' = s_2|s = s_2, a^1 \neq a^2) = 1, \\ &p(s' = s_1|s = s_2, a^1 = a^2) = 1. \end{aligned} \tag{10}$$

Specifically, $a^1 = a^2$ includes two cases: $a^1 = a^2 = a_1$ or $a^1 = a^2 = a_2$. Similarly, $a^1 \neq a^2$ includes two cases: $a^1 = a_1, a^2 = a_2$ or $a^1 = a_2, a^2 = a_1$.

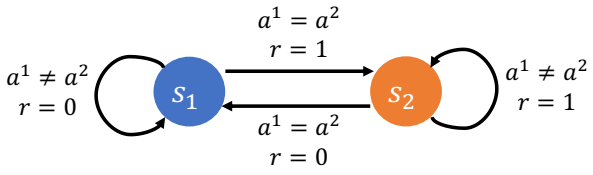

Figure 5: A two-agent two-state game example. Agents get reward 1 at state s_1 if they choose the same action. Agents get reward 1 at state s_2 if they choose different actions.

Two agents share the same reward function:

$$r(s, a^1, a^2) = \begin{cases} 1, & a^1 = a^2, \text{and } s = s_1, \\ 0, & a^1 \neq a^2, \text{and } s = s_1, \\ 0, & a^1 = a^2, \text{and } s = s_2, \\ 1, & a^1 \neq a^2, \text{and } s = s_2. \end{cases} \tag{11}$$

For the power of the adversary, we allow the adversary to perturb any state to the other state:

$$\mathcal{P}_s^1 = \mathcal{P}_s^2 = \{s_1, s_2\}. \tag{12}$$

We use $\gamma = 0.99$ as the discount factor.

B.1 Optimal Agent Policy Without Adversaries

When there is no adversary, the optimal policy for agents is to choose the same action in s_1 and choose different actions in s_2. One example is $\pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_1|s_1) = \pi^2(a_2|s_2) = 1$. The agents keep receiving rewards. The values for each state are $\tilde{V}(s_1) = \tilde{V}(s_2) = \frac{1}{1-\gamma} = 100$. Because agents share the same reward function, they also share the same values for each state. However, this policy receives $V(s_1) = V(s_2) = 0$ when agents are facing adversaries $\chi^i(s_1|s_2) = \chi^i(s_2|s_1) = 1$ for $i = 1, 2$ and always taking the wrong actions with 0 reward.

B.2 A Stochastic Policy With Adversaries

We consider a stochastic policy $\pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_1|s_1) = \pi^2(a_2|s_2) = 0.5$. Under this policy, the probabilities of taking the same or different actions are the same for each state $\Pr(a^1 = a^2 \mid s_1) = \Pr(a^1 \neq a^2 \mid s_1) = \Pr(a^1 = a^2 \mid s_2) = \Pr(a^1 \neq a^2 \mid s_2) = 0.5$. Agents randomly stay or transit in each state and receive a positive reward with a 50% probability. The adversary has no power under this policy because π is the same for both states. The values for each state are $V(s_1) = V(s_2) = \tilde{V}(s_1) = \tilde{V}(s_2) = \frac{0.5}{1-\gamma} = 50$.

B.3 Deterministic Policies With Adversaries

Since each agent has two actions for each state, there are in total $2^4 = 16$ possible deterministic policies for the two-agent two-state game example. All possible deterministic policies can be classified into three cases: (1) If agents select the same action in one state s_i and select different actions in the other state s_j, then we always have $V(s_1) = V(s_2) = 0$. This is because adversaries can always use $\chi^k(s_1|s_j) = \chi^k(s_2|s_i) = 1$ for $k = 1, 2$ such that agents always receive a 0 reward. (2) If agents always select different actions in both states, then $V(s_1) = 0, V(s_2) = 100$. This is because agents never transit to the other state and keep receiving the same reward. (3) If agents always select the same action in both states, then $V(s_1) = \frac{1}{1-\gamma^2} \approx 50.25, V(s_2) = \frac{\gamma}{1-\gamma^2} \approx 49.75$. This is because agents circulate through both states and adversaries have no power to change it.

B.4 Optimal Adversary Policy

In this section, we study the optimal adversary and the optimal agent policy in a State-Adversarial Markov Game (SAMG). We use the notation $\chi^*(\pi)$ to denote the optimal adversary policy for the agent policy π. In the following proposition, we show the existence of the optimal adversary in an SAMG.

Proposition 1 (Existence of Optimal Adversary Policy). *Given an SAMG $G = (\mathcal{N}, \mathcal{S}, \mathcal{A}, r, \mathcal{P}_s, p, \gamma)$, for any fixed agent policy π, there exists an optimal adversary policy χ^* such that $V_{\pi,\chi^*(\pi)}(s) \leq V_{\pi,\chi}(s)$ for any $s \in \mathcal{S}$ and any χ.*

Proof. We prove this by constructing an MDP $M = (\mathcal{S}, \hat{\mathcal{A}}, \hat{r}, \hat{p}, \gamma)$ such that the optimal policy of M is the optimal adversary policy χ^* for the SAMG given the fixed π. In the MDP M, we take all adversaries as a joint adversary agent. The joint adversary learns a policy χ to find a joint perturbed state given the current true state. The action space $\hat{\mathcal{A}} = \mathcal{S} \times \mathcal{S} \times \cdots \times \mathcal{S}$. Note that the joint admissible perturbed state set in Definition 1 $\mathcal{P}_s \subseteq \hat{\mathcal{A}}$.

The reward function $\hat{r}$ is defined as:

$$\hat{r}(s, \hat{a}) = -\sum_{a \in \mathcal{A}} \pi(a|\hat{a}) r(s, a) \text{ for } \hat{a} \in \mathcal{P}_s. \tag{13}$$

The transition probability $\hat{p}$ is defined as

$$\hat{p}(s'|s, \hat{a}) = \sum_{a \in \mathcal{A}} \pi(a|\hat{a}) p(s'|s, a) \text{ for } \hat{a} \in \mathcal{P}_s. \tag{14}$$

The reward function is defined based on the intuition that when the agent receives r given s, a, the reward of the adversary is the negative of the agent reward, that is to say, $\hat{r} = -r$. Considering that $r(s, a) = \mathbb{E}[R|s, a] = -\mathbb{E}[\hat{R}|s, a]$,

$$\begin{aligned} \hat{r}(s, \hat{a}) &= \mathbb{E}[\hat{R}|s, \hat{a}] \\ &= \sum_{\hat{R}} \hat{R} \sum_{a \in \mathcal{A}} \Pr[\hat{R}|s, a] \pi(a|\hat{a}) \\ &= \sum_{a \in \mathcal{A}} \left[\sum_{\hat{R}} \hat{R} \Pr[\hat{R}|s, a] \right] \pi(a|\hat{a}) \\ &= \sum_{a \in \mathcal{A}} \mathbb{E}[\hat{R}|s, a] \pi(a|\hat{a}) \\ &= -\sum_{a \in \mathcal{A}} \mathbb{E}[R|s, a] \pi(a|\hat{a}) \\ &= -\sum_{a \in \mathcal{A}} r(s, a) \pi(a|\hat{a}). \end{aligned} \tag{15}$$

Based on the properties of MDP (Sutton, Barto et al. 1998; Puterman 2014), we know that the MDP M has an optimal policy χ^* that satisfies $\hat{V}_{\pi,\chi^*}(s) \geq \hat{V}_{\pi,\chi}(s)$ for all s and all χ, where $\hat{V}_{\pi,\chi}$ is the state value function of the MDP M.

The Bellman equation for the MDP M is

$$\begin{aligned} \hat{V}_{\pi,\chi}(s) &= \sum_{\hat{a} \in \mathcal{P}_s} \chi(\hat{a}|s) \left(\hat{r} + \gamma \sum_{s' \in \mathcal{S}} \hat{p}(s'|s, \hat{a}) \hat{V}_{\pi,\chi}(s') \right) \\ &= \sum_{\hat{a} \in \mathcal{P}_s} \chi(\hat{a}|s) \sum_{a \in \mathcal{A}} \pi(a|\hat{a}) \left(-r + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a) \hat{V}_{\pi,\chi}(s') \right). \end{aligned} \tag{16}$$

By multiplying -1 on both sides, we have

$$(-\hat{V}_{\pi,\chi}(s)) = \sum_{\hat{a}\in\mathcal{P}_s} \chi(\hat{a}|s) \sum_{a\in\mathcal{A}} \pi(a|\hat{a}) \left[r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a)(-\hat{V}_{\pi,\chi}(s')) \right]. \quad (17)$$

On the other side, for the SAMG, we have the Bellman equation for any fixed policies π and χ as

$$V_{\pi,\chi}(s) = \sum_{\rho\in\mathcal{P}_s} \chi(\rho|s) \sum_{a\in\mathcal{A}} \pi(a|\rho) \left(r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) V_{\pi,\chi}(s') \right). \quad (18)$$

When π and χ are fixed, they can be taken together as a single policy, and the existing results from Dec-POMDP can be directly applied. Comparing Eq. (18) and (17), we know that $V_{\pi,\chi}(s) = (-\hat{V}_{\pi,\chi}(s))$.

The optimal adversary policy χ^* for the MDP M satisfies $\hat{V}_{\pi,\chi^*}(s)) \geq \hat{V}_{\pi,\chi}(s))$ for any s and any χ. Therefore, χ^* also satisfies $V_{\pi,\chi^*}(s) \leq V_{\pi,\chi}(s)$ for any s and any χ, and the optimal policy of the MDP M is the optimal adversary policy χ^* for the SAMG given the fixed π.

□

B.5 Optimal Agent Policy With Adversaries

Now we know the existence of the optimal adversary in an SAMG. Then we consider what is an optimal agent policy under the optimal adversary. We first show that the deterministic agent policy is not always better than a stochastic policy in an SAMG in the following proposition.

Proposition 5. *There exists an SAMG and some stochastic policy π such that we cannot find a better deterministic policy π' satisfying $V_{\pi',\chi^*(\pi')}(s) \geq V_{\pi,\chi^*(\pi)}(s)$ for all $s \in \mathcal{S}$.*

Proof. We prove this theorem by giving a counter-example where no deterministic policy is better than a stochastic policy. As shown in the two-agent two-state game example in Fig. 5, all 16 deterministic policies are no better than the stochastic policy $\pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_1|s_1) = \pi^2(a_2|s_2) = 0.5$. □

Finally, we show the optimal agent policy π^* does not always exist such that $V_{\pi^*,\chi^*(\pi^*)}(s) \geq V_{\pi,\chi^*(\pi)}(s)$ for any π and all $s \in \mathcal{S}$ under the optimal adversary policy χ^* in an SAMG in the following theorem.

Theorem 1 (Non-existence of Optimal Agent Policy). *Under the optimal adversary policy χ^*, an optimal agent policy π^* does not always exist for an SAMG such that $V_{\pi^*,\chi^*(\pi^*)}(s) \geq V_{\pi,\chi^*(\pi)}(s)$ for any π and any $s \in \mathcal{S}$.*

Proof. We prove this theorem by showing that the two-agent two-state game in Fig. 5 does not have an optimal policy. We first show that the policy $\pi_1 : \pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_2|s_1) = \pi^2(a_2|s_2) = 1$ is not an optimal policy. Because agents always select different actions in both states, agents always stay in the same state and adversaries have no power to change it. The values for each state are $V_{\pi_1,\chi^*(\pi_1)}(s_1) = 0, V_{\pi_1,\chi^*(\pi_1)}(s_2) = 100$. Now we consider the stochastic policy $\pi_2 : \pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_1|s_1) = \pi^2(a_2|s_2) = 0.5$. The values for each state are $V_{\pi_2,\chi^*(\pi_2)}(s_1) = V_{\pi_2,\chi^*(\pi_2)}(s_2) = 50$. Because $V_{\pi_2,\chi^*(\pi_2)}(s_1) > V_{\pi_1,\chi^*(\pi_1)}(s_1)$, the policy π_1 is not the optimal policy for agents.

If there exists an optimal policy π^*, then it must be better than π_1 and have $V_{\pi^*,\chi^*(\pi^*)}(s_1) > 0, V_{\pi^*,\chi^*(\pi^*)}(s_2) = 100$. In order to have $V_{\pi^*,\chi^*(\pi^*)}(s_2) = 100$, agents must select different actions in s_2 and keep receiving the positive rewards from each step. In order to have $V_{\pi^*,\chi^*(\pi^*)}(s_1) > 0$, agents must have a chance to select the same action in s_1, i.e., $\Pr(a^1 = a^2 \mid s_1) > 0$. However, if $\Pr(a^1 = a^2 \mid s_1) > 0$, then adversaries can have $\chi^i(s_1|s_2) > 0$ for $i = 1, 2$ to perturb the state s_2 to s_1 and reduce $V_{\pi^*,\chi^*(\pi^*)}(s_2)$. Therefore, no policy can do better than π_1 and since π_1 is not the optimal policy, there is no optimal policy for agents. □

In the comparison of π_1 and π_2 in the above proof, we see that it is not always possible to maximize the state value of all the states. Sometimes, we have to make trade-offs among different states. We cannot tell which policy is better between π_1 and π_2 according to the traditional definition of an optimal policy. However, if we use the worst-case expected state value concept in the Definition 4 and assume that the initial state is always s_2, then we can say π_1 is the optimal agent policy that gives the maximum worst-case expected state value 100 in this case.

C Stage-wise Equilibrium, Robust Nash Equilibrium, and Robust Agent Policy

As proved in Theorem 1, the optimal agent policy does not always exist for an SAMG. In this section, we discuss other solution concepts for the agent policy in an SAMG. We first show the existence of a unique robust state value function for each agent in C.1. Based on this property, we show the existence of a stage-wise equilibrium for each state in C.2. Though the stage-wise equilibrium exists for each state, the robust Nash equilibrium does not always exist as shown in C.3. Instead, we propose a new solution concept, the robust agent policy, and show its existence in C.4.

We first give a review of the Nash equilibrium used in the literature. Nash equilibrium is a solution concept proposed by Nash in (Nash 1951) for general-sum finite one-shot games. In Nash equilibrium, each player selects the best response strategy to the others' strategies. No player would like to deviate from the Nash equilibrium otherwise its utility may become worse. This concept is extended to infinite games by Debreu (Debreu 1952), Glicksberg (Glicksberg 1952), and Fan (Fan 1952). Markov game is first defined by Shapley in (Shapley 1953) in a two-player zero-sum setting with a sequential decision process. Fink extends the Nash equilibrium concept to Markov games in (Fink 1964) and proves that an equilibrium point exists in n-player general-

sum discounted Markov games. The uncertainty in transition dynamics of a Markov game is considered in (Nilim and El Ghaoui 2005; Iyengar 2005) using a robust optimization approach with independent proofs for the existence of the equilibrium point. Besides uncertainty in transition dynamics, uncertainty in utility (it is called "reward" in reinforcement learning) is also considered in (Kardeş, Ordóñez, and Hall 2011) for n-player finite state/action discounted Markov games with the proof for the existence of the equilibrium point.

However, the uncertainty in the state has not been studied yet for Markov games. To the best of our knowledge, we are the first to formulate the problem for n-player finite state/action discounted Markov games with state uncertainty and show the existence of the stage-wise equilibrium and the non-existence of the robust Nash equilibrium.

We use the following Assumption 1 throughout this section.

Assumption 1. *The global state set $\mathcal{S}$ and the global action set $\mathcal{A}$ are finite sets.*

C.1 Unique Robust State Value Function

Denote the agent policies and adversary policies of all other agents and adversaries except agent i and adversary i as π^{-i} and χ^{-i} respectively. We show that there exists a unique robust state value function $V^i_{*,\pi^{-i},*,\chi^{-i}}$ for agent i given any π^{-i} and χ^{-i}.

Definition 6 (Robust state value function). *A state value function $V^i_{*,\pi^{-i},*,\chi^{-i}} : \mathcal{S} \to \mathbb{R}$ for agent i given π^{-i} and χ^{-i} is called a robust state value function if for all $s \in \mathcal{S}$,*

$$V^i_{*,\pi^{-i},*,\chi^{-i}}(s) = \max_{\pi^i} \min_{\chi^i} \sum_{\rho\in\mathcal{P}_s} \chi(\rho|s) \sum_{a\in\mathcal{A}} \pi(a|\rho) \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) V^i_{*,\pi^{-i},*,\chi^{-i}}(s') \right). \quad (19)$$

Note that we use $\pi(a|\rho) = \Pi_{i=1}^n \pi^i(a^i|\rho^i)$ to denote the joint agent policy. We use $\chi(\rho|s) = \Pi_{i=1}^n \chi^i(\rho^i|s)$ to denote the joint adversary policy.

Before proving the existence of the unique robust state value function, we first introduce some notations for this proof. For a given state value function $V^i_{*,\pi^{-i},*,\chi^{-i}} : \mathcal{S} \to \mathbb{R}$ defined on a finite state set $\mathcal{S}$, we can construct a state value vector $v^i = \mathrm{vec}(V^i_{*,\pi^{-i},*,\chi^{-i}}) = [V^i_{*,\pi^{-i},*,\chi^{-i}}(s)]_{s\in\mathcal{S}} \in \mathcal{V} := \mathbb{R}^{|\mathcal{S}|}$ by traversing all states, where $\mathrm{vec}(\cdot)$ is a vectorization function. The infinity norm on $\mathcal{V}$ is $\|v^i\|_\infty = \max_{s\in\mathcal{S}} |V^i(s)|$. Define the total expected return in state s for π^i and χ^i as

$$f^i_s(v^i, \pi^i, \pi^{-i}, \chi^i, \chi^{-i}) = \sum_{\rho\in\mathcal{P}_s} \chi(\rho|s) \sum_{a\in\mathcal{A}} \pi(a|\rho) \left(r(s,a) + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) [\mathrm{vec}^{-1}(v^i)](s') \right), \quad (20)$$

where π^{-i} and χ^{-i} denotes the agent policies and the adversary policies of all other agents except agent i.

Define the robust state value in state s given π^{-i} and χ^{-i} as a function $\psi^i_s : \mathcal{V} \to \mathbb{R}$,

$$\psi^i_s(v^i, \pi^{-i}, \chi^{-i}) = \max_{\pi^i} \min_{\chi^i} f^i_s(v^i, \pi^i, \pi^{-i}, \chi^i, \chi^{-i}). \quad (21)$$

Note that ψ^i_s gives a real number that denotes the total expected return in state s given π^{-i} and χ^{-i}. We can construct a mapping $\Psi^i_{\pi,\chi} : \mathcal{V} \to \mathcal{V}$ from any state value vector v^i to a robust state value vector $[\Psi^i_{\pi,\chi}(v^i)]_{s\in\mathcal{S}}$ by traversing all s, that is to say, $[\Psi^i_{\pi,\chi}(v^i)]_{s\in\mathcal{S}} = \psi^i_s(v^i, \pi^{-i}, \chi^{-i})$.

Lemma 4. *For any $i \in \mathcal{N}$, the function $\Psi^i_{\pi,\chi} : \mathcal{V} \to \mathcal{V}$ is a contraction mapping given any π^{-i} and χ^{-i} of other agents and adversaries except agent i and adversary i.*

Proof. Let us consider two vectors $v^i, z^i \in \mathcal{V}$. For any $i \in \mathcal{N}$, given any π^{-i} and χ^{-i}, for all $s \in \mathcal{S}$, we have

$$\begin{aligned} \psi^i_s(v^i, \pi^{-i}, \chi^{-i}) &= \max_{\pi^i} \min_{\chi^i} f^i_s(v^i, \pi^i, \pi^{-i}, \chi^i, \chi^{-i}) \\ &= f^i_s(v^i, \pi^{i*}, \pi^{-i}, \chi^{i*}, \chi^{-i}), \end{aligned} \quad (22)$$

where π^{i*} is the corresponding maximizer, and χ^{i*} is the corresponding optimizer for π^{i*}. Similarly, with the optimizers ω^{i*} and φ^{i*}_1 for the following maximin optimization problem, we have

$$\begin{aligned} \psi^i_s(z^i, \pi^{-i}, \chi^{-i}) &= \max_{\omega^i} \min_{\varphi^i} f^i_s(z^i, \omega^i, \pi^{-i}, \varphi^i, \chi^{-i}) \\ &= f^i_s(z^i, \omega^{i*}, \pi^{-i}, \varphi^{i*}_1, \chi^{-i}) \\ &\geq f^i_s(z^i, \pi^{i*}, \pi^{-i}, \varphi^{i*}_2, \chi^{-i}), \end{aligned} \quad (23)$$

where

$$\varphi^{i*}_2 = \arg\min_{\varphi^i} f^i_s(z^i, \pi^{i*}, \pi^{-i}, \varphi^i, \chi^{-i}). \quad (24)$$

Then, for any $i \in \mathcal{N}$, given any π^{-i} and χ^{-i}, for all $s \in \mathcal{S}$, it holds that

$$\begin{aligned} &\psi^i_s(v^i, \pi^{-i}, \chi^{-i}) - \psi^i_s(z^i, \pi^{-i}, \chi^{-i}) \\ &= f^i_s(v^i, \pi^{i*}, \pi^{-i}, \chi^{i*}, \chi^{-i}) - f^i_s(z^i, \omega^{i*}, \pi^{-i}, \varphi^{i*}_1, \chi^{-i}) \\ &\leq f^i_s(v^i, \pi^{i*}, \pi^{-i}, \chi^{i*}, \chi^{-i}) - f^i_s(z^i, \pi^{i*}, \pi^{-i}, \varphi^{i*}_2, \chi^{-i}) \\ &\leq f^i_s(v^i, \pi^{i*}, \pi^{-i}, \varphi^{i*}_2, \chi^{-i}) - f^i_s(z^i, \pi^{i*}, \pi^{-i}, \varphi^{i*}_2, \chi^{-i}) \\ &= \sum_{\rho\in\mathcal{P}_s} \varphi^{i*}_2(\rho^i|s) \prod_{j\neq i} \chi^j(\rho^j|s) \sum_{a\in\mathcal{A}} \pi^{i*}(a^i|\rho^i) \times \\ &\quad \prod_{k\neq i} \pi^k(a^k|\rho^k) \left(r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) [\mathrm{vec}^{-1}(v^i)](s') \right) \\ &\quad - \sum_{\rho\in\mathcal{P}_s} \varphi^{i*}_2(\rho^i|s) \prod_{j\neq i} \chi^j(\rho^j|s) \sum_{a\in\mathcal{A}} \pi^{i*}(a^i|\rho^i) \times \\ &\quad \prod_{k\neq i} \pi^k(a^k|\rho^k) \left(r + \gamma \sum_{s'\in\mathcal{S}} p(s'|s,a) [\mathrm{vec}^{-1}(z^i)](s') \right) \\ &= \sum_{\rho\in\mathcal{P}_s} \varphi^{i*}_2(\rho^i|s) \prod_{j\neq i} \chi^j(\rho^j|s) \sum_{a\in\mathcal{A}} \pi^{i*}(a^i|\rho^i) \times \end{aligned}$$

$$\prod_{k\neq i}\pi^k(a^k|\rho^k)\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)\times$$
$$\left\{[\mathrm{vec}^{-1}(v^i)](s')-[\mathrm{vec}^{-1}(z^i)](s')\right\}$$
$$\leq\sum_{\rho\in\mathcal{P}_s}\varphi_2^{i*}(\rho^i|s)\prod_{j\neq i}\chi^j(\rho^j|s)\sum_{a\in\mathcal{A}}\pi^{i*}(a^i|\rho^i)\times$$
$$\prod_{k\neq i}\pi^k(a^k|\rho^k)\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)\|v^i-z^i\|_\infty$$
$$=\gamma\|v^i-z^i\|_\infty. \tag{25}$$

The second inequality in Eq. (25) follows

$$\chi^{i*}=\arg\min_{\chi^i}f_s^i(v^i,\pi^{i*},\pi^{-i},\chi^i,\chi^{-i}). \tag{26}$$

Because for any $i\in\mathcal{N}$, given any π^{-i} and χ^{-i}, for all $s\in\mathcal{S}$

$$\psi_s^i(v^i,\pi^{-i},\chi^{-i})-\psi_s^i(z^i,\pi^{-i},\chi^{-i})\leq\gamma\|v^i-z^i\|_\infty, \tag{27}$$

Based on symmetry, we have

$$\psi_s^i(z^i,\pi^{-i},\chi^{-i})-\psi_s^i(v^i,\pi^{-i},\chi^{-i})\leq\gamma\|z^i-v^i\|_\infty$$
$$=\gamma\|v^i-z^i\|_\infty. \tag{28}$$

Thus, it holds that for any $i\in\mathcal{N}$, given any π^{-i} and χ^{-i}

$$\|\Psi_{\pi,\chi}^i(v^i)-\Psi_{\pi,\chi}^i(z^i)\|_\infty\leq\gamma\|v^i-z^i\|_\infty, \tag{29}$$

that is to say, the function $\Psi_{\pi,\chi}^i$ is a contraction mapping. □

Theorem 5 (**Existence of Unique State Value Function**). *For an SAMG with finite state and finite action spaces, for any $i\in\mathcal{N}$, given any π^{-i} and χ^{-i} of other agents and adversaries except agent i and adversary i, there exists a unique robust state value function $V_{*,\pi^{-i},*,\chi^{-i}}^i:\mathcal{S}\to\mathbb{R}$ for agent i such that for all $s\in\mathcal{S}$,*

$$V_{*,\pi^{-i},*,\chi^{-i}}^i(s)=\max_{\pi^i}\min_{\chi^i}\sum_{\rho\in\mathcal{P}_s}\chi(\rho|s)\sum_{a\in\mathcal{A}}\pi(a|\rho)$$
$$\left(r(s,a)+\gamma\sum_{s'\in\mathcal{S}}p(s'|s,a)V_{*,\pi^{-i},*,\chi^{-i}}^i(s')\right). \tag{30}$$

Proof. For any $i\in\mathcal{N}$, there exists a state value function $V_{*,\pi^{-i},*,\chi^{-i}}^i$ satisfying (30) if and only if $v^i=\mathrm{vec}(V_{*,\pi^{-i},*,\chi^{-i}}^i)$ is a fixed point of $\Psi_{\pi,\chi}^i:\mathcal{V}\to\mathcal{V}$, where $[\Psi_{\pi,\chi}^i(v^i)]_{s\in\mathcal{S}}=\psi_s^i(v^i,\pi^{-i},\chi^{-i})$ and $\psi_s^i(v^i,\pi^{-i},\chi^{-i})$ is defined in (21). We use Banach's fixed point theorem to prove this as follows.

Because any finite-dimensional normed vector space is complete (Kreyszig 1991), the $(\mathcal{V},\|\cdot\|_\infty)$ is a complete Banach space. Also, for any $i\in\mathcal{N}$, given any π^{-i} and χ^{-i}, the function $\Psi_{\pi,\chi}^i$ is a contraction mapping according to Lemma 4. Therefore, by Banach's fixed point theorem, there is a unique fixed point v^i such that $\Psi_{\pi,\chi}^i(v^i)=v^i$. In other words, for any $i\in\mathcal{N}$, given any π^{-i} and χ^{-i}, there exists a unique $V_{*,\pi^{-i},*,\chi^{-i}}^i$ such that

$$V_{*,\pi^{-i},*,\chi^{-i}}^i(s)=\max_{\pi^i}\min_{\chi^i}f_s^i(v^i,\pi^i,\pi^{-i},\chi^i,\chi^{-i}). \tag{31}$$

□

Denote the state value function for agent i given any π^{-i} and χ^{-i} of other agents and adversaries except agent i and adversary i as

$$V_{\pi^i,\pi^{-i},\chi^i,\chi^{-i}}^i(s)=f_s^i(v^i,\pi^i,\pi^{-i},\chi^i,\chi^{-i}), \tag{32}$$

where $v^i=\mathrm{vec}(V_{*,\pi^{-i},*,\chi^{-i}}^i)$. Then we have the following corollary for Theorem 5.

Corollary 5.1. *For an SAMG with finite state and finite action spaces, let $V_{*,\pi^{-i},*,\chi^{-i}}^i$ be the unique robust state value function for agent i given any π^{-i} and χ^{-i} such that for all $s\in\mathcal{S}$,*

$$V_{*,\pi^{-i},*,\chi^{-i}}^i(s)=\max_{\pi^i}\min_{\chi^i}f_s^i(v^i,\pi^i,\pi^{-i},\chi^i,\chi^{-i})$$
$$=f_s^i(v^i,\pi^{i*},\pi^{-i},\chi^{i*},\chi^{-i}), \tag{33}$$

where $v^i=\mathrm{vec}(V_{*,\pi^{-i},*,\chi^{-i}}^i)$, π^{i*} is the corresponding maximizer at state s, and χ^{i*} is the corresponding optimizer for π^{i*} at state s, then for state s it holds that $V_{\pi^{i*},\pi^{-i},\chi^{i*},\chi^{-i}}^i(s)\geq V_{\pi^i,\pi^{-i},\chi^{i*},\chi^{-i}}^i(s)$ for any π^i, and $V_{\pi^{i*},\pi^{-i},\chi^{i*},\chi^{-i}}^i(s)\leq V_{\pi^{i*},\pi^{-i},\chi^i,\chi^{-i}}^i(s)$ for any χ^i.

C.2 Existence of the Stage-wise Equilibrium

Before we show the existence of the robust Nash equilibrium, we first show a concept of the stage-wise equilibrium.

Definition 7 (**Stage-wise Equilibrium**). *For an SAMG, the policy (π^*,χ^*) is a stage-wise equilibrium for state s if for all $i\in\mathcal{N}$ and all π^i and χ^i, it holds that*

$$V_{\pi^i,\pi^{-i*},\chi^{i*},\chi^{-i*}}^i(s)\leq V_{\pi^{i*},\pi^{-i*},\chi^{i*},\chi^{-i*}}^i(s)$$
$$\leq V_{\pi^{i*},\pi^{-i*},\chi^i,\chi^{-i*}}^i(s), \tag{34}$$

where π^{-i} and χ^{-i} denotes the agent policies and adversary policies of all the other agents except agent i, respectively.

Originally, the Nash equilibrium concept is proposed by Nash for the finite one-shot game, where the state transition of the environment is not considered. When the Nash equilibrium concept is extended for Markov games, the existence of the Nash equilibrium is shown by the existence of the state-wise equilibrium for each state. If a policy is a stage-wise equilibrium for all states, then this policy is a Nash equilibrium for the Markov game.

We show the existence of the stage-wise equilibrium defined in Definition 7 in the following theorem.

Theorem 6 (**Existence of Stage-wise equilibrium**). *For an SAMG with finite state and finite action spaces, the stage-wise equilibrium defined in Definition 7 exists for any $s\in\mathcal{S}$.*

Proof. Let us construct a $2n$ player game for any $s\in\mathcal{S}$. We have n agents and n adversaries in the player set. We introduce uniform notations for the agents and adversaries to describe a $2n$ player game at state s. The player set $\mathcal{I}=\{1,...,n,n+1,...,2n\}$. The first half of the player set $\{1,...,n\}$ represents agents, while the second half $\{n+1,...,2n\}$ represents adversaries. The set of available actions for player i is

$$A_s^i=\begin{cases}\underbrace{\mathcal{A}^i\times\mathcal{A}^i\cdots\times\mathcal{A}^i}_{\text{total number: }|\mathcal{P}_s^i|}, & i=1,...,n;\\ \mathcal{P}_s^{i-n}, & i=n+1,...,2n.\end{cases} \tag{35}$$

Each adversary's action set includes all possible perturbed states in the admissible perturbed state set at state s. Each agent's action set includes all possible joint actions given every possible perturbed state. Take the two-agent two-state game in Fig. 5 as an example, the player set $\mathcal{I} = \{1, 2, 3, 4\}$. Player 3 is the adversary for agent player 1. Player 4 is the adversary for agent player 2. If the current true state is s_1, then $A^1_{s_1} = A^2_{s_1} = \{(a_1, a_1), (a_1, a_2), (a_2, a_2), (a_2, a_1)\}$ are the action sets for two agent players. In $A^1_{s_1}$ for agent 1, the joint action (a_1, a_2) means selecting a_1 if the perturbed state for agent 1 is s_1 and selecting a_2 if the perturbed state for agent 1 is s_2. For two adversary players, $A^3_{s_1} = A^4_{s_1} = \{s_1, s_2\}$, as adversaries can perturb the true state s_1 to s_2.

We consider the mixed strategy $\sigma^i_s \in \Delta(A^i_s)$ for player i. Note that the mixed strategy for each adversary gives us the probability distribution of all possible perturbed states for state s, i.e. $\chi^{i-n}(\rho^{i-n}|s) = \sigma^i_s(\rho^{i-n})$ for $i = n + 1, ..., 2n$. Then we show how we can get each agent's policy $\pi^i(a^i|\rho^i)$ based on its mixed strategy σ^i_s by calculating the marginal probabilities. Denote the total number of possible perturbed state for agent i at state s as P such that $P = |\mathcal{P}^i_s|$. Here we drop the subscript s in P_s for a concise representation. The perturbed state set for agent i is represented as $\{\rho^i_1, \rho^i_2, \ldots, \rho^i_P\}$. Denote the joint action of agent i as $b^i = (b^i_1, b^i_2, \ldots, b^i_P)$ where b^i_k is the action selected for the perturbed state $\rho^i_k \in \mathcal{P}^i_s$. Then the mixed strategy $\sigma^i_s(b^i_1, b^i_2, ..., b^i_P)$ gives us the joint probability of selecting b^i_k for ρ^i_k for all $k = 1, 2, ..., P$. We can get the marginal probability of selecting action a^i given the perturbed state $\rho^i_k \in \mathcal{P}^i_s$ as

$$\pi^i(a^i|\rho^i_k) = \sum_{\{b^i \in A^i_s | b^i_k = a^i\}} \sigma^i_s(b^i_1, b^i_2, ..., b^i_P). \tag{36}$$

The marginal probability of selecting action a^i given the perturbed state ρ^i_k is calculated by summing up the joint probability over all joint actions in which agent i selects a^i given the perturbed state ρ^i_k. Take the two-agent two-state game in Fig. 5 as an example, if the current perturbed state for agent 1 is $\rho^1 = s_1$, then agent 1's policy is

$$\begin{aligned}\pi^1(a_1|\rho^1 = s_1) &= \sigma^1(a_1, a_1) + \sigma^1(a_1, a_2)\\ \pi^1(a_2|\rho^1 = s_1) &= \sigma^1(a_2, a_1) + \sigma^1(a_2, a_2).\end{aligned} \tag{37}$$

Note that the mixed strategy $\sigma^i_s \in \Delta(A^i_s)$ only gives part of the agent and adversary policies. For example, the mixed strategy for the adversaries only gives a distribution of the perturbed states for $s_t = s$. We construct the complete agent and adversary policies as follows: For $i = 1, ..., n$, the agent i's policy is

$$\pi^i(a^i|\rho^i) = \begin{cases} \sum_{\{b^i \in A^i_s | b^i_k = a^i\}} \sigma^i_s(b^i_1, b^i_2, ..., b^i_P), \\ \text{for } \rho^i = \rho^i_k \in \mathcal{P}^i_s; \\ \mathcal{U}(\mathcal{A}^i), \\ \text{for } \rho^i \notin \mathcal{P}^i_s, \end{cases} \tag{38}$$

where $\mathcal{U}(\mathcal{A}^i)$ represents a uniform distribution on $\mathcal{A}^i$. For $i = 1, ..., n$, the adversary i's policy is

$$\chi^i(\rho^i|s_t) = \begin{cases} \sigma^{i+n}_s(\rho^i), & \text{for } s_t = s; \\ \mathcal{U}(\mathcal{P}^i_s), & \text{for } s_t \neq s, \end{cases} \tag{39}$$

where $\mathcal{U}(\mathcal{P}^i_s)$ represents a uniform distribution on $\mathcal{P}^i_s$.

The utility function for player i is

$$u^i_s(\sigma^i_s, \sigma^{-i}_s) = \begin{cases} f^i_s(v^{i*}, \pi^i, \pi^{-i}, \chi^i, \chi^{-i}), \\ \text{for } i = 1, ..., n; \\ -f^{i-n}_s(v^{(i-n)*}, \pi^{i-n}, \pi^{-(i-n)}, \\ \chi^{i-n}, \chi^{-(i-n)}), \\ \text{for } i = n+1, ..., 2n. \end{cases} \tag{40}$$

where σ^{-i}_s denotes the strategies of all other players except player i, $v^{i*} = \text{vec}(V^i_{*,\pi^{-i},*,\chi^{-i}})$, and $V^i_{*,\pi^{-i},*,\chi^{-i}}$ is the unique robust state value function of agent i when the policies of other agents and adversaries are given by π^{-i} and χ^{-i}. The v^{i*} satisfies

$$[\text{vec}^{-1}(v^{i*})](s) = \max_{\pi^i} \min_{\chi^i} f^i_s(v^{i*}, \pi^i, \pi^{-i}, \chi^i, \chi^{-i}), \tag{41}$$

where f^i_s is defined for player i in (20) as

$$\begin{aligned} f^i_s(v^i, \pi^i, \pi^{-i}, \chi^i, \chi^{-i}) &= \sum_{\rho \in \mathcal{P}_s} \chi(\rho|s) \sum_{a \in \mathcal{A}} \pi(a|\rho) \\ &\left(r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a)[\text{vec}^{-1}(v^i)](s') \right). \end{aligned}$$

Note that σ^{-i}_s includes both π^{-i} and χ^{-i} for any $i \in \mathcal{I}$, and the existence of $V^i_{*,\pi^{-i},*,\chi^{-i}}$ is guaranteed by Theorem 5. Thus, the utility function is well-defined.

Since the state set $\mathcal{S}$ is finite, $\mathcal{P}^i_s \subseteq \mathcal{S}$ is a finite set for all $i \in \mathcal{N}$. Also, $\mathcal{A}^i$ is a finite set for all $i \in \mathcal{N}$. Therefore, $\Delta(A^i_s)$ is compact and convex for all $i \in \mathcal{I}$. Moreover, for all $i \in \mathcal{I}$, $u^i_s(\sigma^i_s, \cdot)$ is linear in σ^i_s and therefore continuous and concave in σ^i_s. According to the theorem (Debreu (Debreu 1952), Glicksberg (Glicksberg 1952), Fan (Fan 1952)), the conditions for the existence of a Nash Equilibrium are satisfied, hence, there exists a Nash equilibrium σ^*_s for this $2n$ player game for any $s \in \mathcal{S}$ such that for any $i \in \mathcal{I}$, $u^i_s(\sigma^{i*}_s, \sigma^{-i*}_s) \geq u^i_s(\sigma^i_s, \sigma^{-i*}_s)$ for any σ^i_s.

Denote the agent and adversary policies as (π^*, χ^*) that are constructed following Eq. (38) and Eq. (39) by plugging in the Nash equilibrium $(\sigma^{i*}_s, \sigma^{-i*}_s)$. Substituting the (π^*, χ^*) into $u^i_s(\sigma^{i*}_s, \sigma^{-i*}_s) \geq u^i_s(\sigma^i_s, \sigma^{-i*}_s)$ and plugging in the definition of the utility functions, for any $i = 1, 2, ..., n$, it holds that

$$f^i_s(v^{i*}, \pi^{i*}, \pi^{-i*}, \chi^{i*}, \chi^{-i*}) \geq f^i_s(v^{i*}, \pi^i, \pi^{-i*}, \chi^{i*}, \chi^{-i*}), \tag{42}$$

for any π^i. Also, for any $i = 1, 2, ..., n$, it holds that

$$f^i_s(v^{i*}, \pi^{i*}, \pi^{-i*}, \chi^{i*}, \chi^{-i*}) \leq f^i_s(v^{i*}, \pi^{i*}, \pi^{-i*}, \chi^i, \chi^{-i*}), \tag{43}$$

for any χ^i. Therefore,

$$\begin{aligned} &\max_{\pi^i} \min_{\chi^i} f^i_s(v^{i*}, \pi^i, \pi^{-i*}, \chi^i, \chi^{-i*}) \\ =&f^i_s(v^{i*}, \pi^{i*}, \pi^{-i*}, \chi^{i*}, \chi^{-i*}). \end{aligned} \tag{44}$$

According to Corollary 5.1, for any π^i, it holds that

$$V^i_{\pi^{i*},\pi^{-i*},\chi^{i*},\chi^{-i*}}(s) \geq V^i_{\pi^i,\pi^{-i*},\chi^{i*},\chi^{-i*}}(s), \tag{45}$$

Also, for any χ^i, it holds that

$$V^i_{\pi^{i*},\pi^{-i*},\chi^{i*},\chi^{-i*}}(s) \leq V^i_{\pi^{i*},\pi^{-i*},\chi^{i},\chi^{-i*}}(s). \quad (46)$$

Thus, the stage-wise equilibrium defined in Definition 7 exists for any $s \in \mathcal{S}$. □

C.3 Non-existence of Robust Nash Equilibrium

Theorem 6 shows the existence of the stage-wise equilibrium for any $s \in \mathcal{S}$. For the classic Markov game (Fink 1964) and Markov game with reward/ transition uncertainties (Kardeş, Ordóñez, and Hall 2011; Nilim and El Ghaoui 2005; Iyengar 2005), this result is naturally extended to be the existence of the Nash equilibrium policy because all agents' and adversaries' policies are based on the current true state. If the stage-wise equilibrium exists for any state $s \in \mathcal{S}$, then a Nash equilibrium can be constructed by taking the policies for each state s from their corresponding stage-wise equilibriums for state s (Fink 1964; Kardeş, Ordóñez, and Hall 2011; Nilim and El Ghaoui 2005; Iyengar 2005). However, this natural extension cannot be used for our SAMG problem because the agent policy is based on the perturbed state instead of the true state. The problem is the agent's stage-wise equilibrium in one state may not be consistent with its stage-wise equilibrium in a different state. We illustrate this idea in the following theorem to show that the robust Nash equilibrium does not always exist for an SAMG.

Theorem 2 (**Non-existence of Robust Nash Equilibrium**). *For an SAMG with finite state and finite action spaces, the robust Nash equilibrium defined in Definition 3 does not always exist.*

Proof. We prove this theorem by showing that the following two-agent two-state game in Fig. 6 does not have a robust Nash equilibrium. The two-agent two-state game in Fig. 6 is

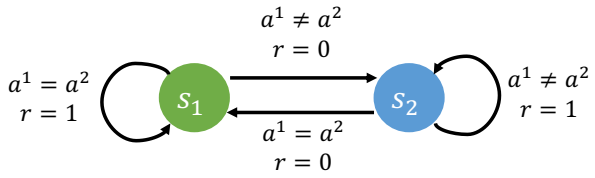

Figure 6: A new two-agent two-state game example. Agents get reward 1 at state s_1 if they choose the same action. Agents get reward 1 at state s_2 if they choose different actions.

basically the same as the two-agent two-state game in Fig. 5. The only difference is we changed the state transition for the state s_1. The new state transition functions for the state s_1 are

$$p(s' = s_2 | s = s_1, a^1 \neq a^2) = 1,$$
$$p(s' = s_1 | s = s_1, a^1 = a^2) = 1. \quad (47)$$

We first consider the stage-wise equilibriums for each state.

For state s_1, the stage-wise equilibrium requires $\Pr(a^1_t = a^2_t) = 1$ for all t. One example of the agent policy is $\pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_1|s_1) = \pi^2(a_1|s_2) = 1$. Note that the agent should have a policy for both s_1 and s_2 even when considering the state-wise equilibrium for the state s_1 (This means the current true state is s_1). This is because the adversary can perturb each agent's state observation to be s_2. There is no requirement for the adversary policy in the state-wise equilibrium because when $\Pr(a^1_t = a^2_t) = 1$, the true state never transits. The state value for s_1 is $V(s_1) = 100$.

Similarly, for state s_2, the stage-wise equilibrium requires $\Pr(a^1_t \neq a^2_t) = 1$ for all t. One example of the agent policy is $\pi^1(a_1|s_1) = \pi^1(a_1|s_2) = \pi^2(a_2|s_1) = \pi^2(a_2|s_2) = 1$. There is no requirement for the adversary policy in the state-wise equilibrium of s_2. The state value for s_2 is $V(s_2) = 100$.

Since the stage-wise equilibriums have conflict requirements for the agent policy in s_1 and s_2, there is no agent policy satisfying the requirements of the stage-wise equilibriums in both s_1 and s_2 at the same time. Therefore, there is no robust Nash equilibrium for agents in this two-agent two-state game. □

We draw a similar conclusion with the Theorem 1. It is not always possible to find a policy to be the stage-wise equilibrium for all the states. When facing adversarial state perturbations, we have to make trade-offs among different states. Therefore, the traditional solution concepts of an optimal agent policy and the robust Nash equilibrium cannot be used for an SAMG.

C.4 Existence of Robust Agent Policy

We need to consider a new objective that is not state-dependent. Therefore, we propose a new objective, the worst-case expected state value, in Definition 4 as

$$\mathbb{E}_{s_0 \sim \Pr(s_0)} \left[V_{\pi,\chi^*(\pi)}(s_0) \right],$$

where $\Pr(s_0)$ is the probability distribution of the initial state. In this section, we show the existence of a robust agent policy to maximize the worst-case expected state value. We first introduce two lemmas for this proof.

Denote $p^{\pi,\chi,s_0}(s_t)$ as the probability of reaching state s_t given the agent policy π, adversary policy χ, and initial state s_0. Let $p^{\pi,\chi,s_0}(s_0) = 1$. The connection between $p^{\pi,\chi,s_0}(s_{t+1})$ and $p^{\pi,\chi,s_0}(s_t)$ is:

$$p^{\pi,\chi,s_0}(s_{t+1}) = \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p(s_{t+1}|s_t, a_t)\pi(a_t|\rho_t)\chi(\rho_t|s_t)p^{\pi,\chi,s_0}(s_t). \quad (48)$$

For a concise representation, we omit the subscript s_t of $\mathcal{P}_{s_t}$ in this section. Consider the function

$$g^{s_0}_t(\pi,\chi) = \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p^{\pi,\chi,s_0}(s_t) \times \pi(a_t|\rho_t)\chi(\rho_t|s_t)\gamma^t r_{t+1}(s_t, a_t). \quad (49)$$

Lemma 7. *The function $g^{s_0}_t$ is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $t = 0, 1, 2, ..., n$ where $n \in \mathbb{N}_+$.*

Proof. To prove the continuity, we construct some equivalent vectors as follows. We define a vector $\vec{\pi} \in \mathbb{R}^{|\mathcal{A}||\mathcal{P}|}$ and $\vec{\pi}(a,\rho) = \pi(a|\rho)$ for $a \in \mathcal{A}, \rho \in \mathcal{P}$, and a vector $\vec{\chi} \in \mathbb{R}^{|\mathcal{P}||\mathcal{S}|}$ where $\vec{\chi}(\rho,s) = \chi(\rho|s)$ for $\rho \in \mathcal{P}, s \in \mathcal{S}$. And a vector constant $\vec{r} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ where $\vec{r}(s,a) = r(s,a)$.

$$\begin{aligned}
\vec{\pi}^\top &= [\pi(a^1|\rho^1), \cdots, \pi(a^{|\mathcal{A}|}|\rho^1), \pi(a^2|\rho^1), \cdots, \pi(a^{|\mathcal{A}|}|\rho^{|\mathcal{P}|})] \\
\vec{\chi}^\top &= [\chi(\rho^1|s^1), \cdots, \chi(\rho^{|\mathcal{P}|}|s^1), \chi(\rho^2|s^1), \cdots, \chi(\rho^{|\mathcal{P}|}|s^{|\mathcal{S}|})] \\
\vec{p}_t &= [p^{\pi,\chi,s_0}(s_t = s^1), \cdots, p^{\pi,\chi,s_0}(s_t = s^{|\mathcal{S}|})] \qquad (50)
\end{aligned}$$

Note that when $\rho \notin \mathcal{P}_s$, then the entry $\chi(\rho|s) = 0$. $\vec{p}_t \in \mathbb{R}^{|S|}$ can be expressed as a linear combination of $\vec{p}_{t-1}, \vec{\pi}$ and $\vec{\chi}$ according to (48). Let's first consider the case $t = 0$,

$$g_0^{s_0}(\pi,\chi) = \sum_{a_0 \in \mathcal{A}} \sum_{\rho_0 \in \mathcal{P}} \pi(a_0|\rho_0)\chi(\rho_0|s_0)r(s_0,a_0) \qquad (51)$$

Function $g_0^{s_0}$ can be expressed as a linear combination of $\vec{r}, \vec{\pi}$ and $\vec{\chi}$. We consider the general case

$$\begin{aligned}
g_t^{s_0}(\pi,\chi) = \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p^{\pi,\chi,s_0}(s_t) \times \\
\pi(a_t|\rho_t)\chi(\rho_t|s_t)\gamma^t r_{t+1}(s_t,a_t). \qquad (52)
\end{aligned}$$

Function $g_t^{s_0}$ can be expressed as a linear combination of $\vec{r}, \vec{p}_t, \vec{\pi}$ and $\vec{\chi}$. Therefore, $g_t^{s_0}$ is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $t = 0, 1, 2, ..., n$ where $n \in \mathbb{N}_+$. □

Lemma 8. *For any $s_0 \in \mathcal{S}$, the series $\{\sum_{t=0}^n g_t^{s_0}(\pi,\chi)\}$, $n = 1, 2, ...,$ converges uniformly on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$.*

Proof. Consider $M_t^{s_0}(\pi,\chi) = \gamma^t R^{max}$, where R^{max} is the largest absolute value of the rewards. We can check that $|g_t^{s_0}(\pi,\chi)| \le M_t^{s_0}(\pi,\chi)$ for $t \ge 0$ as follows.

$$\begin{aligned}
&|g_t^{s_0}(\pi,\chi)| \\
=&\left| \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p^{\pi,\chi,s_0}(s_t)\pi(a_t|\rho_t)\chi(\rho_t|s_t)\gamma^t r_{t+1}(s_t,a_t) \right| \\
\le&\left| \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p^{\pi,\chi,s_0}(s_t)\pi(a_t|\rho_t)\chi(\rho_t|s_t)\gamma^t R^{max} \right| \\
=&\gamma^t R^{max} \times \left| \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p^{\pi,\chi,s_0}(s_t)\pi(a_t|\rho_t)\chi(\rho_t|s_t) \right| \\
=&\gamma^t R^{max} \times \left| \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} \Pr(s_t, a_t, \rho_t \mid s_0, \pi, \chi) \right| \\
=&\gamma^t R^{max} \times 1 = M_t^{s_0}(\pi,\chi). \qquad (53)
\end{aligned}$$

Meanwhile,

$$\sum_{t=0}^{\infty} M_t^{s_0}(\pi,\chi) = \sum_{t=0}^{\infty} \gamma^t R^{max} = \frac{R^{max}}{1-\gamma}, \qquad (54)$$

so $\sum g_t^{s_0}$ converges uniformly on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ according to the Weierstrass M-test in Theorem 7.10 of (Rudin et al. 1976). □

Lemma 8 shows the series $\{\sum_{t=0}^n g_t^{s_0}(\pi,\chi)\}, n = 1, 2, ...,$ converges uniformly on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $s_0 \in \mathcal{S}$. In the following lemma, we show $\sum_{t=0}^{\infty} g_t^{s_0}(\pi,\chi)$ is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $s_0 \in \mathcal{S}$. Denote $h^{s_0}(\pi,\chi) = \sum_{t=0}^{\infty} g_t^{s_0}(\pi,\chi)$.

Lemma 9. *The function h^{s_0} is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $s_0 \in \mathcal{S}$.*

Proof. Consider $h_n^{s_0}(\pi,\chi) = \sum_{t=0}^n g_t^{s_0}(\pi,\chi)$ for $n \in \mathbb{N}_+$. Since $h_n^{s_0}$ is a linear combination of $\{g_t^{s_0}\}_{t=0,1,2,\cdots,n}$ and $g_t^{s_0}$ is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $t = 0, 1, 2, \cdots, n$ according to Lemma 7, the sequence $\{h_n^{s_0}\}$ is a sequence of continuous functions on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$. Meanwhile, $h_n^{s_0} \to h^{s_0}$ uniformly on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $s_0 \in \mathcal{S}$ according to Lemma 8, therefore h^{s_0} is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $s_0 \in \mathcal{S}$ according to the uniform limit theorem in Theorem 7.12 of (Rudin et al. 1976). □

Finally, we show the existence of the robust agent policy to maximize the worst-case expected state value in the following theorem.

Theorem 3. *For an SAMG with finite state and finite action spaces, there exists a robust agent policy π to maximize the worst-case expected state value defined in Definition 4*

Proof. According to Proposition 2, finding an agent policy π to maximize the worst-case expected state value under the optimal adversary for π is equivalent to the following maximin problem:

$$\begin{aligned}
&\max_{\pi} F(\pi) \\
:=&\max_{\pi} \mathbb{E}_{s_0 \sim \Pr(s_0)} \left[V_{\pi,\chi^*(\pi)}(s_0) \right] \\
=&\max_{\pi} \min_{\chi} \sum_{s_0} \Pr(s_0) V_{\pi,\chi}(s_0) \\
=&\max_{\pi} \min_{\chi} J(\pi,\chi), \qquad (55)
\end{aligned}$$

where the objective function in (55) can be expanded as follows:

$$\begin{aligned}
&J(\pi,\chi) \\
=&\mathbb{E}_{s_0 \sim \Pr(s_0)} \left[V_{\pi,\chi}(s_0) \right] \\
=&\sum_{s_0} \Pr(s_0) V_{\pi,\chi}(s_0) \\
=&\sum_{s_0} \Pr(s_0)\, \mathbb{E}_{a_t \sim \pi, \rho_t \sim \chi} \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1}(s_t,a_t) \mid s_0 \right] \\
=&\sum_{s_0} \Pr(s_0) \sum_{t=0}^{\infty} \mathbb{E}_{a_t \sim \pi, \rho_t \sim \chi} \left[\gamma^t r_{t+1}(s_t,a_t) \mid s_0 \right] \\
&\text{(linearity of the expectation)} \\
=&\sum_{s_0} \Pr(s_0) \sum_{t=0}^{\infty} \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \sum_{\rho_t \in \mathcal{P}} p^{\pi,\chi,s_0}(s_t) \times \\
&\pi(a_t|\rho_t)\chi(\rho_t|s_t)\gamma^t r_{t+1}(s_t,a_t) \\
=&\sum_{s_0} \Pr(s_0) \sum_{t=0}^{\infty} g_t^{s_0}(\pi,\chi)
\end{aligned}$$

$$= \sum_{s_0} \Pr(s_0) h^{s_0}. \tag{56}$$

Because $J(\pi, \chi)$ is a linear combination of $\{h^{s_0}\}_{s_0 \in \mathcal{S}}$, $\mathcal{S}$ is finite, and h^{s_0} is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$ for any $s_0 \in \mathcal{S}$ according to Lemma 9, the objective function $J(\pi, \chi) = \sum_{s_0} \Pr(s_0) h^{s_0}$ is continuous on $\Delta(\mathcal{A}) \times \Delta(\mathcal{P})$. Consider the function $F(\pi) = \min_\chi J(\pi, \chi)$. Since the adversary policy space $\Delta(\mathcal{P})$ is compact, the function F is continuous in π. Meanwhile, the agent policy space $\Delta(\mathcal{A})$ is closed. Therefore, there exists an agent policy π to maximize F according to the extreme value theorem.

□

D Implementation Detail

All hyperparameters used in experiments are listed in table 4.

D.1 Environments

We have tested our algorithm in environments provided by (Lowe et al. 2017) as shown in Fig. 7.

Cooperative navigation (CN) This is a cooperative task. There are 3 agents and 3 landmarks. Agents want to occupy/cover all the landmarks. They need to cooperate through physical actions about their preferred landmark to cover. Also, they will be penalized when collisions happen.

Exchange target (ET) This is a cooperative task. There are 2 agents and 3 landmarks. Each agent needs to get to its target landmark, which is known only by another agent. They have to learn communication and get to landmarks. Besides, both of them are generous agents that pay more attention to helping others, i.e. rewarded more if the other agent gets closer to the target landmark.

Keep-away (KA) This is a competitive task. There is 1 agent, 1 adversary, and 1 landmark. The agent knows the position of the target landmark and wants to reach it. The adversary does not know the target landmark and wants to prevent the agent from reaching the target by pushing them away or occupying the target temporarily.

Physical deception (PD) This is a mixed cooperative and competitive task. There are 2 collaborative agents, 2 landmarks including a target, and 1 adversary. Both the collaborative agents and the adversary want to reach the target, but only collaborative agents know the correct target. The collaborative agents should learn a policy to cover all landmarks so that the adversary does not know which one is the true target.

Covert communication (CC) This is a mixed cooperative and competitive task. There are 2 agents, Alice and Bob, and 1 adversary Eve. This environment is from symmetric-key encryption considering eavesdropping attacks in cryptography. Alice and Bob want to transfer a plaintext while Eve can eavesdrop on it. Alice and Bob need to encode their message to ciphertext using a randomly generated secret key.

D.2 Baselines

We compare the performance of our algorithm with MADDPG (Lowe et al. 2017), M3DDPG (Li et al. 2019), and MAPPO (Yu et al. 2021a) and follow their open-source implementation. We have a brief introduction of these methods in the following sections. There is no robustness considered in MADDPG and MAPPO. The M3DDPG considers the robustness of training partner's policies, but it does not consider state uncertainty. The MAPPO is the multi-agent version of the Proximal Policy Optimization (PPO), a popular policy gradient algorithm. Because MAPPO only works in fully cooperative tasks, we only report its results in cooperative navigation and exchange target. Note that MAPPO is also used in (Guo et al. 2020) but they do not provide an open-source implementation. Therefore, we select the latest implementation in (Yu et al. 2021a) with the open-source code.

D.3 Multi-Agent Deep Deterministic Policy Gradient (MADDPG)

It is difficult to apply single-agent RL algorithms directly to the multi-agent case because the environment's state transition is also influenced by the policy of other agents and it is non-stationary from a single agent's view. To alleviate this problem and stabilize training, the MADDPG algorithm is proposed using a centralized Q function that has global state and global action information (Lowe et al. 2017). It assumes all agents are self-interested and every agent's objective is to maximize its own total expected return. The objective for agent i is $J(\theta^i) = \mathbb{E}[R^i]$ and its gradient is

$$\begin{aligned} &\nabla_{\theta^i} J(\theta^i) = \\ &\mathbb{E}_{\mathbf{x}, a \sim \mathcal{D}} \left[\nabla_{\theta^i} \mu^i(o^i) \nabla_{a^i} Q^i(\mathbf{x}, a^1, ..., a^n)|_{a^i = \mu^i(o^i)} \right], \end{aligned} \tag{57}$$

where $Q^i(\mathbf{x}, a^1, ..., a^n)$ is a centralized action-value function, $\mathbf{x} = (o^1, \ldots, o^n)$, and o^i represents agent i's observation. The experience replay buffer $\mathcal{D}$ contains transition experience $\mathbf{x}, a^1, ..., a^n, \mathbf{x}', r^1, ..., r^n$ to decorrelate data. The centralized Q^i can be trained using the Bellman loss:

$$\begin{aligned} &\mathcal{L}(\theta^i) = \mathbb{E}_{\mathbf{x}, a, \mathbf{r}, \mathbf{x}' \sim \mathcal{D}} [y - Q^i(\mathbf{x}, a^1, ..., a^n)]^2, \\ &y = r^i + \gamma Q^{i\prime}(\mathbf{x}', a^{1\prime}, ..., a^{n\prime})|_{a^{j\prime} = \mu^{j\prime}(o^j)}, \end{aligned} \tag{58}$$

where $Q^{i\prime}$ is the target network whose parameters are copied from Q with a delay to stabilize the moving target. Note that this algorithm adopts a centralized training and decentralized execution paradigm. When testing, each agent can only access its local observation to select actions.

In M3DDPG (Li et al. 2019), the uncertainty from the training partner's policies is considered: all other partners are considered as adversaries that select actions to minimize the total expected return of the training agent. In other words, when updating both actor and critic, they select training partner's actions by $a^{j \neq i} = \arg\min_{a^{j \neq i}} Q^i(\mathbf{x}, a^1, ..., a^n)$.

D.4 Gradient Descent Ascent (GDA)

Gradient Descent Ascent (GDA) (Lin, Jin, and Jordan 2020) is currently one widely-used algorithm for solving the fol-

Parameter	RMARL	M3DDPG	MADDPG	MAPPO
optimizer for the critic network	Adam	Adam	Adam	Adam
learning rate for agent policy π	0.01	0.01	0.01	0.0007
learning rate for adversary policy χ	0.001	/	/	/
discount factor	0.95	0.95	0.95	0.99
replay buffer size	10^6	10^6	10^6	/
activation function	Relu	Relu	Relu	Relu
number of hidden layers	2	2	2	1
number of hidden units per layer	64	64	64	64
number of samples per minibatch	1024	1024	1024	1
target network update coefficient τ	0.01	0.01	0.01	/
GDA optimizer steps	20	/	/	/
radius d	1.0	/	/	/
uncertainty level λ	0.5	0.5	0.5	0.5
upper boundary u	1.0	1.0	1.0	1.0
lower boundary l	-1.0	-1.0	-1.0	-1.0
episodes in training	10k	10k	10k	10k
time steps in one episode	25	25	25	25

Table 4: Hyperparameters for our RMARL algorithm and the baselines.

lowing minimax optimization problem:

$$\min_x \max_y \ f(x, y). \tag{59}$$

GDA simultaneously performs gradient descent update on the variable x and gradient ascent update on the variable y according to (60) with step sizes η_x and η_y.

$$\begin{aligned} x_{t+1} &= x_t - \eta_x \nabla_x f(x_t, y_t), \\ y_{t+1} &= y_t + \eta_y \nabla_y f(x_t, y_t). \end{aligned} \tag{60}$$

It has a variety of variants to accommodate different types of geometries of the minimax problem, such as convex-concave geometry, nonconvex-concave geometry, nonconvex-nonconcave geometry, etc.

D.5 Cooperative Navigation With 6 Agents

		CN with 6 agents
MADDPG w/χ^*	μ	-3405.274
	σ	66.18
M3DDPG w/χ^*	μ	-3452.22
	σ	80.16
MAPPO w/χ^*	μ	-3121.90
	σ	18.49
RMARL w/χ^*	μ	**-3079.37**
	σ	**16.16**

Table 5: Mean episode rewards of 2000 episodes during testing under well-trained adversarial state perturbations in the cooperative navigation environment with 6 agents. Our RMARL policy achieves up to 9.57% higher mean episode reward than the baselines with well-trained χ^*.

We compare our RMARL algorithm with baselines in the cooperative navigation scenario with more agents added. The original cooperative navigation environment has 3 agents and the training results are shown in Fig. 2. We show the training results with 6 agents in Fig. 8. After increasing the total number of agents in the environment, our RMARL algorithm still gets higher mean episode rewards than baselines under adversarial state perturbations.

We also test the learned policies in the 6-agent Cooperative Navigation (CN) environment to show our RMARL policy is more robust under adversarial state perturbations. During testing, the mean episode rewards are averaged across 2000 episodes and 10 test runs for each algorithm. We put all the well-trained agents using different algorithms into the 6-agent CN environment with well-trained adversary policies χ^* to perturb states. The result is shown in Table 5. Our RMARL policy achieves up to 9.57% higher mean episode reward than the baselines with well-trained adversarial state perturbations. The result shows that our RMARL algorithm achieves higher robustness for a multi-agent system under adversarial state perturbations.

E Discussions and Future Work

In this section, we add several discussions of our work as a first attempt to study the SAMG problem formally. We also point out several future directions for the SAMG problem.

E.1 GDA Convergence

In Alg. 1, we use Gradient Descent Ascent (GDA) optimizer (Lin, Jin, and Jordan 2020) to update parameters for each agent's actor network and the adversary network. Each agent updates the actor network to maximize the worst-case expected state value in Definition 4, while the corresponding adversary updates the adversary network to minimize the worst-case expected state value. How to solve a non-convex non-concave minimax problem is a very challenging and not yet well-solved problem. To the best of our knowledge, the GDA optimizer is currently one of the most widely used and accepted optimizers for this type of problem, though it

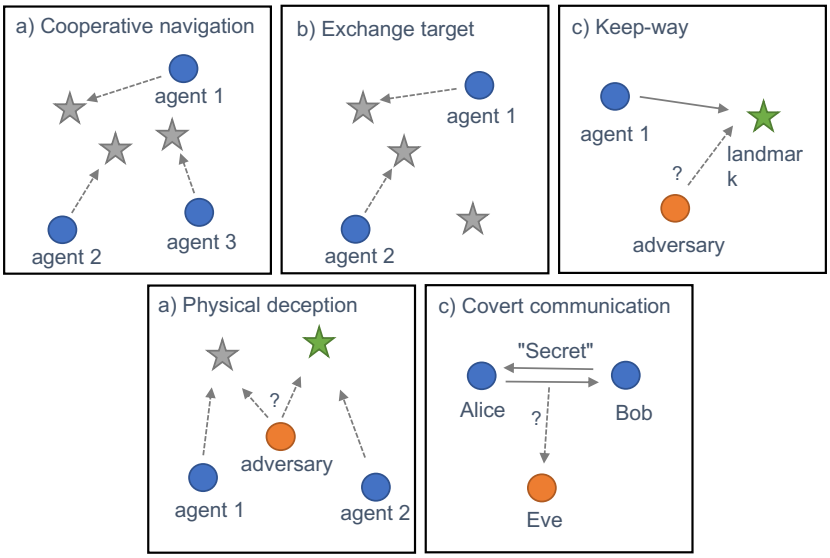

Figure 7: Some environments to test our algorithm, including a) Cooperative navigation (CN) b) Exchange target (ET) c) Keep-away (KA) d) Physical deception (PD) e) Covert Communication (CC).

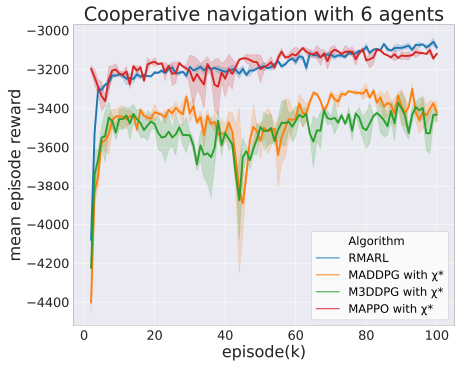

Figure 8: Our RMARL algorithm compared with baselines during the training process in the cooperative navigation scenario with 6 agents added. Our algorithm gets higher mean episode rewards in the environment with an increased agent number.

is not guaranteed to always converge (Jin, Netrapalli, and Jordan 2020; Razaviyayn et al. 2020; Lin, Jin, and Jordan 2020). Our robust MARL (RMARL) algorithm with GDA optimizer shows performance improvement in terms of policy robustness in our experiments. Note that we only use the GDA optimizer as a tool in our algorithm by leveraging the existing literature on solving non-convex non-concave minimax problems. Future advances of numerical algorithms and solvers for this kind of minimax problem will also benefit our algorithm by replacing the GDA optimizer with new advances.

E.2 Non-Markovian Policy

In this work, we give the first attempt to focus on the Markovian policy under adversarial state perturbations. Dealing with the non-Markovian policy will significantly complicate the problem. We are aware of the suboptimality of Markovian policies, however, considering the computational cost of the non-Markovian policy of MARL, we decide to focus on Markovian policies in this work for computational tractability. Moreover, as shown in Proposition 3, our SAMG problem is different from a Dec-POMDP. Considering a non-Markovian policy based on the observation-action history may not give an advantage to the agents. For example, for the two-agent two-state game in Fig. 5, if the adversary randomly perturbs the state with $\chi^i(s_1|s_2) = 0.5$ for $i = 1, 2$, then the agents still only have a 50% chance to guess the true state even with observation-action history. Considering another example for the two-agent two-state game in Fig. 5, if the adversary perturbs all states to state s_1 with $\chi^i(s_1|s_2) = 1$ and $\chi^i(s_1|s_1) = 1$ for $i = 1, 2$, then the agents cannot get extra information for the true state even with observation-action history. We leave the formal analysis of non-Markovian, non-stationary policy as future work.

E.3 Non-collaborative Game

In the problem formulation, we consider a collaborative game, where all agents share one stage-wise reward function. The new objective for the SAMG, the worst-case expected state value under state perturbations, is well-defined as proved in Theorem 3. For non-collaborative games, if each agent has its own reward function, and adversary i wants to minimize the total expected return of agent i, then for a fixed agent policy π, the n adversaries are playing a Markov game. In this case, only the Nash equilibrium exists among n adversaries, but optimal adversary policy may not exist. Therefore, for non-collaborative games, the worst-case expected state value is not well-defined. Even though the worst-case expected state value is not well-defined for non-collaborative games, the experiment results of the competitive games and mixed-cooperative-competitive game environments in Table 2 and Table 3 also show that our RMARL

algorithm can get larger mean episode rewards in non-collaborative games under adversarial state perturbations. Hence, our RMARL algorithm can increase the robustness of policies of non-collaborative games in empirical experiments. We leave the formal analysis of the non-collaborative games as future work.