

Graphical Abstract

Attacking Cooperative Multi-Agent Reinforcement Learning by Adversarial Minority Influence

Simin Li^{a, e}, Jun Guo^a, Jingqiao Xiu^a, Yuwei Zheng^a, Pu Feng^a, Xin Yu^a, Jiakai Wang^b, Aishan Liu^a, Yaodong Yang^d, Bo An^e, Wenjun Wu^a, Xianglong Liu^{a, b, c} ^①

^①Corresponding author: Xianglong Liu. Email address: xliu@buaa.edu.cn.

Highlights

Attacking Cooperative Multi-Agent Reinforcement Learning by Adversarial Minority Influence

Simin Li^{a, e}, Jun Guo^a, Jingqiao Xiu^a, Yuwei Zheng^a, Pu Feng^a, Xin Yu^a, Jiakai Wang^b, Aishan Liu^a, Yaodong Yang^d, Bo An^e, Wenjun Wu^a, Xianglong Liu^{a, b, c} ^①

- We develop AMI, a strong and practical attack towards c-MARL, which leverages the intricate influence among agents and the cooperative nature of victims in c-MARL.
- We introduce the *unilateral influence filter* and *targeted adversarial oracle* to optimize the deviation of victim policies and deceive victims into suboptimal cooperation, thereby ensuring a potent attack capability.
- AMI achieves the first successful attack against real world robot swarms and effectively fool agents in simulation environments into collectively worst-case scenarios, including StarCraft II and Multi-agent Mujoco.

^①Corresponding author: Xianglong Liu. Email address: xlliu@buaa.edu.cn.

Attacking Cooperative Multi-Agent Reinforcement Learning by Adversarial Minority Influence

Simin Li^{a, e}, Jun Guo^a, Jingqiao Xiu^a, Yuwei Zheng^a, Pu Feng^a, Xin Yu^a, Jiakai Wang^b, Aishan Liu^a, Yaodong Yang^d, Bo An^e, Wenjun Wu^a, Xianglong Liu^{a, b, c}^①

^a*State Key Lab of Software Development Environment, Beihang University, Beijing, China*

^b*Zhongguancun Laboratory, Beijing, China*

^c*Institute of Data Space, Hefei Comprehensive National Science Center, Hefei, China*

^d*Institute of Artificial Intelligence, Peking University, Beijing, China*

^e*Nanyang Technological University, Singapore*

Abstract

This study probes the vulnerabilities of cooperative multi-agent reinforcement learning (c-MARL) under adversarial attacks, a critical determinant of c-MARL's worst-case performance prior to real-world implementation. Current observation-based attacks, constrained by white-box assumptions, overlook c-MARL's complex *multi-agent* interactions and *cooperative* objectives, resulting in impractical and limited attack capabilities. To address these shortcomings, we propose *Adversarial Minority Influence* (AMI), a practical and strong for c-MARL. AMI is a practical black-box attack and can be launched without knowing victim parameters. AMI is also strong by considering the complex *multi-agent* interaction and the *cooperative* goal of agents, enabling a single adversarial agent to *unilaterally* misleads majority victims to form *targeted* worst-case cooperation. This mirrors minority influence phenomena in social psychology. To achieve maximum deviation in victim policies under complex agent-wise interactions, our *unilateral* attack aims to characterize and maximize the impact of the adversary on the victims. This is achieved by adapting a unilateral agent-wise relation metric derived from mutual information, thereby mitigating the adverse effects of victim influence on the adversary. To lead the victims into a jointly

^①Corresponding author: Xianglong Liu. Email address: xlliu@buaa.edu.cn.

detrimental scenario, our *targeted* attack deceives victims into a long-term, cooperatively harmful situation by guiding each victim towards a specific target, determined through a trial-and-error process executed by a reinforcement learning agent. Through AMI, we achieve the first successful attack against real-world robot swarms and effectively fool agents in simulated environments into collectively worst-case scenarios, including Starcraft II and Multi-agent Mujoco. The source code and demonstrations can be found at: <https://github.com/DIG-Beihang/AMI>.

Keywords: Multi-agent reinforcement learning, trustworthy reinforcement learning, adversarial attack, algorithmic testing

1. Introduction

Cooperative multi-agent reinforcement learning (c-MARL) involves the coordination of multiple agents to maximize their shared objective in an environment [1, 2, 3, 4, 5, 6, 7]. Applications of c-MARL includes cooperative gaming [8], traffic signal management [9], distributed resource allocation [10], and cooperative swarm control [11, 12, 13, 3, 14, 15].

While c-MARL has achieved notable success, research has exposed the vulnerability of c-MARL agents to observation-based adversarial attacks [16, 17], wherein adversaries introduce perturbations to an agent’s observation, causing it to execute suboptimal actions. Given the interrelated nature of victim actions for cooperation, other c-MARL agents may become disoriented and being non-robust (see Fig. 1). As c-MARL algorithms frequently feature in security-sensitive applications, assessing their worst-case performance against potential adversarial interference is crucial before real world deployment. However, observation-based attacks against c-MARL depend on white-box access to victim and complete control of agent observations, rendering them highly impractical. For instance, in autonomous driving scenarios, it can be prohibitively hard for attackers to access the architectures, weights, and gradients utilized by a vehicle or to introduce arbitrary pixel-wise manipulations to camera input at each timestep [16, 18, 19].

In this paper, we take a practical and black-box alternative by conducting a policy-based adversarial attack (*i.e.*, adversarial policy) [20, 21, 22] to assess the robustness of c-MARL. Unlike direct observation manipulation, policy-based attacks perturb observations in a natural manner by incorporating an adversarial agent within the c-MARL environment—a feasible approach in

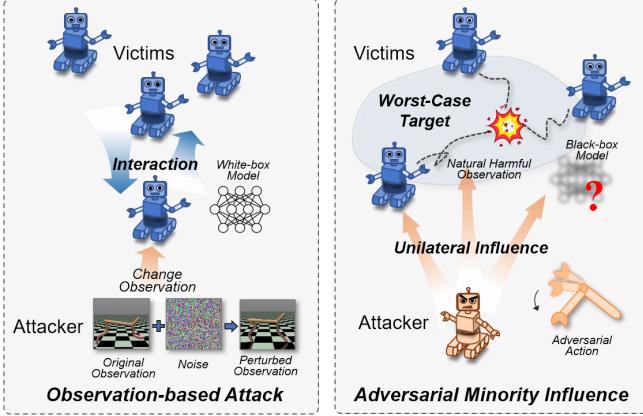


Figure 1: While observation-based attack requires white-box access to victim and manipulates agent observation directly, our adversarial minority influence attack is a black-box, policy-based attack that leverages one minority attacker to unilaterally influence majority victims towards a jointly worst target.

numerous c-MARL applications. For instance, adversaries can legitimately participate in distributed resource allocation clusters as individual workers [23, 10] or control their own vehicles while influencing other vehicles in autonomous driving [24, 25]. By interacting with the environment, the adversary learns an adversarial policy that directs it to execute physically plausible actions, which in turn adversely influence the observations of victim agents.

Although policy-based attacks have been investigated in two-agent competitive games [20, 21, 22], these studies have neglected two crucial challenges in c-MARL, resulting in diminished attack efficacy. Ideally, the adversary should *influence* victims toward a *cooperatively* inferior policy. This gives rise to two issues: (1) The *influence* problem, wherein all agents mutually influence one another in c-MARL; consequently, attacking a single victim impacts the policies of other victims as well. In the face of such intricate agent-wise influence, it becomes difficult for the adversary to maximally deviate victim policies and identify the optimal attack strategy. (2) The *cooperation* problem, which arises as merely perturbing victim actions arbitrarily or toward locally suboptimal cases is insufficient to indicate the failure of cooperative victims. The adversary faces the challenge of exploring and deceiving victims into a long-term, jointly detrimental failure scenario.

To address these challenges, we introduce *Adversarial Minority Influence* (*AMI*), a black-box policy-based attack for c-MARL. Shown in Fig. 1, *AMI* differs from other attacks by deceiving victims in an *unilateral* and *targeted*

manner. The term *minority influence* originates from social psychology [26], wherein minorities (adversary) can *unilaterally* sway majorities (victims) to adopt its own *targeted* belief. Technically, to maximally deviate victim policies amidst complex agent interactions, we quantify and maximize the adversarial impact from the adversary to each victim. Our *unilateral influence filter* adapts mutual information, a bilateral agent-wise relation metric, into a unilateral relation metric that represents the influence from the adversary to the victim. This is achieved by decomposing mutual information and filtering out the harmful influence from victims to the adversary. To deceive victims into a jointly worst-case failure, we learn a long-term, worst-case target for each victim, which leads to collective failure. The target is determined by the *targeted adversarial oracle*, a reinforcement learning agent that generates worst-case target actions for each victim by co-adapting with our adversarial policy. Ultimately, the attacker modifies its policy to direct victims toward these adversarial target actions through unilateral influence. Our **contributions** are listed as follows:

- We develop AMI, a strong and practical attack towards c-MARL, which leverages the intricate influence among agents and the cooperative nature of victims in c-MARL.
- We introduce the *unilateral influence filter* and *targeted adversarial oracle* to optimize the deviation of victim policies and deceive victims into suboptimal cooperation, thereby ensuring a potent attack capability.
- AMI achieves the first successful attack against real world robot swarms and effectively fool agents in simulation environments into collectively worst-case scenarios, including StarCraft II and Multi-agent Mujoco.

2. Related Work

2.1. Overview of Adversarial Attacks

Initially proposed in the field of computer vision, adversarial attacks consist of carefully crafted perturbations that, while imperceptible to humans, can deceive deep neural networks (DNNs) into making incorrect predictions [27, 28, 29]. Given a DNN F_θ , a clean image \mathbf{x} , a perturbed image \mathbf{x}_{adv} , and the ground truth label y , an adversarial example can be formulated as follows:

$$F_\theta(x_{adv}) \neq y \quad s.t. \quad \|\mathbf{x} - \mathbf{x}_{adv}\| \leq \epsilon. \quad (1)$$

In this formulation, $\|\cdot\|$ represents a distance metric used to constrain the distance between \mathbf{x} and \mathbf{x}_{adv} by ϵ . Subsequently, it was demonstrated that reinforcement learning (RL) is also susceptible to adversarial attacks [18, 30, 31]. Owing to the sequential decision-making nature of RL, adversarial attacks in this context aim to generate a perturbation policy π^α that minimizes the victim's cumulative reward $\sum_t \gamma^t r_t$, which can be expressed as:

$$\min_{\pi^\alpha} \sum_t \gamma^t r_t. \quad (2)$$

Adversarial attacks are important to distinguish as test-time attacks, where the adversary targets a specific victim without participating in the training process. As another line of research, training-time attacks interfere with victim training, resulting in trained victims either failing to perform well (poisoning attack) [32, 33] or executing adversary-specified actions when specific triggers are present (backdoor attack) [34, 35, 36]. Note that our method is a test-time attack, and is not related to training-time attacks.

2.2. RL Attacks by Observation Perturbation

Test-time perturbation of RL observations can deceive the policy of RL agents, causing them to execute suboptimal actions and fail to achieve their goals. For single-agent RL attacks, early research employed heuristics such as preventing victims from selecting the best action [18] or choosing actions with the lowest value at critical time steps [30, 31]. Later work framed the adversary and victim within an MDP [19], enabling the optimal observation perturbation to be learned as an action within the current state using an RL agent [37, 38]. For c-MARL attacks, Lin et al. [16] proposed a two-step attack that first learns a worst-case attack policy and then employs a gradient-based attack [39] to execute it. [17] generated attacks on one victim and transfer it to the rest of the victims. However, they assume attacker can modify victim observations and has white-box access to victim parameters, which can be impractical in real world. Another line of research, termed adversarial communication, targets communicative c-MARL by sending messages to victim agents that cause failure upon receiving the adversarial message. Adversarial messages can be added to representations [40] or learned by the adversary [41, 42, 43]. However, these methods are inapplicable when victim agents do not communicate, a common assumption in many mainstream c-MARL algorithms [1, 3, 2].

2.3. RL Attacks by Adversarial Policy

Distinct from observation-based attacks, adversarial policy attacks do not necessitate access to victim observations or parameters (black-box). Rather, they introduce an adversarial agent whose actions are designed to deceive victim agents, causing them to take counterintuitive actions and ultimately fail to achieve their goals. In this paper, the terms “policy-based attack” and “adversarial policy” are used interchangeably. Gleave et al. [20] were the first to introduce adversarial policy in two-agent zero-sum games. This approach was latter applied to multi-agent consensus game, where agents have similar, but non-identical objectives [44]. Subsequent research has enhanced adversarial policy by exploiting victim agent vulnerabilities. Wu et al. [21] induced larger deviations in victim actions by perturbing the most sensitive feature in victim observations, identified through a saliency-based approach. However, larger deviations in victim actions do not necessarily correspond to strategically worse policies. Guo et al. [22] extended adversarial policies to general-sum games by simultaneously maximizing the adversary’s reward and minimizing the victim’s rewards. Yet, none of these studies have considered adversarial policies in c-MARL settings. To better evaluate the performance of our attack in multi-agent adversarial policy scenario, we adapt some observation-based attack in MARL [17] as baselines.

3. Problem Formulation

We conceptualize adversarial attacks targeting c-MARL agents within the framework of a partially observable Stochastic game (POSG) [45], which can be characterized by a tuple:

$$\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \mathcal{O}, O, \mathcal{A}, \mathcal{T}, R, \gamma \rangle. \quad (3)$$

Specifically, $\mathcal{N} = \{\mathcal{N}^\alpha, \mathcal{N}^\nu\} = \{1, \dots, N\}$ is the set containing N agents. Throughout this paper, we use α for adversaries and ν for victims. \mathcal{S} is the global state space, $\mathcal{O} = \times_{i \in \mathcal{N}} \mathcal{O}^i$ is the observation space, O is the observation emission function, $\mathcal{A} = \times_{i \in \mathcal{N}} \mathcal{A}_i$ is the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ is the state transition probability. The reward $R^\alpha : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is shared for all adversaries and $R^\nu : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is shared for all victims. $\gamma \in [0, 1)$ is the discount factor.

At each timestep, each agent i observes $o_{t,i} = O(s_t, i)$ and add it to history $h_{t,i} = [o_{0,i}, a_{0,i}, \dots, o_{t,i}]$ to alleviate partial observability [46]. For victims,

we assume its joint policies $\pi^\nu(\mathbf{a}^\nu|\mathbf{h}) = \prod_{i \in \mathcal{N}^\nu} \pi^\nu(a_i^\nu|h_i)$ are fixed during deployment [47] and take joint actions according to joint policies $\mathbf{a}^\nu \sim \pi^\nu(\cdot|\mathbf{h})$. For adversary, it takes actions by an adversarial policy $a^\alpha \sim \pi^\alpha(\cdot|h_i)$. Next, the game environment proceeds to the next state via transition probability $\mathcal{T}(s_{t+1}|s_t, a^\alpha, \mathbf{a}^\nu)$ and receive the reward for adversary, $r_t^\alpha = R^\alpha(s_t, a_t^\alpha, \mathbf{a}_t^\nu)$. The objective of the adversary is to learn an adversarial policy π^α to maximize its expected discounted cumulative reward, *i.e.*, $J(\pi^\alpha) = \mathbb{E}_{s, a^\alpha, \mathbf{a}^\nu} [\sum_t \gamma^t r_t^\alpha]$. Since defender policies are fixed, the attacker must solve a reinforcement learning problem, but can exploit the weakness in policies of other agents by maximally deviating victims into a jointly worst-case situation.

Assumptions. To keep our attack practical, we assume attacker cannot manipulate victim observations or choose which agent to control. Additionally, attacker does not have the models (*i.e.*, architectures, weights, gradients) and rewards of victims. Following the centralized training, decentralized execution (CTDE) paradigm for c-MARL attack [16, 40, 42, 43], we posit that adversaries have access to the state and reward only during training. However, during actual attack deployment, they must rely solely on their local observation histories. To implement our AMI attack in physical environment, we adopt a Sim2Real paradigm [47], such that the adversary first trains its adversarial policy within a simulated environment, then freezes the policy’s parameters and deploys the attack to real-world robots.

Applications. Our adversary can be launched as a malicious participant in a multi-agent distributed system [23, 10]. Adversaries can also hijack an agent directly [48, 49, 50] and utilize the controlled agent to take an adversarial policy. Apart from malicious use, our AMI attack also functions as a testing algorithm to evaluate the worst-case robustness of c-MARL algorithms, which helps managing algorithmic risks before deploying it in risk-sensitive applications.

4. Method

As previously mentioned, adversarial policy towards c-MARL presents two challenges: the *influence* challenge, which necessitates the adversary to maximize victim policy deviations under intricate agent-wise interactions; and the *cooperation* challenge, which requires the adversary to deceive agents into jointly worst failures. In this paper, we address these challenges through our *adversarial minority influence (AMI)* framework. As depicted in Fig. 2, AMI achieves potent attack capabilities by *unilaterally* guiding victims

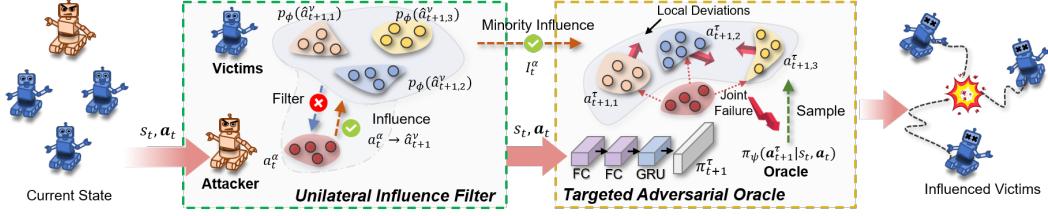


Figure 2: Framework of AMI. Unilateral influence filter decompose mutual information into minority influence and majority influence terms, while keeping latter for asymmetric influence. Targeted adversarial oracle is an RL agent that generates a worst-case target for each victim. Attacking victims towards this target results in jointly worst cooperation.

towards a *targeted* worst-case scenario. To maximize policy deviation of victims, we propose the *unilateral influence filter* that characterizes the adversarial impact from the adversary to victims by decomposing the bilateral mutual information metric and eliminating the detrimental influence from victims to the adversary. To steer victims towards jointly worst actions, the *targeted adversarial oracle* is a reinforcement learning agent that co-adapts with the adversary and generates cooperatively worst-case target actions for each victim at every timestep.

4.1. Unilateral Influence Filter

In a c-MARL attack, due to the intercorrelated nature of agent policies, targeting one victim inevitably impacts the policies of other victims as well. In light of such intricate relationships, maximizing policy deviations for victims necessitates that the adversary first characterizes the influence of its actions on each victim before maximizing that influence. Consequently, to delineate the unilateral influence from the adversary to the victim, we draw inspiration from the minority influence theory in social psychology [26].

Using mutual information as a starting point, we emphasize that the policies of the adversary and the victims are interdependent. However, mutual information fails to fully reflect the deviation in victim policy as it includes the reverse influence from the victims to the adversary, which is counterproductive for the attack. By focusing on the one-way influence from adversary to victims only, our approach aligns with the principles of minority influence, where a small, focused group can effectively influence the behavior of a larger group.

We begin by examining mutual information, a commonly employed bilateral influence metric in the c-MARL literature [51, 52, 53, 54], which captures the relationship between agents. For instance, considering social

influence [51], the mutual information between the adversary action a_t^α and the influenced victim action $a_{t+1,i}^\nu$ can be expressed as $I(a_t^\alpha; a_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu)$, where a_t^α denotes the action taken by adversary α at time t , and $a_{t+1,i}^\nu$ represents the action taken by the i^{th} victim ν at time $t + 1$. Since victim observation and parameter is unknown, the action probability of $a_{t+1,i}^\nu$ can be approximated by network p_ϕ in a supervised learning objective, *i.e.*, $\max_\phi \sum_{i=1}^{\mathcal{N}^\nu} \sum_{t=0}^{T-1} \log p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t)$, where T denotes the total timesteps of an episode. In the remainder of our paper, we use \hat{a} to signify that the action is predicted, rather than ground truth.

Although mutual information is capable of characterizing agent-wise influence in c-MARL, it is important to note that *attacks towards c-MARL differ from cooperation due to the fixed nature of victim parameters*, which renders victim actions more challenging to modify compared to those of the adversary. To demonstrate the effects of fixed victim policy, we decompose the mutual information between the adversary action a_t^α and a victim agent $a_{t+1,i}^\nu$ as follows:

$$I(a_t^\alpha; \hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) = \underbrace{-H(\hat{a}_{t+1,i}^\nu | s_t, a_t^\alpha, \mathbf{a}_t^\nu)}_{\text{majority}} + \underbrace{H(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu)}_{\text{minority}}. \quad (4)$$

In the context of a c-MARL attack, we refer to the first term as *majority influence*, that is, the extent to which the minority (attacker) adapts its policy to conform with the policies of the majority (victims). In order to maximize mutual information, $H(\hat{a}_{t+1,i}^\nu | s_t, a_t^\alpha, \mathbf{a}_t^\nu)$ should be *minimized*, such that having knowledge of a_t^α reduces the uncertainty surrounding the victim policy. In policy-based attacks within c-MARL, the majority influence term in mutual information causes attackers to comply with victim policies, yielding high mutual information but weak attack capability. To elucidate this, consider that the parameters of victims are fixed, whereas the adversary policy is learned. We assume that it is significantly more straightforward to modify the adversary's policy than the victim's policy. Consequently, in order to minimize majority influence, the most effective approach for an attacker is to adjust its action a_t^α to render it more predictive of victim actions, without actually altering it.

Simultaneously, the second term, $H(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu)$, can be interpreted as a form of *minority influence* devoid of victim impact, reflecting solely the

adversary's effect on victims. This term accounts for the entropy of the victim policy without conditioning on a_t^α , thereby establishing an *unilateral* metric. Given that the influence of the adversary action a_t^α is marginalized, the adversary is unable to modify its action to cater to the policy of victims.

Building upon the aforementioned discourse, we examine and generalize minority influence within mutual information to enhance attack capabilities. In particular, maximizing minority influence requires current adversary policy to have large uncertainty in victim policy:

$$\begin{aligned} H(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) &= -D_{KL}(p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) || \mathcal{U}) + c \\ &= -D_{KL}(\mathbb{E}_{\tilde{a}_t^\alpha \sim \pi_t^\alpha} [p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu)] || \mathcal{U}) + c, \end{aligned} \quad (5)$$

see Appendix A for detailed derivation. In the equation, \mathcal{U} represents a uniform distribution for victim actions applicable to both discrete and continuous action spaces, D_{KL} is the Kullback-Leibler divergence, c is a constant, \tilde{a}_t^α refers to the counterfactual action sampled from the adversarial policy π^α .

Equation A.1 shows the second term is equivalent to minimizing the KL divergence between the victim policy and the uniform distribution. For a more general relation metric between adversary and victim, we can release the constraints of Eqn. A.1 and replace \mathcal{U} by a worst-case target distribution \mathcal{D} and generalize D_{KL} to any distance metric $d(\cdot, \cdot)$. Consequently, the unilateral influence can be expressed as:

$$I = d(\mathbb{E}_{\tilde{a}_t^\alpha \sim \pi^\alpha} [p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu)], \mathcal{D}). \quad (6)$$

In this manner, influencing victims unilaterally is tantamount to minimizing the distance between the expected victim policy under the adversary policy and a target distribution.

4.2. Targeted Adversarial Oracle

Upon elucidating the maximization of victim policy deviations through unilateral influence, it remains crucial to ensure that victims are guided toward an optimal target. Merely deviating the victim policy arbitrarily or in the direction of a locally inferior case does not guarantee a globally worst-case failure for c-MARL. To enhance attack capability, it is necessary to ascertain globally worst target actions for victims and subsequently steer each victim towards its target \mathcal{D} (Eqn. 6) using the proposed unilateral influence approach. Consequently, we introduce a reinforcement learning agent that

learns these jointly worst target actions by co-adapting its policy with the attacker in a trial-and-error process.

To accomplish the targeted attack objective, we introduce the *Targeted Adversarial Oracle (TAO)*, a reinforcement learning agent π^τ that guides the attacker to influence each victim toward their globally worst-case direction. To achieve this, TAO uses global state as input, and is used to guide the attacker *only* in training. During execution, the attacker acts on its own without the guidance of TAO. As an RL agent, TAO adjusts to the current perturbation budget of the adversary through a trial-and-error process: if the adversary can significantly impact victim policies, TAO can generate the target more aggressively, directing victims to undertake riskier actions, thus achieving greater attack capability; conversely, if the adversary has limited influence on victims, TAO strives to introduce minor yet effective perturbations to victims, which is feasible under the current perturbation budget. Analogous to the attacker, TAO's objective is to maximize the adversary's goal. We define the following Bellman operator \mathcal{B}^τ to update the value function of TAO:

$$(\mathcal{B}^\tau Q^\tau)(s, a^\tau, a^\alpha) = r_t^\alpha + \gamma \sum_{s'} \mathcal{T}(s'|s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \\ \pi(a'^\tau|s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha|h'_i) \pi^\nu(\mathbf{a}'^\alpha|\mathbf{h}') Q^\tau(s', a'^\tau, a'^\alpha). \quad (7)$$

To avoid non-stationarity, Q^τ also depends on a^α , the adversary's action. While a^τ does not directly interact with the environment, the attacker's policy is *implicitly* influenced by a^τ through the unilateral influence term added to attacker's reward. Assuming the space of state, action of TAO and adversary are finite, we can proof \mathcal{B}^τ is a contraction operator. Consequently, updating $Q^\tau(s, a^\tau, a^\alpha)$ by \mathcal{B}^τ will converge to optimal value $Q^{\tau,*}(s, a^\tau, a^\alpha)$. See the proof in Appendix B.

In practice, we use proximal policy optimization (PPO) [55] to optimize the policy of TAO. The value function used by PPO is an advantage function A_t^τ , with the policy of TAO π^τ updated by PPO objective:

$$\max_{\pi^\tau} \mathbb{E}_{\pi^\tau} [\min (\text{clip} (\rho_t, 1 - \epsilon, 1 + \epsilon) A_t^\tau, \rho_t A_t^\tau)], \\ \text{where } \rho_t = \frac{1}{N^\nu} \sum_{i=1}^{|\mathcal{N}^\nu|} \frac{\pi^\tau(a_{t+1,i}^\tau | s_t, \mathbf{a}_t^\nu, a_t^\alpha)}{\pi_{old}^\tau(a_{t+1,i}^\tau | s_t, \mathbf{a}_t^\nu, a_t^\alpha)}, \quad (8)$$

where π_{old}^τ and π^τ represent the previous and updated policies for TAO, respectively. A_t^τ is the advantage function determined by the generalized advantage estimation (GAE) [56]. The function $\text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)$ constrains the input ρ_t within the limits of $1-\epsilon$ and $1+\epsilon$. $|\mathcal{N}^\nu|$ is the number of victims. In this case, the policy $\pi^\tau(a_{t+1,i}^\tau | s_t, \mathbf{a}_t^\nu, a_t^\alpha)$, as determined by TAO, functions as the desired target distribution \mathcal{D} in Eqn. 6 for victim i at time t . Notably, as PPO is an on-policy algorithm, we derive \mathcal{D} by sampling a target action $a_{t+1,i}^\tau$ from π^τ instead.

4.3. Overall Training

In summary, the influence metric I_t^α used by AMI at time t combines the optimal target action $a_{t+1,i}^\tau$ generated by TAO as the target distribution \mathcal{D} for unilateral influence in Eqn. 6:

$$I_t^\alpha = \sum_{i=1}^{N^\nu} \left[d\left(\mathbb{E}_{\tilde{a}_t^\alpha \sim \pi^\alpha} \left[p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu) \right], a_{t+1,i}^\tau \sim \pi^\tau(\cdot | s_t, a_t^\alpha, \mathbf{a}_t^\nu) \right) \right], \quad (9)$$

In the case of $d(\cdot, \cdot)$ as the distance function for AMI, the calculation differs depending on the control setting. For *discrete control*, the distance function is computed as $d(p(\hat{a}_i^\nu | s, \mathbf{a}), a_i^\tau) = -||p(\hat{a}_i^\nu | s, \mathbf{a}) - \mathbf{1}_{\mathcal{A}}(a_i^\tau)||_1$, where $p(\hat{a}_i^\nu | s, \mathbf{a})$ is a shorthand for $\mathbb{E}_{\tilde{a}_i^\alpha \sim \pi^\alpha} [p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu)]$ and a_i^τ denotes $a_{t+1,i}^\tau$. $\mathbf{1}_{\mathcal{A}}(a_i^\tau) = [a_i^\tau \in \mathcal{A}_i^\tau]$ is the indicator function with action space \mathcal{A}_i^τ . A negative sign is added to minimize the distance between the one-hot target action $\mathbf{1}_{\mathcal{A}}(a_i^\tau)$ and the victim action probability. In the case of *continuous control*, the distance function is calculated as $d(p(\hat{a}_i^\nu | s, \mathbf{a}), a_i^\tau) = p(a_i^\tau | s, \mathbf{a})$, such that the target action $a_{t+1,i}^\tau$ has a high probability in the estimated victim action probability distribution. As verified in ablations, while we tune the distance functions for best performance, our method is not sensitive to the choice of distance functions.

In the final step, I_t^α serves as an auxiliary reward that is optimized by the policy of the adversarial agent $\pi^\alpha(a_t^\alpha | h_t^\alpha)$. The reward r_t^{AMI} for the adversarial agent π_θ to optimize can be expressed as:

$$r_t^{AMI} = r_t^\alpha + \lambda \cdot I_t^\alpha, \quad (10)$$

In this case, λ is a hyperparameter that balances the trade-off between the adversary reward r_t^α and maximizing the influence I_t^α on the victim agents.

We define the following Bellman operator \mathcal{B}^α to update the value function of adversary's policy:

$$\begin{aligned}
(\mathcal{B}^\alpha Q^\alpha)(s, a^\tau, a^\alpha) = & r^\alpha + \lambda \cdot I^\alpha + \gamma \sum_{s'} \mathcal{T}(s'|s, a^\alpha, \mathbf{a}') \\
& \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau|s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha|h'_i) \pi^\nu(\mathbf{a}'^\alpha|\mathbf{h}') Q^\alpha(s', a'^\tau, a'^\alpha).
\end{aligned} \tag{11}$$

Here, the dependency on a^τ is added to avoid non-stationarity. Using similar techniques for proofing the convergence of \mathcal{B}^τ , we can proof \mathcal{B}^α is a contraction operator, thus updating $Q^\alpha(s, a^\tau, a^\alpha)$ via \mathcal{B}^α converge to the optimal value $Q^{\alpha,*}(s, a^\tau, a^\alpha)$. See the proof in Appendix. Appendix C.

In practice, we compute the advantage function A_t^α using r_t^{AMI} via GAE [56] and train the adversary using PPO [55]:

$$\begin{aligned}
& \max_{\pi^\alpha} \mathbb{E}_{\pi^\alpha} [\min (\text{clip} (\rho_t, 1 - \epsilon, 1 + \epsilon) A_t^\alpha, \rho_t A_t^\alpha)], \\
& \text{where } \rho_t = \frac{\pi^\alpha(a_t^\alpha|h_t^\alpha)}{\pi_{old}^\alpha(a_t^\alpha|h_t^\alpha)}.
\end{aligned} \tag{12}$$

The complete training procedure is outlined in Algorithm 1.

5. Experiments

In this section, we perform comprehensive experiments in both simulated and real-world environments to assess the effectiveness of our AMI approach in terms of attack capability.

5.1. Experimental Setup

5.1.1. Environments

We assess the effectiveness of AMI in three distinct environments: (1) A real-world multi-robot rendezvous environment, in which robot swarms learn to gather together, known as *rendezvous* [12]. (2) StarCraft Multi-Agent Challenge (SMAC) [8], involving discrete control across six tasks, where the objective is to control a group of agents in a StarCraft game to defeat an opposing group; (3) Multi-Agent Mujoco (MAMujoco) [11] for continuous control, comprising six tasks that require controlling robotic joints to optimize speed in a specific direction. All victim policies were trained using MAPPO [3]. During the attack, the first agent is selected as the adversary.

Algorithm 1 Adversarial Minority Influence Algorithm.

Input: Policy of victims π^ν , adversary π^α and targeted adversarial oracle (TAO) π^τ . Value function of adversary V^α and TAO V^τ . Opponent modelling network p_ϕ .

Output: Trained policy network of adversary agent π_θ .

- 1: **for** $k = 0, 1, 2, \dots, K$ **do**
 - 2: Perform rollout using current adversarial policy network π^α and TAO agent π^τ . Collect a set of trajectories $\mathcal{D}_k = \tau_i$, where $i = 1, 2, \dots, |\mathcal{D}_k|$.
 - 3: Update opponent modelling model p_ϕ .
 - 4: Calculate advantage function of TAO A_t^τ by GAE, using value function V^α and adversary reward r^α ; Update value function network V^τ of TAO.
 - 5: Update the policy network π^τ of TAO using Eqn. 8;
 - 6: Calculate reward r_t^{AMI} for adversary by Eqn. 10.
 - 7: Calculate advantage function A_t^α by GAE, using value function V^α and reward r^{AMI} ; Update value function network V^α of adversary.
 - 8: Update policy network π^α of adversary by Eqn. 12.
 - 9: **end for**
-

5.1.2. Compared methods and evaluation metrics

We benchmark AMI against state-of-the-art adversarial policy methods, including single-agent adversarial policy, *Gleave et al.* [20], *Wu et al.* [21], *Guo et al.* [22] and multi-agent attack GMA [17]. We adapt adversarial policy for single-agent case to multi-agent by substituting a single RL victim with multiple RL victims. As for multi-agent observation-based GMA

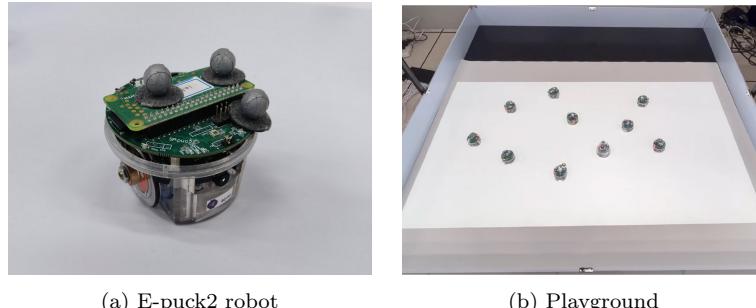


Figure 3: Illustration of the robot and playground for our real-world multi-robot rendezvous environment.

attack, we allow their attack to manipulate victim action arbitrarily, same as our setting. To ensure a fair comparison, AMI and all baselines employ the same codebase, network structure, and hyperparameters. For method-specific hyperparameters, we tune their values for optimal performance. The adversary’s goal is to maximize the adversarial reward r^α , defined as (1) maximizing the loss of allies and minimizing the loss of enemies for SMAC. (2) minimizing the speed of agents in $+x$ direction for MAMujoco. (3) maximizing the euclidean distance between all agents for rendezvous. All experiments were conducted with five random seeds, and results are presented with a 95% confidence interval. The hyperparameters used for all experiments are listed in Appendix. Appendix D.

5.2. AMI Attack in Real World

To highlight the effectiveness of AMI, we evaluate AMI in real-world multi-robot environments. To the best of our knowledge, this is the first evaluation of adversarial policy in real world. As shown in Fig. 3, we create an environment with 10 e-puck2 robots (Fig. 3a) [57] in an indoor playground (Fig. 3b). The task is called *rendezvous*, where robots are randomly dispersed in the arena and must gather together.

We train these robots using the widely adopted Sim2Real paradigm [47] in the RL community, in which agents first learn their policy in a simulated environment before being deployed in the real world with fixed parameters. To evaluate attack performance, we present results from both simulated and real-world scenarios in Fig. 4. Each method was tested 10 times in the real world, leading to several key findings:

(1) AMI outperforms all baselines in both simulated and real-world environments. The improvement of AMI in the real world is statistically signifi-

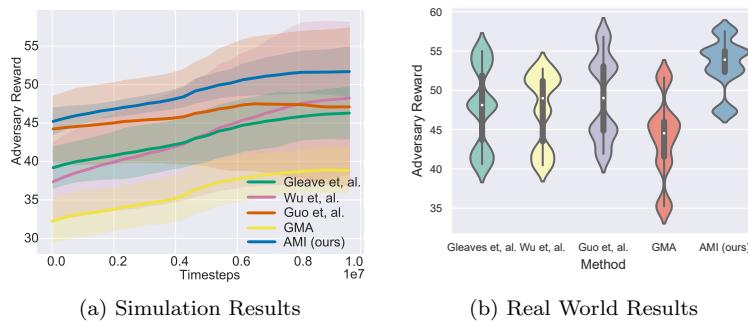


Figure 4: Comparisons of AMI against baselines in simulation and real-world experiments.

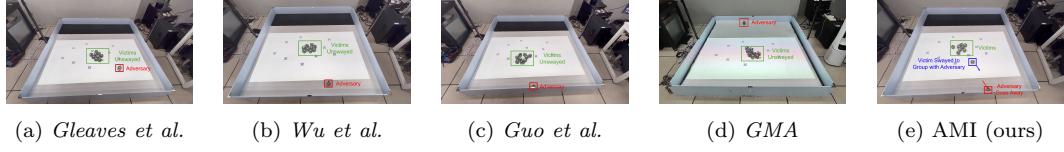


Figure 5: Behaviors of robot swarms under our AMI attack, adversary indicated by red square. Our adversary is the only one to fool away an agent to group with our adversary.

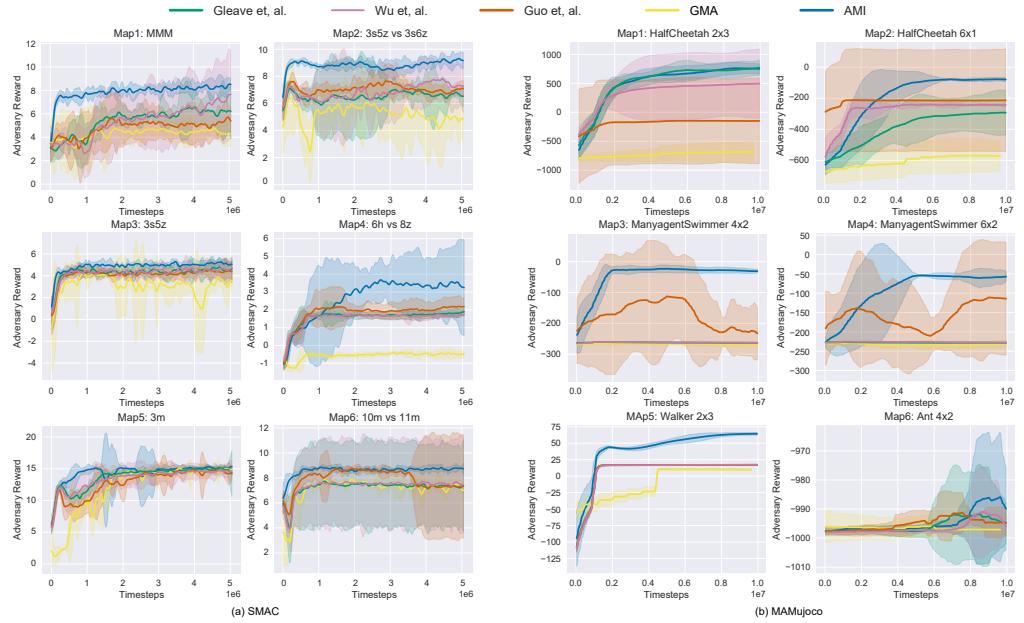


Figure 6: Learning curves for AMI and baseline attack methods in six SMAC and six MAMujoco environments. The solid curves correspond to the mean, and the shaped region represents the 95% confidence interval over 5 random seeds.

cant ($p < .05$) compared to all baselines under a paired samples t-test and, on average, 5.43 higher than the best-performing baseline in the real world.

(2) The superiority of AMI can be further demonstrated by agent behaviors. As shown in the final state photographed in Fig. 5, under baseline attacks, victims gather together as usual without being influenced, with adversaries moving away (Fig. 5a-5d). However, with our AMI attack, one victim gets influenced by the adversary, and are fooled to gather with the adversary, instead of majority victims (Fig. 5e). Notably, our AMI is the only method to achieve this. *See video demonstrations in <https://github.com/DIG-Beihang/AMI>.*



Figure 7: Understanding the behavior of AMI. Attacker and victims in red, enemies in blue. (a) attackers entice victims to get back. (b) victims were influenced into a bad position. (c) some victims encounter enemies, while others are still moving. (d) first-arrived victims died, and enemies focused fire on the rest. (e) attacker moves near to get killed.

5.3. AMI Attack in Simulation Environment

Apart from real world results, we further evaluate the effectiveness of AMI in 12 tasks in simulated environments for completeness, including six discrete control tasks (SMAC) and six continuous control tasks (MAMu-joco), demonstrating its superior performance. For simplicity, we assume the attacker controls the first agent in all environments. As shown in Fig. 6, by strategically influencing victims toward a jointly worst target, our AMI outperforms the competing methods in 10 out of 12 environments across both continuous and discrete control, highlighting the effectiveness of our approach.

5.4. Analysis of AMI policies

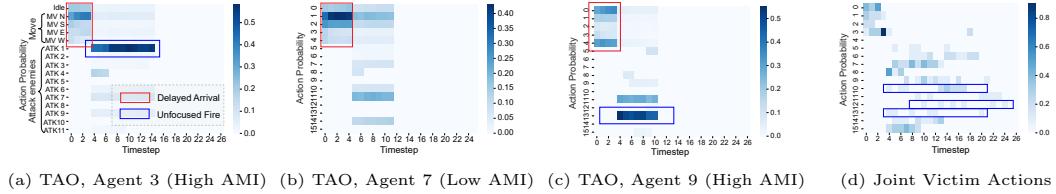


Figure 8: Actions suggested by TAO and taken by victims, evaluated in $10m$ vs $11m$. Red and blue bracket indicates delayed arrival and unfocused fire behavior of AMI. MV N means move north and ATK X means attack enemy ID X.

To investigate the attack behavior of AMI, we visualize the behavior of victims subjected to AMI attacks in $10m$ vs $11m$ environment of SMAC, shown in Fig. 7 and 8.

Victim behavior under AMI. Victims under AMI attack exhibit two critical behaviors that contributes to collective failure: (1) *delayed arrival*. As depicted in Fig. 7, victims are influenced by the attacker and divided into two groups. These groups encounter enemies at different timesteps:

while some victims confront the full force of their adversaries, others are still approaching and do not have enemies within their firing range. (2) *unfocused fire*. In SMAC, focused fire involves allies collaboratively attacking and defeating enemies one at a time. However, in the presence of the attacker, allies fire in an *unfocused* manner. As illustrated in Fig. 8d, victims’ shots at enemies (Action ID 5-15) are randomly distributed, lacking a coordinated target. Consequently, victims fail to eliminate enemy units and face stronger enemy fire.

Influence through the lens of TAO. The behavior of victims under AMI can be explained by the target actions generated by TAO. As demonstrated in Fig. 8, for understanding *delayed arrival*, at the game’s onset, victims 3 and 7 are encouraged to move north (Action ID 1), arriving later compared to agents moving directly east; for victim 9, which is moving east, it is encouraged to move west or stay idle: had it followed this target, it would have arrived slightly later than agents moving east directly (*i.e.*, not being influenced), but still earlier than agents moving north, also resulting in delayed arrival. To comprehend *unfocused fire*, agents 3, 7, and 9 are encouraged to attack *different* enemies at the current timestep, limiting the number of victims attacking each enemy and suppressing focused fire.

Adaptive target generation. Furthermore, we discover that TAO can generate adaptive policies for victims with varying susceptibility, as illustrated in Fig. 8. By calculating AMI, we find that agents 3 and 9 are more influenced, and their target policies learned by TAO are more deterministic. In this manner, TAO generates a *targeted* goal for susceptible agents, as they have a higher probability of being influenced toward the collectively worst target policy. Conversely, since agent 7 is less influenced, the target policy learned by TAO is less deterministic. In this case, TAO generates an *untargeted* goal for insusceptible agents: as their policies are difficult to influence, the attacker achieves the best results by preventing them from playing the optimal policy at the current timestep. In this way, TAO automatically learns different policies for susceptible and insusceptible victims under the current attacks.

5.5. Ablations

In this section, we verify the effectiveness of each component in our model. We conduct all experiments on *3s5z vs 3s6z* for SMAC and *HalfCheetah 6x1* for MAMujoco.

5.5.1. Ablations on unilateral and targeted properties

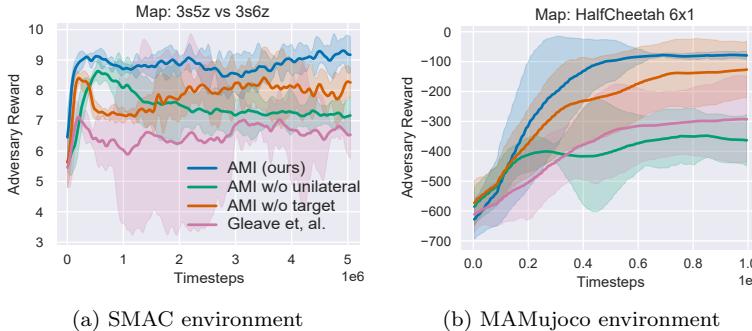


Figure 9: Ablation of unilateral and targeted properties of AMI. Both properties improve attack performance.

In this ablation study, we seek to assess the individual contributions of AMI’s unilateral and targeted attack properties. For a version of AMI without unilateral attack capabilities, we integrate the majority influence term from Eqn. 4 into Eqn. 9, resulting in bilateral influence. In contrast, for AMI without a targeted attack, we employ Eqn. A.1 as the influence metric. We also provide a comparison with the results of *Gleave et al.* (*i.e.*, absent of unilateral and targeted attacks). As illustrated in Fig. 9, the effectiveness of AMI relies on both enhancements. Notably, we observe that AMI without unilateral attack capabilities underperforms *Gleave et al.* in the *HalfCheetah 6x1* environment. This finding implies that majority influence can strongly diminish attack capability by encouraging the attacker to modify its actions to overfit to victim policies.

5.5.2. Ablations on distance metric

Next, we investigate the performance of AMI when utilizing various distance metrics. We present alternative distance metrics for both discrete and continuous environments in Table 1, where $\hat{\mu}_i^\nu(s, \mathbf{a})$ represents the mean predicted by the opponent modeling for continuous control, under the assumption that actions adhere to a Gaussian distribution. The outcomes are illustrated in Fig. 10, which reveals that the attack capability of AMI remains largely unaffected by the choice of distance metrics for both continuous and discrete control scenarios. Notably, the metrics employed in our AMI (ℓ_1 for discrete control and Prob for continuous control) yield the most favorable results.

Table 1: In this section, we address the distance metrics employed for both discrete and continuous environments. AMI is determined using these metrics and subsequently maximized by the attacker under the optimal hyperparameter λ .

| Name | Equation | Environment |
|---------------|---|-------------|
| ℓ_1 | $- p(\hat{a}_i^\nu s, \mathbf{a}) - \mathbf{1}_{\mathcal{A}}(a_i^\tau) _1$ | Discrete |
| ℓ_2 | $- p(\hat{a}_i^\nu s, \mathbf{a}) - \mathbf{1}_{\mathcal{A}}(a_i^\tau) _2$ | Discrete |
| ℓ_∞ | $- p(\hat{a}_i^\nu s, \mathbf{a}) - \mathbf{1}_{\mathcal{A}}(a_i^\tau) _\infty$ | Discrete |
| CE | $\log(p(a_i^\tau s, \mathbf{a}))$ | Discrete |
| Prob | $p(a_i^\tau s, \mathbf{a})$ | Discrete |
| ℓ_1 | $- \hat{\mu}_i^\nu(s, \mathbf{a}) - a_i^\tau _1$ | Continuous |
| CE | $\log(p(a_i^\tau s, \mathbf{a}))$ | Continuous |
| Prob | $p(a_i^\tau s, \mathbf{a})$ | Continuous |

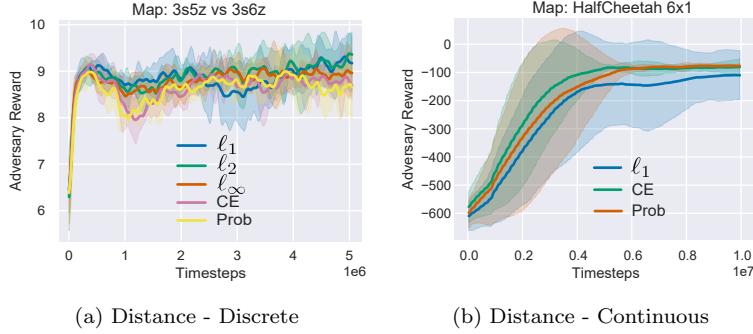


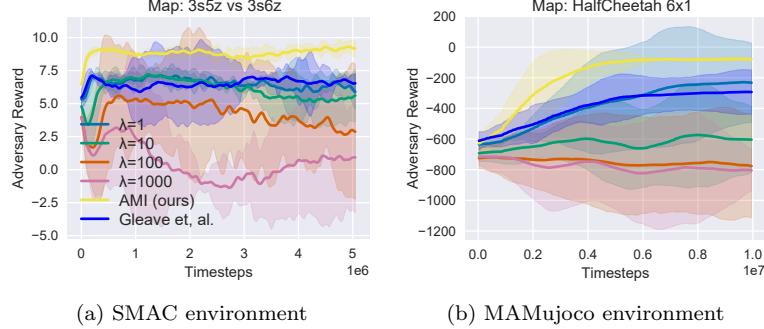
Figure 10: Ablation on distance metric. The performance of AMI is stable under different distance metrics.

5.5.3. Comparing with mutual information

Since unilateral influence is derived from mutual information, we also take mutual information [51] that depicts the bilateral influence between victims and adversary as a baseline. Specifically, we evaluate the result for four different λ values that modulate the magnitude of mutual information. As illustrated in Fig. 11, utilizing mutual information often results in inferior performance compared to *Gleave et al.* (*i.e.*, not employing mutual information). Moreover, the outcome worsens as λ increases. These observations serve as motivation for the development of AMI, a unilateral and targeted influence designed for c-MARL attacks, which yields superior performance.

5.5.4. Ablations on hyperparameter

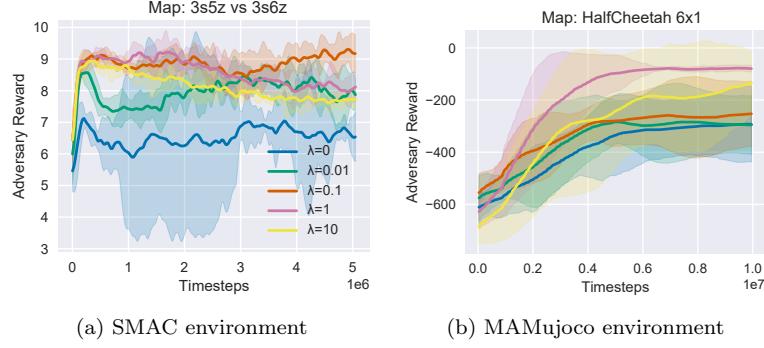
Lastly, we evaluate AMI under varying hyperparameters λ , which modulate the contribution of the AMI reward I_t^α to the total reward. Specifically,



(a) SMAC environment

(b) MAMujoco environment

Figure 11: Using mutual information proposed in [51] as influence metric. Since mutual information is bidirectional, such metric was ineffective for attack.

Figure 12: Ablation on hyperparameter λ .

we assess our AMI attack for $\lambda = \{0, 0.01, 0.1, 1, 10\}$, where $\lambda = 0$ reduces to the method of *Gleave et al.* As depicted in Fig. 12, incorporating AMI enhances the overall attack performance compared to *Gleave et al.*, thereby demonstrating the effectiveness of our approach. Moreover, selecting an appropriate hyperparameter leads to optimal AMI performance ($\lambda = 0.1$ in *3s5z vs 3s6z* and $\lambda = 1$ in *HalfCheetah 6x1*). We also observe that performance does not improve monotonically with increasing λ values. This can be attributed to the following factors: (1) a large λ generates high rewards, which may introduce instability when training the critic network, and (2) a substantial λ causes errors in the opponent modeling module p_ϕ to accumulate, resulting in excessive emphasis on the attack capability of the subsequent timestep.

6. Conclusion

In this paper, we present AMI as a strong and practical black-box attack to assess the robustness of c-MARL, in which the attacker unilaterally influences victims to establish a worst-case collaboration. Firstly, we adapt agent-wise bilateral mutual information to a unilateral adversary-victim influence for policy-based attacks by decomposing mutual information and filtering out the influence from victims to the adversary. Secondly, we employ a reinforcement learning agent to generate the jointly worst-case target for the attacker to influence in order to maximize team reward. Through AMI, we pioneers the successful execution of adversarial attacks on real-world robotic swarms, and demonstrate its effectiveness in compelling agents towards collectively unfavorable outcomes in simulated environments. Our findings not only offer valuable insights into system vulnerabilities but also open opportunities to strengthen the resilience of cooperative multi-agent systems.

Acknowledgements

This work was supported by the National Key Research and Development Plan of China (2022ZD0116405), the National Natural Science Foundation of China (62022009, 62206009, 62306025), and the State Key Laboratory of Software Development Environment.

Appendix A. Detailed Derivation of Eqn. 5

Here we present the detailed derivation of Eqn. 5 in our main paper.

$$\begin{aligned}
& H(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) \\
&= - \sum_{\hat{a}_{t+1,i}^\nu \in \mathcal{A}_i} p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) \log(p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu)) \\
&= - \sum_{\hat{a}_{t+1,i} \in \mathcal{A}_i} p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) \log(p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) - p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) \log(\mathcal{U}) + c, \\
&= -D_{KL}(p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \mathbf{a}_t^\nu) || \mathcal{U}) + c \\
&= -D_{KL}\left(\sum_{\tilde{a}_t^\alpha} p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu) \cdot p(\tilde{a}_t^\alpha | s_t, \mathbf{a}_t^\nu)\right) || \mathcal{U}) + c, \\
&= -D_{KL}\left(\sum_{\tilde{a}_t^\alpha} p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu) \cdot \pi^\alpha(\tilde{a}_t^\alpha | s_t)\right) || \mathcal{U}) + c, \\
&= -D_{KL}\left(\mathbb{E}_{\tilde{a}_t^\alpha \sim \pi^\alpha} [p_\phi(\hat{a}_{t+1,i}^\nu | s_t, \tilde{a}_t^\alpha, \mathbf{a}_t^\nu)] || \mathcal{U}\right) + c,
\end{aligned} \tag{A.1}$$

Appendix B. Convergence of Value Function $Q^\tau(s, a^\tau, a^\alpha)$

Here we present the full proof of convergence of $Q^\tau(s, a^\tau, a^\alpha)$. We first show that updating Q^τ by Bellman operator \mathcal{B}^τ is a contraction on Banach space, with \mathcal{B}^τ defined as:

$$\begin{aligned}
(\mathcal{B}^\tau Q^\tau)(s, a^\tau, a^\alpha) &= r_t^\alpha + \gamma \sum_{s'} \mathcal{T}(s' | s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \\
&\quad \pi(a'^\tau | s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha | h'_i) \pi^\nu(\mathbf{a}'^\alpha | \mathbf{h}') Q^\tau(s', a'^\tau, a'^\alpha).
\end{aligned} \tag{B.1}$$

Define two Q functions $Q_1^\tau(s, a^\tau, a^\alpha)$ and $Q_2^\tau(s, a^\tau, a^\alpha)$, we need to show the Bellman operator \mathcal{B}^τ is a contraction in sup-norm:

$$\begin{aligned}
& \|\mathcal{B}^\tau Q_1^\tau - \mathcal{B}^\tau Q_2^\tau\|_\infty \\
&= \max_{s, a^\tau, a^\alpha} |(\mathcal{B}^\tau Q_1^\tau)(s, a^\tau, a^\alpha) - (\mathcal{B}^\tau Q_2^\tau)(s, a^\tau, a^\alpha)| \\
&= \max_{s, a^\tau, a^\alpha} \left| r_t^\alpha + \gamma \mathcal{T}(s'|s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau|s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \right. \\
&\quad \pi^\alpha(a'^\alpha|h'_i) \pi^\nu(\mathbf{a}'^\alpha|\mathbf{h}') Q_1^\tau(s', a'^\tau, a'^\alpha) - r_t^\alpha - \gamma \mathcal{T}(s'|s, a^\alpha, \mathbf{a}^\nu) \\
&\quad \left. \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau|s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha|h'_i) \pi^\nu(\mathbf{a}'^\alpha|\mathbf{h}') Q_2^\tau(s', a'^\tau, a'^\alpha) \right| \\
&= \max_{s, a^\tau, a^\alpha} \gamma \left| \mathcal{T}(s'|s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau|s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \right. \\
&\quad \left. \pi^\nu(\mathbf{a}'^\alpha|\mathbf{h}') (Q_1^\tau(s', a'^\tau, a'^\alpha) - Q_2^\tau(s', a'^\tau, a'^\alpha)) \right| \\
&\leq \max_{s, a^\tau, a^\alpha} \gamma \mathcal{T}(s'|s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau|s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha|h'_i) \\
&\quad \pi^\nu(\mathbf{a}'^\alpha|\mathbf{h}') |Q_1^\tau(s', a'^\tau, a'^\alpha) - Q_2^\tau(s', a'^\tau, a'^\alpha)| \\
&\leq \gamma \|Q_1^\tau(s', a'^\tau, a'^\alpha) - Q_2^\tau(s', a'^\tau, a'^\alpha)\|_\infty
\end{aligned} \tag{B.2}$$

Thus, \mathcal{B}^τ is a contraction operator. Finally, by Banach's fixed point theorem, with finite joint action space \mathcal{A} , state space \mathcal{S} , and assume each state-action pair is visited infinitesimally often, updating $Q^\tau(s, a^\tau, a^\alpha)$ by Bellman operator \mathcal{B}^τ will converge to the optimal value function $Q^{\tau,*}(s, a^\tau, a^\alpha)$. Note that the guaranteed convergence happens in tabular case. This motivates us to use PPO algorithm as a practical solver of this problem.

Appendix C. Convergence of Value Function $Q^\alpha(s, a^\tau, a^\alpha)$

The convergence of $Q^\alpha(s, a^\alpha, a^\alpha)$ follows the same technique with the convergence proof of $Q^\tau(s, a^\tau, a^\alpha)$. We state the proof here again for completeness.

Again, we first show that updating Q^α by Bellman operator \mathcal{B}^α is a

contraction on Banach space, with \mathcal{B}^α defined as:

$$\begin{aligned}
(\mathcal{B}^\alpha Q^\alpha)(s, a^\tau, a^\alpha) = & r^\alpha + \lambda \cdot I^\alpha + \gamma \sum_{s'} \mathcal{T}(s' | s, a^\alpha, \mathbf{a}^\nu) \\
& \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau | s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha | h'_i) \pi^\nu(\mathbf{a}'^\alpha | \mathbf{h}') Q^\alpha(s', a'^\tau, a'^\alpha).
\end{aligned} \tag{C.1}$$

Define two Q functions $Q_1^\alpha(s, a^\tau, a^\alpha)$ and $Q_2^\alpha(s, a^\tau, a^\alpha)$, we need to show the Bellman operator \mathcal{B}^α is a contraction in sup-norm:

$$\begin{aligned}
& ||\mathcal{B}^\alpha Q_1^\alpha - \mathcal{B}^\alpha Q_2^\alpha||_\infty \\
= & \max_{s, a^\tau, a^\alpha} |(\mathcal{B}^\alpha Q_1^\alpha)(s, a^\tau, a^\alpha) - (\mathcal{B}^\alpha Q_2^\alpha)(s, a^\tau, a^\alpha)| \\
= & \max_{s, a^\tau, a^\alpha} \left| r_t^\alpha + \lambda \cdot I^\alpha + \gamma \mathcal{T}(s' | s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau | s') \right. \\
& \left. \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha | h'_i) \pi^\nu(\mathbf{a}'^\alpha | \mathbf{h}') Q_1^\alpha(s', a'^\tau, a'^\alpha) - r_t^\alpha \right. \\
& \left. - \lambda \cdot I^\alpha - \gamma \mathcal{T}(s' | s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau | s') \right. \\
& \left. \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha | h'_i) \pi^\nu(\mathbf{a}'^\alpha | \mathbf{h}') Q_2^\alpha(s', a'^\tau, a'^\alpha) \right| \tag{C.2} \\
= & \max_{s, a^\tau, a^\alpha} \gamma \left| \mathcal{T}(s' | s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau | s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha | h'_i) \right. \\
& \left. \pi^\nu(\mathbf{a}'^\alpha | \mathbf{h}') (Q_1^\alpha(s', a'^\tau, a'^\alpha) - Q_2^\alpha(s', a'^\tau, a'^\alpha)) \right| \\
\leq & \max_{s, a^\tau, a^\alpha} \gamma \mathcal{T}(s' | s, a^\alpha, \mathbf{a}^\nu) \sum_{a'^\tau \in \mathcal{A}} \pi(a'^\tau | s') \sum_{(\mathbf{a}'^\nu, a'^\alpha) \in \mathcal{A}} \pi^\alpha(a'^\alpha | h'_i) \\
& |\pi^\nu(\mathbf{a}'^\alpha | \mathbf{h}')| |Q_1^\alpha(s', a'^\tau, a'^\alpha) - Q_2^\alpha(s', a'^\tau, a'^\alpha)| \\
\leq & \gamma ||Q_1^\alpha(s', a'^\tau, a'^\alpha) - Q_2^\alpha(s', a'^\tau, a'^\alpha)||_\infty
\end{aligned}$$

Thus, \mathcal{B}^α is a contraction operator. Finally, by Banach's fixed point theorem, with finite joint action space \mathcal{A} , state space \mathcal{S} , and assume each state-action pair is visited infinitesimally often, updating $Q^\alpha(s, a^\tau, a^\alpha)$ by Bellman operator \mathcal{B}^α will converge to the optimal value function $Q^{\alpha,*}(s, a^\tau, a^\alpha)$.

Again, the guaranteed convergence happens in tabular case. We thus use PPO algorithm as a practical solver of this problem.

Appendix D. Experiment Hyperparameters

In this section, we describe the hyperparameters of AMI and baselines. For **SMAC environment**, Table. D.2 describes the shared parameters used by all methods in SMAC experiments. Table. D.3 denotes the hyperparameters used for each individual methods.

Table D.2: Shared hyperparameters for SMAC, used in AMI and all baselines.

| Hyperparameter | Value | Hyperparameter | Value |
|----------------|-------|-----------------|-------|
| lr | 1e-4 | mini-batch num | 1 |
| parallel envs | 32 | max grad norm | 10 |
| gamma | 0.99 | max episode len | 150 |
| actor network | mlp | actor lr | =lr |
| hidden dim | 64 | critic lr | =lr |
| hidden layer | 1 | PPO epoch | 4 |
| activation | ReLU | PPO clip | 0.2 |
| optimizer | Adam | entropy coef | 0.01 |
| GAE lambda | 0.95 | eval episode | 20 |

Table D.3: Method-specific parameters for *Wu et al.*, *Guo et al.* and our AMI in SMAC environment.

| Hyperparameters for Wu et al. | | | |
|--------------------------------|-----------|----------------|----------------|
| Hyperparameter | Value | Hyperparameter | Value |
| epsilon_state | 0.1 | epsilon_action | 0.1 |
| lr_state | =actor lr | lr_action | =actor lr |
| Hyperparameters for Guo et al. | | | |
| Hyperparameter | Value | Hyperparameter | Value |
| victim critic lr | =lr | | |
| Hyperparameters for AMI | | | |
| Hyperparameter | Value | Hyperparameter | Value |
| p_ϕ lr | =lr | TAO lr | =lr |
| TAO critic lr | =lr | AMI lambda | [.03, .05, .1] |

For **MAMujoco environment**, Table. D.4 describes the parameters used for all experiments. Table. D.5 denotes the hyperparameters used for each individual methods.

Table D.4: Shared hyperparameters for MAMujoco, used in AMI and all baselines in MAMujoco environment.

| Hyperparameter | Value | Hyperparameter | Value |
|----------------|-------|-----------------|--------------|
| parallel envs | 32 | mini-batch num | 40 |
| gamma | 0.99 | max grad norm | 10 |
| gain | 0.01 | max episode len | 1000 |
| actor network | mlp | actor lr | [5e-6, 5e-4] |
| std y coef | 0.5 | critic lr | 5e-3 |
| std x coef | 1 | Huber loss | True |
| hidden dim | 64 | Huber delta | 10 |
| hidden layer | 1 | PPO epoch | 5 |
| activation | ReLU | PPO clip | 0.2 |
| optimizer | Adam | entropy coef | 0.01 |
| GAE lambda | 0.95 | eval episode | 32 |

For **rendezvous environment**, the hyperparameters and implementations follows MAMujoco environment, with small variations to achieve best performance in this scenario. Table. D.6 describes the parameters used for the rendezvous environment. Table. D.7 denotes the hyperparameters used for each individual methods.

References

- [1] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, S. Whiteson, Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning, in: International conference on machine learning, PMLR, 2018, pp. 4295–4304.
- [2] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, Advances in neural information processing systems 30 (2017).
- [3] C. Yu, A. Velu, E. Vinitksy, Y. Wang, A. Bayen, Y. Wu, The surprising effectiveness of ppo in cooperative, multi-agent games, arXiv preprint arXiv:2103.01955 (2021).

Table D.5: Method-specific parameters for *Wu et al.*, *Guo et al.* and our AMI in MAMU-joco environment.

| Hyperparameters for Wu et al. | | | |
|---------------------------------------|-----------|----------------|------------------|
| Hyperparameter | Value | Hyperparameter | Value |
| epsilon_state | 0.1 | epsilon_action | 0.1 |
| lr_state | =actor lr | lr_action | =actor lr |
| Hyperparameters for <i>Guo et al.</i> | | | |
| Hyperparameter | Value | Hyperparameter | Value |
| victim critic lr | =lr | | |
| iteration | 30 | | |
| Hyperparameters for AMI | | | |
| Hyperparameter | Value | Hyperparameter | Value |
| p_ϕ lr | =lr | TAO lr | =lr |
| TAO critic lr | =lr | AMI lambda | [.01, .1, .3, 1] |

Table D.6: Shared hyperparameters for rendezvous, used in AMI and all baselines.

| Hyperparameter | Value | Hyperparameter | Value |
|----------------|-------|-----------------|-------|
| parallel envs | 32 | mini-batch num | 40 |
| gamma | 0.99 | max grad norm | 10 |
| gain | 0.01 | max episode len | 1000 |
| actor network | mlp | actor lr | 5e-5 |
| std y coef | 0.5 | critic lr | 5e-3 |
| std x coef | 1 | Huber loss | True |
| hidden dim | 64 | Huber delta | 10 |
| hidden layer | 1 | PPO epoch | 5 |
| activation | ReLU | PPO clip | 0.2 |
| optimizer | Adam | entropy coef | 0.01 |
| GAE lambda | 0.95 | eval episode | 32 |

Table D.7: Method-specific parameters for *Wu et al.*, *Guo et al.* and our AMI in rendezvous environment.

| Hyperparameters for Wu et al. | | | |
|---------------------------------------|-----------|----------------|-----------|
| Hyperparameter | Value | Hyperparameter | Value |
| epsilon_state | 0.1 | epsilon_action | 0.1 |
| lr_state | =actor lr | lr_action | =actor lr |
| Hyperparameters for <i>Guo et al.</i> | | | |
| Hyperparameter | Value | Hyperparameter | Value |
| victim critic lr | =lr | | |
| Hyperparameters for AMI | | | |
| Hyperparameter | Value | Hyperparameter | Value |
| p_ϕ lr | =lr | TAO lr | =lr |
| TAO critic lr | =lr | AMI lambda | .003 |

- [4] J. Ji, T. Qiu, B. Chen, B. Zhang, H. Lou, K. Wang, Y. Duan, Z. He, J. Zhou, Z. Zhang, et al., Ai alignment: A comprehensive survey, arXiv preprint arXiv:2310.19852 (2023).
- [5] J. Ji, B. Chen, H. Lou, D. Hong, B. Zhang, X. Pan, J. Dai, Y. Yang, Aligner: Achieving efficient alignment through weak-to-strong correction, arXiv preprint arXiv:2402.02416 (2024).
- [6] J. Ji, M. Liu, J. Dai, X. Pan, C. Zhang, C. Bian, B. Chen, R. Sun, Y. Wang, Y. Yang, Beavertails: Towards improved safety alignment of llm via a human-preference dataset, Advances in Neural Information Processing Systems 36 (2024).
- [7] J. Ji, K. Wang, T. Qiu, B. Chen, J. Zhou, C. Li, H. Lou, Y. Yang, Language models resist alignment, arXiv preprint arXiv:2406.06144 (2024).
- [8] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, S. Whiteson, The starcraft multi-agent challenge, arXiv preprint arXiv:1902.04043 (2019).
- [9] T. Chu, J. Wang, L. Codecà, Z. Li, Multi-agent deep reinforcement learning for large-scale traffic signal control, IEEE transactions on intelligent transportation systems 21 (3) (2019) 1086–1095.

- [10] C. Zhang, V. Lesser, P. Shenoy, A multi-agent learning approach to online distributed resource allocation, in: Twenty-first international joint conference on artificial intelligence, 2009.
- [11] B. Peng, T. Rashid, C. Schroeder de Witt, P.-A. Kamienny, P. Torr, W. Böhmer, S. Whiteson, Facmac: Factored multi-agent centralised policy gradients, *Advances in Neural Information Processing Systems* 34 (2021) 12208–12221.
- [12] M. Hüttenrauch, S. Adrian, G. Neumann, et al., Deep reinforcement learning for swarm systems, *Journal of Machine Learning Research* 20 (54) (2019) 1–31.
- [13] W. J. Yun, S. Park, J. Kim, M. Shin, S. Jung, D. A. Mohaisen, J.-H. Kim, Cooperative multiagent deep reinforcement learning for reliable surveillance via autonomous multi-uav control, *IEEE Transactions on Industrial Informatics* 18 (10) (2022) 7086–7096.
- [14] X. Yu, R. Shi, P. Feng, Y. Tian, J. Luo, W. Wu, Esp: Exploiting symmetry prior for multi-agent reinforcement learning, in: ECAI 2023, IOS Press, 2023, pp. 2946–2953.
- [15] X. Yu, W. Wu, P. Feng, Y. Tian, Swarm inverse reinforcement learning for biological systems, in: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2021, pp. 274–279.
- [16] J. Lin, K. Dzeparoska, S. Q. Zhang, A. Leon-Garcia, N. Papernot, On the robustness of cooperative multi-agent reinforcement learning, in: 2020 IEEE Security and Privacy Workshops (SPW), IEEE, 2020, pp. 62–68.
- [17] L. Zan, X. Zhu, Z.-L. Hu, Adversarial attacks on cooperative multi-agent deep reinforcement learning: a dynamic group-based adversarial example transferability method, *Complex & Intelligent Systems* (2023) 1–12.
- [18] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, P. Abbeel, Adversarial attacks on neural network policies, *arXiv preprint arXiv:1702.02284* (2017).

- [19] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, C.-J. Hsieh, Robust deep reinforcement learning against adversarial perturbations on state observations, *Advances in Neural Information Processing Systems* 33 (2020) 21024–21037.
- [20] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, S. Russell, Adversarial policies: Attacking deep reinforcement learning, *arXiv preprint arXiv:1905.10615* (2019).
- [21] X. Wu, W. Guo, H. Wei, X. Xing, Adversarial policy training against deep reinforcement learning, in: *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1883–1900.
- [22] W. Guo, X. Wu, S. Huang, X. Xing, Adversarial policy learning in two-player competitive games, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 3910–3919.
- [23] J. Wu, X. Xu, Decentralised grid scheduling approach based on multi-agent reinforcement learning and gossip mechanism, *CAAI Transactions on Intelligence Technology* 3 (1) (2018) 8–17.
- [24] S. Shalev-Shwartz, S. Shammah, A. Shashua, Safe, multi-agent, reinforcement learning for autonomous driving, *arXiv preprint arXiv:1610.03295* (2016).
- [25] S. Bhalla, S. Ganapathi Subramanian, M. Crowley, Deep multi agent reinforcement learning for autonomous driving, in: *Canadian Conference on Artificial Intelligence*, Springer, 2020, pp. 67–78.
- [26] W. D. Crano, V. Seyranian, Majority and minority influence, *Social and Personality Psychology Compass* 1 (1) (2007) 572–589.
- [27] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, *arXiv preprint arXiv:1312.6199* (2013).
- [28] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, *arXiv preprint arXiv:1412.6572* (2014).
- [29] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: *2017 ieee symposium on security and privacy (sp)*, IEEE, 2017, pp. 39–57.

- [30] J. Kos, D. Song, Delving into adversarial attacks on deep policies, arXiv preprint arXiv:1705.06452 (2017).
- [31] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, M. Sun, Tactics of adversarial attack on deep reinforcement learning agents, arXiv preprint arXiv:1703.06748 (2017).
- [32] Y. Huang, Q. Zhu, Deceptive reinforcement learning under adversarial manipulations on cost signals, in: International Conference on Decision and Game Theory for Security, Springer, 2019, pp. 217–237.
- [33] F. Wu, L. Li, C. Xu, H. Zhang, B. Kailkhura, K. Kenthapadi, D. Zhao, B. Li, Copa: Certifying robust policies for offline reinforcement learning against poisoning attacks, arXiv preprint arXiv:2203.08398 (2022).
- [34] V. Behzadan, W. Hsu, Sequential triggers for watermarking of deep reinforcement learning policies, arXiv preprint arXiv:1906.01126 (2019).
- [35] P. Kiourtzi, K. Wardega, S. Jha, W. Li, Trojdrl: Trojan attacks on deep reinforcement learning agents, arXiv preprint arXiv:1903.06638 (2019).
- [36] L. Wang, Z. Javed, X. Wu, W. Guo, X. Xing, D. Song, Backdoorl: Backdoor attack against competitive reinforcement learning, arXiv preprint arXiv:2105.00579 (2021).
- [37] H. Zhang, H. Chen, D. Boning, C.-J. Hsieh, Robust reinforcement learning on state observations with learned optimal adversary, arXiv preprint arXiv:2101.08452 (2021).
- [38] Y. Sun, R. Zheng, Y. Liang, F. Huang, Who is the strongest enemy? towards optimal and efficient evasion attacks in deep rl, arXiv preprint arXiv:2106.05087 (2021).
- [39] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE, 2016, pp. 372–387.
- [40] J. Tu, T. Wang, J. Wang, S. Manivasagam, M. Ren, R. Urtasun, Adversarial attacks on multi-agent communication, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7768–7777.

- [41] W. Xue, W. Qiu, B. An, Z. Rabinovich, S. Obraztsova, C. K. Yeo, Mis-speak or mis-lead: Achieving robustness in multi-agent communicative reinforcement learning, arXiv preprint arXiv:2108.03803 (2021).
- [42] J. Blumenkamp, A. Prorok, The emergence of adversarial communication in multi-agent reinforcement learning, in: Conference on Robot Learning, PMLR, 2021, pp. 1394–1414.
- [43] R. Mitchell, J. Blumenkamp, A. Prorok, Gaussian process based message filtering for robust multi-agent cooperation in the presence of adversarial communication, arXiv preprint arXiv:2012.00508 (2020).
- [44] M. Figura, K. C. Kosaraju, V. Gupta, Adversarial attacks in consensus-based multi-agent reinforcement learning, in: 2021 American Control Conference (ACC), IEEE, 2021, pp. 3050–3055.
- [45] E. A. Hansen, D. S. Bernstein, S. Zilberstein, Dynamic programming for partially observable stochastic games, in: AAAI, Vol. 4, 2004, pp. 709–715.
- [46] F. A. Oliehoek, C. Amato, A concise introduction to decentralized POMDPs, Springer, 2016.
- [47] S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard, et al., Sim2real in robotics and automation: Applications and challenges, IEEE transactions on automation science and engineering 18 (2) (2021) 398–400.
- [48] S. M. Giray, Anatomy of unmanned aerial vehicle hijacking with signal spoofing, in: 2013 6th International Conference on Recent Advances in Space Technologies (RAST), IEEE, 2013, pp. 795–800.
- [49] B. Ly, R. Ly, Cybersecurity in unmanned aerial vehicles (uavs), Journal of Cyber Security Technology 5 (2) (2021) 120–137.
- [50] N. M. Rodday, R. d. O. Schmidt, A. Pras, Exploring security vulnerabilities of unmanned aerial vehicles, in: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2016, pp. 993–994.

- [51] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, N. De Freitas, Social influence as intrinsic motivation for multi-agent deep reinforcement learning, in: International conference on machine learning, PMLR, 2019, pp. 3040–3049.
- [52] T. Wang, J. Wang, Y. Wu, C. Zhang, Influence-based multi-agent exploration, arXiv preprint arXiv:1910.05512 (2019).
- [53] A. Fayad, M. Ibrahim, Influence-based reinforcement learning for intrinsically-motivated agents, arXiv preprint arXiv:2108.12581 (2021).
- [54] P. Li, H. Tang, T. Yang, X. Hao, T. Sang, Y. Zheng, J. Hao, M. E. Taylor, Z. Wang, Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration, arXiv preprint arXiv:2203.08553 (2022).
- [55] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).
- [56] J. Schulman, P. Moritz, S. Levine, M. Jordan, P. Abbeel, High-dimensional continuous control using generalized advantage estimation, arXiv preprint arXiv:1506.02438 (2015).
- [57] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, A. Martinoli, The e-puck, a robot designed for education in engineering, in: Proceedings of the 9th conference on autonomous robot systems and competitions, Vol. 1, IPCB: Instituto Politécnico de Castelo Branco, 2009, pp. 59–65.