Adversarial Policies and Defense in Cooperative Multi-Agent Reinforcement Learning

Abstract

Cooperative Multi-Agent Reinforcement Learning (c-MARL) enables a team of agents to collaboratively determine the global optimal policy that maximizes the sum of their local accumulated rewards. In this paper, we investigate the robustness of c-MARL to adversaries capable of targeting and weaponizing one agent, termed *compromised agent*, to create natural observations that are adversarial for its team. The goal of this attack is to lure the compromised agent to follow an adversarial policy that pushes activations of its cooperating agents' policy networks off-distribution. This paper proves the feasibility of such attack using 3 attack strategies in white-box and black-box settings against the state-of-the-art c-MARL algorithm MADDPG as a case study. By compromising a single agent in 2 different MADDPG particle environments, our attacks have a highly negative impact on the overall team reward. In the physical deception environment, it reduces the team reward by 86.6% and 85% in the white-box and black-box settings, respectively. In the cooperative navigation environment, it achieved a reward drop of 42.8% and 37.5% for white-box and black-box settings, respectively.

1 INTRODUCTION

Advances in single-agent reinforcement learning RL algorithms sparked new interest in cooperative Multi-Agent Reinforcement Learning (c-MARL). Several c-MARL training algorithms have been developed in response to demand in such areas as cyber-physical systems, sensor/communication networks, and social science. Massive effort now focuses on either identifying new learning criteria or setups [Foerster et al., 2016, Zazo et al., 2016, Zhang et al., 2018, Subramanian and Mahajan, 2019], or developing new algorithms for existing setups, using deep learning [Heinrich and Silver, 2016, Lowe et al., 2020, Foerster et al., 2017, Gupta et al., 2017, Omidshafiei et al., 2017, Zhang et al., 2019b], operations research [Mazumdar et al., 2020, Jin et al., 2019, Sidford et al., 2019], and multi-agent systems [Arslan and Yüksel, 2016, Yongacoglu et al., 2019, Zhang et al., 2019a]. Despite the emergent, broadly applicable algorithms, RL agents are still vulnerable to adversaries perturbing their observations with adversarial examples [Huang et al., 2017, Kos and Song, 2017], as well as adversaries directly controlling the actions of one of the victim's opponents [Gleave et al., 2020]. In c-MARL, agents work in environments populated by other agents, including humans, who can only modify another agent's observations via their own actions.

Adversarial attacks on c-MARL are different from those on an individual RL agents in several ways. First, each agent in c-MARL interacts with the environment through a sequence of actions where each action involves modifying the state of the environment not only for itself but for its cooperative agents as well. For an episode of L steps, an adversary has 2^L choices to attack at least one agent at each time step which implicitly affects other agents' observations of the environment states. Hence, the attack surface for c-MARL is significantly amplified over the attack surface of the individual RL agent. Second, an adversary to c-MARL systems has different goals, such as reducing the final rewards of the whole team. The adversary can maliciously use some agents' actions to create naturally adversarial observations, luring other agents to dangerous states. We call this attack method; *compromised agent attack* which is different from adversarial attacks against an individual RL agent that aim to directly lure that agent to non-preferred states. The compromised agent attack against c-MARL has been only investigated in one recent work [Lin et al., 2020] which has produced one white-box attack against a c-MARL algorithm (QMIX) using a modified version of the adversarial example attack, JSMA, in one multi-agent environment.

Submitted to the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021). **To be used for reviewing only**.

c-MARL has been applied in settings as varied as autonomous driving in Platoons [Peake et al., 2020], negotiation [Tang, 2020], and automated scalping trading [Jo et al., 2019]. The compromised agent threat is very relevant to such domains since it is not usually possible for the adversary to directly change the victim policy's input via adversarial examples in those domains. For example, in platoons of autonomous vehicles, pedestrians and other drivers can take actions in the world that affect the camera image, but only in a physical fashion. They cannot add noise to image pixels, or make a building disappear [Gleave et al., 2020]. Similarly, in financial trading an adversary can send orders to an exchange which will appear in the victim's market data feed, but the attacker cannot modify observations of a third party's orders.

In this paper, we show the impact an adversary could have on the long-term reward of a team of agents by weaponizing one compromised agent to follow an adversarial policy that pushes activations of cooperating agents' policy networks off-distribution. In the white-box setting, the adversary has access to the reward function, state transition function, and the policies of all agents. In the black-box setting, the adversary has no knowledge about any agent. Hence, we use a deep learning-based behavioral cloning to recover the agents' state transition function and the compromised agent's policy. Our contributions are listed as follows:

1. We design 3 attack strategies to craft physically realistic adversarial policies for the compromised agent that create natural observations that are adversarial for its cooperative agents in a white-box setting.

2. We conduct the same attacks in a black-box setting in which the adversary has no knowledge about the configurations of the c-MARL agents including the compromised one.

3. We measure the success rate of each attack using the team average reward and other environment-specific measures.

4. We demonstrate the feasibility and effectiveness of our compromised agent attacks on the leading c-MARL algorithm MADDPG [Lowe et al., 2020] in two environments from the Multi-Agent Particle Environments benchmark. Our adversary brings down the team reward rate in both environments by 86.6% and 42.8%, respectively, by compromising a single agent's policy.

5. We develop a set of anomalous behavior detection models using deep learning as a potential method to mitigate our attacks.

2 BACKGROUND

2.1 C-MARL

In this paper, we model the c-MARL system using stochastic games [Shapley, 1953]. For an n-agent stochastic game, we define a tuple $G = \left(S, A^1, ..., A^n, r^1, ..., r^n, T, \gamma\right)$, where S denotes the state space, A^i and r^i are the action space and the reward function for agent $i \in 1, ..., n$ respectively. γ is the discount factor future rewards, and T is a joint state transition function $T : S \times A_1 \times A_2 .. \times A_n \rightarrow \triangle(S)$ where $\triangle(S)$ is a probability distribution on S. Agent i chooses its action $a^i \in A^i$ according to its policy $\pi^i_{\theta^i}(a^i|s)$ parameterized by θ^i conditioning on some given state $s \in S$. The collection of all agents' policies π_θ is called the joint policy and θ represents the joint parameter. For convenience, we interpret the joint policy from the perspective of agent i as:

$$\pi_\theta = (\pi^i_{\theta^i}(a^i|s)\pi^{-i}_{\theta^{-i}}(a^{-i}|s)) \tag{1}$$

where $a^{-i} = (a^j)_{j\neq i}, \theta^{-i} = (\theta^j)_{j\neq i}$, and $\pi^{-i}_{\theta^{-i}}(a^{-i}|s)$ is a compact representation of the joint policy of all complementary agents of i [Liu et al., 2020]. At each stage of the game, actions are taken simultaneously. Each agent is assumed to pursue the maximal cumulative reward [Sutton and Barto, 2018], expressed as

$$\max \eta^i(\pi_{\theta^i}) = E\left[\sum_{t=1}^{\infty} \gamma^t r^i(s_t, a^i_t, a^{-i}_t)\right], \tag{2}$$

with (a^i_t, a^{-i}_t) sample from $(\pi^i_{\theta^i}, \pi^{-i}_{\theta^{-i}})$. Correspondingly, for a game with an infinite time horizon, the state-action Q-function can be defined by $Q^i_{\pi^i_\theta}(s_t, a^i_t, a^{-i}_t) = E\left[\sum_{l=0}^{\infty} \gamma^l r^i(s_{t+1}, a^i_{t+1}, a^{-i}_{t+1})\right]$.

3 RELATED WORK

Previous work has shown the vulnerability of individual RL agents to adversarial attacks, in which the adversary perturbs the agent's observation to degrade its performance [Huang et al., 2017]. Other attacks reduce the number of adversarial examples needed to decrease the agent's reward [Lin et al., 2019] or trigger misbehavior of the agent [Zhao et al., 2019]. None of this work studies c-MARL, and the effects of cooperation on the success of the attacks. The closest work to ours is [Lin et al., 2020] which proposes a two-step attack for a c-MARL system with the objective of reducing the total team reward by perturbing the observation of a single agent. This work extends an existing adversarial example method JSMA to create d-JSMA which is more suitable for attacking an RL model with a low-dimensional feature space. They focus on white-box settings to launch their attack by assuming the knowledge of the team reward function. Our work differs from this approach by using a physically realistic attack model that does not depend on

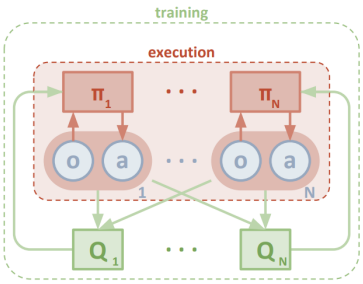

Figure 1: MADDPG architecture [Lowe et al., 2020]

directly modifying the agents' observations with adversarial examples. We also conduct our attacks in both white and black box settings. We furthermore implement a defense mechanism using an anomalous behavior detection based on ensemble deep learning model. Another work that is close to our work is Gleave et al. [Gleave et al., 2020]. However, they focus on attacks in the competitive multi-agent setting whereas we consider cooperative teams of agents in a both a fully-cooperative and a competing environment.

4 APPROACH

In this paper, we show the feasibility of the compromised agent threat using three attack strategies on c-MARL systems: counterfactual reasoning based, randomly-timed, and strategically-timed attacks. We focus on attacking the centralized-training decentralized-execution c-MARL paradigm using 2 multi-agent particle environments from the state-of-the-art algorithm MADDPG [Lowe et al., 2020].

4.1 C-MARL: MADDPG AS A CASE STUDY

The centralized learning and decentralized execution paradigm of MARL has been followed by the major c-MARL algorithms including MADDPG [Lowe et al., 2020], COMA [Foerster et al., 2017], MF-AC [Yang et al., 2018], Multi-Agent Soft-Q [Wei et al., 2018], LOLA [Foerster et al., 2018], and Q-Mix [Rashid et al., 2018].The focus of this paper is on attacking MADDPG [Lowe et al., 2020], the lead algorithm in this paradigm.

MADDPG extends DDPG into a multi-agent policy gradient algorithm in which decentralized agents learn based on the observations and actions of all agents using a centralized unit called the critic. Then, each agent has a local policy (called an actor) that only uses local information (i.e. its own observations) at execution time. The critic is augmented with extra information about the policies of other agents, while the actor only has access to its local information. After training is completed, only the local actors are used at execution phase, acting in a decentralized manner Fig.1

4.2 ADVERSARIAL CAPABILITIES

We assume that the adversary is able to take over one benign agent (i.e the compromised agent m) and use it to create naturally adversarial observations via its actions to attack other agents $-m$ in the system. In the white-box setting, the adversary may have access to the environment's reward function, the policies of every agent in the system, and the ground truth state transition function, $p(s_{t+1}|s_t, a_0, \ldots, a_n)$, where s_t is the concatenation of each agents' observations, $(o_1, \ldots, o_n)$. Conversely, a black box adversary can only access the observations of each agent in the system, and their taken actions. Additionally, the adversary accesses the actions generated by the compromised agent's policy, but not the policy values themselves and chooses to either override the chosen action, or not to act.

4.3 ATTACK STRATEGIES

We model the target MARL system as a Markov game and denote the compromised agent and victim agents by subscript m and $-m$ respectively. We assume all agents, including the compromised one, follow fixed policies corresponding to the common case of a pre-trained model, deployed with static weights. This model holds particularly well for safety-critical systems, where it is a standard practice to validate a model, then freeze it, to ensure that it does not develop any new problems due to further training [Gleave et al., 2020].

4.3.1 Counterfactual Reasoning Based Attack

This attack strategy predicts the compromised agent's counterfactual reasoning process about how its actions will affect the other agents and then postulates actions that would enable it to achieve maximal destruction to the system. The joint policy in Eq.1 for all agents M in the system can be reformulated as [Wen et al., 2019, Xia et al., 2017]:

$$\begin{aligned}\pi_\theta(a^m, a^{-m}|s) &= \underbrace{\pi^m_{\theta^m}(a^m|s)\pi^{-m}_{\theta^{-m}}(a^{-m}|s,a^m)}_{\text{Compromised agent's perspective}} \\ &= \underbrace{\pi^{-m}_{\theta^{-m}}(a^{-m}|s)\pi^m_{\theta^m}(a^m|s,a^{-m})}_{\text{other agents' perspective}}\end{aligned} \tag{3}$$

From the perspective of the compromised agent m, the first equality in Eq.3 indicates that the joint policy can be essentially decomposed into two parts. The conditional part of the first equality $\pi^{-m}_{\theta^{-m}}(a^{-m}|s,a^m)$ represents what actions would be taken by victim agents $-m$ given the fact that they know the current state of the environment and agent m's action based on their original policy $\pi^{-m}_{\theta^{-m}}(a^{-m}|s,a^m)$. Reasoning about different potential actions that agent m thinks victim agents $-m$ would take, the adversary uses agent m's marginal policy $\pi^m_{\theta^m}(a^m|s)$ to find the most destructive actions. Recall that the policies for agents $-m$ are fixed so

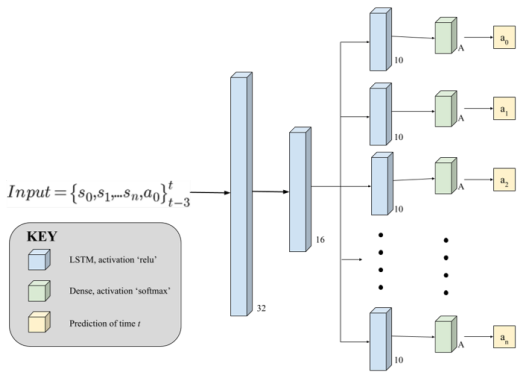

Figure 2: The architecture of the deep learning based behavioral cloning.

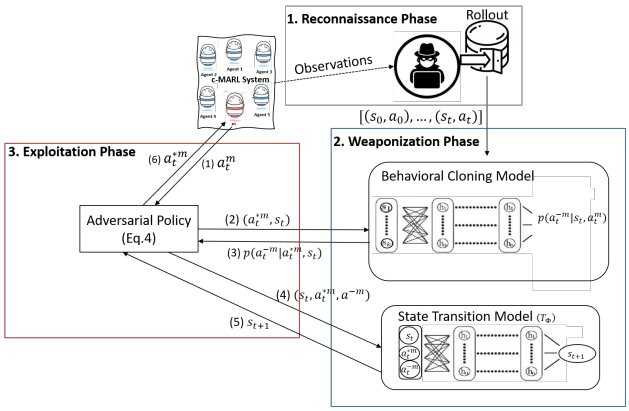

Figure 3: Overall counterfactual reasoning based attack mechanism.

they cannot apply the same inference logic about the compromised agent m during execution. To launch this attack in the white-box setting, the adversary has direct access to the agents' policies and the ground truth of the state transition function.

In the black-box setting, Eq.3 requires the adversary to have full access to the original policy of the compromised agent m, the actual conditional policy of the victims $\pi_{\theta^{-m}}^{-m}(a^{-m}|s,a^m)$, and the state transition function. To satisfy those requirements, we use a deep learning based behavioral cloning model to approximate each agent's policy, and another supervised model for predicting the state transition function, as in model-based reinforcement learning.

The deep learning based behavioral cloning is trained to approximate the policies of the compromised agent m and other agents $-m$ using the collected state-action $(s_t, at)_i \ \forall i \in M$ pairs from observing the c-MARL agents during the reconnaissance phase. This model is a multi-headed model (Fig.2) which reads input from the three previous timesteps and the current timestep for each agent's state, as well as the compromised agent's action. When the input data is processed by the first 2 LSTM layers, the model branches off into separate heads such that each head is responsible for predicting the corresponding agents' actions. Each head then adds an additional LSTM layer and ends with a Dense layer with a softmax activation function that predicts the probability confidence of all of the possible actions performed by that specific agent $a \in A$. We train this model by minimizing the loss function $L(a, \pi_\theta)$ for each agent's policy. Using this model, we train an adversarial RL policy to intercept agent m's original actions and replace them with intentionally destructive actions based on the current state and the expected response from m. We train the adversarial RL policy with a learnable parameter Ω as

$$att_\Omega = \arg\max \sum_{\forall a^{*m} \in A} \mathbf{KL}\left(p(a_t^{-m}|a_t^m, s_t)||p(a_t^{-m}|a_t^{*m}, s_t)\right) \quad (4)$$

where a^{*m} represents the set of counterfactual actions to the action a^m generated by the original policy of agent m; $\pi_{\theta^m}^m$. This attack model maximizes the KL divergence between the conditional policy of $-m$ on the action a^m at time t and the same conditional policy if agent m takes a counterfactual action a^{*m} at the same time. The adversarial RL policy can then intervene on a_t^m by replacing it with the best counterfactual action, a_t^{*m} which can be used to compute a new distribution over $-m$ next action, $p(a_t^{-m}|a_t^{*m}, s_t^m)$. Essentially, the attack model in Eq.4 asks a retrospective question: How would the predicted actions of agents $-m$, a_t^{-m} at t change to negatively impact other agents' behavior, if agent m had acted differently at t? The supervised learning model for the state transition function is similar to the behavioral cloning model with multiple heads but with a different learnable parameter:

$$T_\Phi = p(s_{t+1}|s_t, a_1, ..., a_n, a*_m) \qquad \forall *m \in A \quad (5)$$

The overall architecture of the counterfactual reasoning attack is depicted in Fig.3.

4.3.2 Randomly-Timed Attack

In this strategy, we test the performance of the c-MARL policies for the victims outside their training distribution (i.e. distribution shift). We evaluate victims' policies by developing a set of randomized off-distribution adversarial policies for the compromised agent. At a certain percentage of timesteps, the adversary randomly changes the compromised agent's action based on the off-distribution policy. In the white box setting, the adversary uses directly the ground-truth policy for the compromised agent. In the black-box variant, the adversary crafts a replica policy using the previously explained behavioral cloning model (Fig.2) to approximate the compromised agent's policy. This attack extends the random attack against individual RL agents in [Gleave et al., 2020].

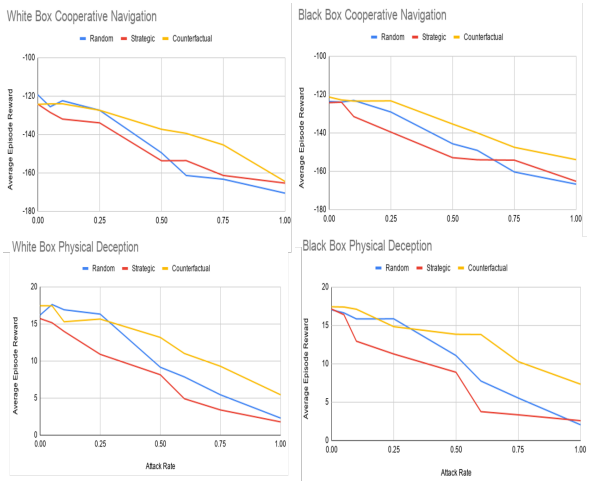

Figure 4: The average episode rewards in 2 particle multi-agent environments under our attacks as a function of the attacking rate

4.3.3 Strategically-Timed Attack

This attack is a policy based attack which is an extension to the individual RL agent attack in [Lin et al., 2019] into a c-MARL model in both the white and black box settings. We first calculate the c-function [Lin et al., 2019] as:

$$c(o_t) = \max_{a_t}(\pi_m(o_t, a_t)) - \min_{a_t}(\pi_m(o_t, a_t))$$

and launch the attack if and only if $c > b$, where b is a chosen threshold that indicates the desired attacking rate. The idea behind this attack is the adversary chooses to alter the compromised agent's action only when the agent strongly prefers a specific action (the action has a relative high probability), which means that it is critical to perform that action or the accumulated reward will be reduced. We test this attack with different b to correspond to different attacking rates as shown in the experiment section, for useful comparison with other attacks. The white-box variant of this attack uses directly the ground-truth policy returned by the MADDPG policy network for the compromised agent. The black-box variant, on the other hand, uses the behavioral cloning as shown in Fig.2 to craft a replica policy for the compromised agent as in the randomly-based attack.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We evaluate our attacks in both the white-box and black-box settings in 2 environments of MADDPG Particle environments: cooperative navigation and physical deception with variant attacking rates. In the cooperative navigation, N cooperative agents must cover L landmarks, and the agents must learn to reach separate landmarks, without communicating their observations to each other. In our experiments, we use $N = L = 3$. In the physical deception environment, N cooperative agents try to fool one adversarial agent. There are L total landmarks, with one being the 'target' landmark and only the cooperative agents know which landmark is the target one. The adversary must try to infer and reach the target landmark from the cooperative agents' positioning, and the cooperative agents must try to deceive the adversary by spacing out. The cooperative agents are rewarded as long a single member of their team reaches the target landmark. We use $N = L = 2$.

The adversary wishes to choose a set of actions for the compromised agent, $A' = [a_{1'}, \ldots, a_{T'}]$, where T is the episode length, such that: $R_{coop}([s_1, \ldots, s_N], A') <= R_{coop}([s_1, ..., s_N], A)$ where R_{coop} represents the reward for the cooperative team. Team reward is a direct measurement of the c-MARL quality; a successful attack should be able to decrease it quickly by injecting natural perturbations to the agents' observations using the compromised agent's adversarial actions. Beside the team reward, we use different environment-specific metrics to measure the attack success rate.

5.2 RESULTS AND DISCUSSION

5.2.1 Adversarial Policies and Cooperative Reward

To learn an adversarial policy for the compromised agent, we used the methods explained previously. We trained an adversarial policy for the compromised agent in each attack strategy to select suboptimal actions that minimize the total reward for its cooperative team. To evaluate the performance of each policy, we directly change the actions of the compromised agent based on the output of the adversarial policy. We ran each attack method with different attacking rates starting from 0% to 100% of the episode length and presented the team average reward in Fig.4. From the results, as the attack rate increases, the team reward decreases, indicating that each attack was successful at harming the system. Strategically-timed has the highest negative impact on team reward with less attacking rates in the physical deception environment with 86.6% and 85% reward drop for 100% attacking rate in white-box and black-box settings respectively while achieved a reward drop in the cooperative navigation of 42.8% and 37.5% for white-box and black-box settings respectively. This attack is the strongest because it attacks only at impactful timesteps as opposed to the randomly-timed attack which attacks at random time steps during the episode. The counterfactual reasoning-based attack achieved the least impact on the team reward for both environments indicating that using the policy (both strategically and randomly timed attacks are policy-based) to evaluate the impact of each action is better than using the

Table 1: The average number of occupied landmarks in the cooperative navigation and the average distance between the cooperative agents and the target landmark in physical deception [Lowe et al., 2020] for 25%, 50%, 75%, and 100% attack rates across all attack methods in both white and black box settings.

Cooperative Navigation $occupied_{landmarks}(A')$						
	White Box			**Black Box**		
Attack Rate	Random	Timed	Counterfactual	Random	Timed	Counterfactual
0%	1.611	1.611	1.611	1.611	1.611	1.611
25%	1.311	1.308	1.375	1.325	1.333	1.382
50%	0.958	0.955	1.226	1.058	1.048	1.251
75%	0.695	0.711	0.990	0.860	0.815	1.0899
100%	0.502	0.562	0.796	0.502	0.688	0.965
Physical Deception $coop_{dist}(A')$						
	White Box			**Black Box**		
Attack Rate	Random	Timed	Counterfactual	Random	Timed	Counterfactual
0%	0.163	0.163	0.163	0.163	0.163	0.163
25%	0.225	0.418	0.174	0.200	0.343	0.190
50%	0.326	0.470	0.261	0.324	0.499	0.230
75%	0.515	0.607	0.348	0.530	0.614	0.314
100%	0.540	0.688	0.481	0.654	0.680	0.416

Table 2: The results of the human-planned attack in the physical deception environment

Metrics	T=25	T=50
Adversary distance w\o attack	0.44	0.55
Adversary distance with attack	0.33	0.5
Cooperative reward w\o attack	3.4	12.1
Cooperative reward with attack	2.5	9 5.9

KL divergence.

Although the results in Fig.4 and Table 1 show that the counterfactual reasoning-based attack using the behavioral cloning in c-MARL environments has succeeded, we want to note that we observe an effect similar in nature to the canonical imitation learning problem. As we start attacking the system using the compromised agent's counterfactual actions, the impact of this attack was not as strong as other attacks. We hypothesize that this behaviour is because the inability of the behavioral cloning model to accurately capture other agents' states that were not part of their optimal policies during the reconnaissance phase. The canonical problem of imitation learning in general accumulates the learning errors, resulting in learner encountering unknown states [Bagnell, 2015]. It is also true for attacks based on the approximated models in a fully Black-box setup as has been shown in [Zhao et al., 2019]. Moreover, this attack builds its adversarial policy based on predicting each agent's action for a sequence of time steps then finds the counterfactual actions with maximum KL divergence using those predictions. Hence, the prediction errors accumulate over the time steps and hinder the attack. However, the counterfactual attack has the least detectable rate among all other attacks as we will show in the defense section which makes it more powerful in the presence of anomalous detection.

Considering the competing nature of the physical deception environment, we have also conducted a human-planned attack strategy designed particularly for this environment. In physical deception environment, the cooperative agents learn to reach a target landmark while distracting the adversarial agent by splitting up. The attacker's goal here is to use their human's knowledge about the environment to modify the actions of the compromised agent to implicitly leak the information about the target landmark to the adversarial agent so it could get closer to the target landmark. The results of this attack in two episode lengths 25 and 50 timesteps are shown in Table 2. The results show that the adversarial distance from the target landmark and the cooperative reward at the end of the episode with and without the attack. It's obvious that when we are attacking, the adversarial agent's distance to the target landmark decreases which means the adversarial agent understood the clue given to it by the compromised agent's actions. However, an interesting finding is that when we attack during episodes of length 50, the distance didn't decrease significantly as expected. Moreover, the cooperative team reward significantly dropped when compared to the optimal reward which means the reduction in the cooperative team reward was not caused by leading the adversarial agent closer to the target.

After carefully observing the environment's animation, we found that the other cooperative agent decided to move away from the target to prevent adversarial agent from knowing the target landmark. This is an interesting finding that our attack was able to leak the information to the adversarial agent in short episodes while the degradation in the performance in long episodes was not because of the leakage itself,

instead triggered by the optimal policy of the cooperative agent moving away from the target to distract the adversarial agent. Our next step in this direction is to automate this attack by training a RL policy to plan the malicious actions in certain directions and distance to avoid triggering the other cooperative agent's policy of moving away from the target landmark.

5.2.2 Qualitative Performance and Behavioral Analysis

We evaluate the qualitative performance of our attacks using environment-specific metrics. In cooperative navigation environment, we use $occupied_{landmarks}(A')$ to measure the average number of the occupied landmarks by the cooperative agents. In physical deception environment, we use $coop_{dist}(A')$ to measure the average distance between the cooperative agents and the target landmark.

Table 1 shows the performance for each attack. Occupied landmarks per timestep decrease as the compromised agent deviates more from its optimal policy. Similarly, the distance from the closest cooperative agent to the target landmark increases as we attack more. The random and strategically-timed attacks achieved higher impact than the counterfactual reasoning attack.

Qualitatively, in the cooperative navigation, the compromised agent learns to move away from the landmarks to distract its teammates from covering all landmarks suggesting that this agent succeeds by manipulating its teammates' observations through its actions. In this environment, there is not much room for the agents to recover from the compromised agent's adversarial actions. In the physical deception environment, we notice that when the other cooperative agent goes towards the non-target landmark, expecting the compromised agent to cover the target, it does not alter its behavior to cover the target. This is a robustness issue in MADDPG algorithm. Ideally, the agents should be robust enough to not have to rely on a teammate who is not doing its job. Similarly, we see that when the compromised agent leaves the non-target landmark uncovered and moves towards the target landmark, the adversary does not always take advantage of this clue. The optimal behavior of the adversary would be to move towards the only landmark which is being covered by a cooperative agent, instead it stays still. This behavior once again shows a lack of robustness in the agents under our attack.

5.3 DISCUSSION ON DEFENSE

We believe the reason behind the vulnerability of compromised agents in centralized-training decentralized-execution paradigm of c-MARL is due to ignoring the implicit connections of agents' actions—the impact of one on the others in the moment and subsequently—which allows the use of one agent as a passive adversary. c-MARL agents trained with such algorithms have difficulty identifying the normal and abnormal behaviors of other agents since their policies were not trained to consider the impact of their actions on each other. To combat this problem, we develop different anomalous behavior detection models using a 4 layer Dense deep learning, a 3 layer binary LSTM, Random Forest, SVM classifier, K-nearest Neighbors, predictive LSTM, and an ensemble of the binary LSTM and the predictive LSTM models. We trained those models on clean datasets with $220,000$ samples of normal datapoints generated from each environment with fully cooperative agents. We found that the ensemble model achieved the highest precision and the least false-negative rate across all models when tested on datasets generated by our attacks. In Table 3, We show the precision percentage and the average false negative (FN) of the attacks detection using the ensemble model at different attacking rates. Our detection model achieves better precision when the attack rate increases across all attacks suggesting that at least 40% of the attacks can easily be traced and avoided as early as at the attack rate of 25% of the episode time. Furthermore, the results show that the counterfactual reasoning based attack evades the detection more than other attacks with higher false negatives in both environments in the white and black box settings.

As shown in 3, equipping c-MARL agents with anomalous behavior detection would help them to evaluate the abnormality of each other's actions before considering their actions in the optimization process. The naive remedy once a compromised agent has been detected is to exclude it from the cooperative agents and mark it as an adversary. However, the deviation in one agent's behavior could occur due a security threat or its lack of knowledge about the environment which is reasonable considering the limited vision range and communication bandwidth in MARL systems. Thus, we think it is time for the research community to extend the assured reinforcement learning (ARL) techniques into c-MARL by embedding formal verification in c-MARL algorithms. Real-time formal verification methods would shield the agents from reaching the dangerous states by training them to trade off between performance and security/safety in the environments containing either malicious (adversarial) or faulty (dysfunctional) agents. Those methods should be developed for run-time verification focusing only on the agent's time-bounded, short-term behavior for scalability. This will increase the robustness of MARL systems functioning in uncertain and unpredictable environments. We leave investigating this idea as part of our future direction.

To solve the problem of the non-correlated factorization of the centralized learning-decentralized execution paradigm in MARL, recent work [Wen et al., 2019, Jaques et al., 2019, Tian et al., 2019] has taken other agents' actions into consideration by decoupling the joint policy as a correlated policy conditioned on the environment state and other agents' ac-

Table 3: The detection results of the attacks using the ensemble LSTM model (*precision%|recall%*) in the cooperative navigation and physical deception environments across the different attack rates in both white and black box settings.

Cooperative Navigation												
	White Box						**Black Box**					
Attack Rate	Random		Timed		Counterfactual		Random		Timed		Counterfactual	
25%	60%	96%	64%	98%	54%	92.5%	61%	97%	62%	99%	55%	92%
50%	85%	98.5%	85.5%	98%	82%	92.5%	84.5%	98%	83.6%	98%	82.6%	92%
75%	94.5%	99%	99%	95%	95%	94%	94%	99%	94.4%	97%	95%	94%
100%	100%	98%	100%	99%	100%	96%	100%	99%	100%	95.5%	100%	96%
Physical Deception												
	White Box						**Black Box**					
Attack Rate	Random		Timed		Counterfactual		Random		Timed		Counterfactual	
25%	51%	96%	54%	97%	45%	91%	52%	97%	55%	99%	40%	97%
50%	77%	97%	77%	96%	74%	92%	78%	97%	77%	97%	72%	94%
75%	90%	97%	90%	95.5%	90%	93%	91%	98%	89%	96%	91%	88%
100%	100%	96%	100%	94.5%	100%	92%	100%	97%	100%	94%	100%	84%

tions. Those algorithms assume that the agents participating in a joint activity act rationally and cooperate to achieve shared goals. However, relying only on such assumption to achieve joint goals without the establishment and reinforcement of trust opens a vulnerability similar to the one we discuss in this work. Studying the compromised agent vulnerability in this paradigm of c-MARL algorithms will thus be part of our future work.

6 CONCLUSION

We investigate the maleficence of c-MARL by targeting and weaponizing one agent, termed *compromised*, to follow an adversarial policy that pushes activation of cooperating agents' policy networks off-distribution. Using 2 particle environments as a benchmark in a white-box and black-box settings, we proposed 4 non-targeted attack strategies to control the compromised agent: 1) counterfactual reasoning based attack, 2) randomly-timed attack, and 3) strategically timed attack. We demonstrated the possibility of controlling the total team reward and its task success by compromising only one agent. We intend to extend our threat model into targeted attacks to lure the agents into risky states. Our attack methods can be utilized to evaluate the robustness of c-MARL algorithms [1].

References

Gürdal Arslan and Serdar Yüksel. Decentralized q-learning for stochastic teams and games, 2016.

J. Andrew (Drew) Bagnell. An invitation to imitation. Technical Report CMU-RI-TR-15-08, Carnegie Mellon University, Pittsburgh, PA, March 2015.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients, 2017.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning, 2016.

Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness, 2018.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning, 2020.

J. Gupta, M. Egorov, and Mykel J. Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *AAMAS Workshops*, 2017.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games, 2016.

Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies, 2017.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning, 2019.

Chi Jin, Praneeth Netrapalli, and Michael I. Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *ArXiv*, abs/1902.00618, 2019.

Uk Jo, Taehyun Jo, Wanjun Kim, Iljoo Yoon, Dongseok Lee, and Seungho Lee. Cooperative multi-agent reinforcement learning framework for scalping trading, 2019.

[1] We will make our code available in the camera-ready version

Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies, 2017.

Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. On the robustness of cooperative multi-agent reinforcement learning, 2020.

Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. Tactics of adversarial attack on deep reinforcement learning agents, 2019.

Minghuan Liu, Ming Zhou, Weinan Zhang, Yuzheng Zhuang, Jun Wang, Wulong Liu, and Yong Yu. Multi-agent interactions modeling with correlated policies, 2020.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments, 2020.

Eric Mazumdar, Lillian J. Ratliff, and S. Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131, Jan 2020. ISSN 2577-0187. doi: 10.1137/18m1231298. URL http://dx.doi.org/10.1137/18M1231298.

Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P. How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability, 2017.

A. Peake, J. McCalmon, B. Raiford, T. Liu, and S. Alqahtani. Multi-agent reinforcement learning for cooperative adaptive cruise control. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 15–22, 2020. doi: 10.1109/ICTAI50040.2020.00013.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning, 2018.

L. S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. ISSN 0027-8424. doi: 10.1073/pnas.39.10.1095. URL https://www.pnas.org/content/39/10/1095.

Aaron Sidford, Mengdi Wang, Lin F. Yang, and Yinyu Ye. Solving discounted stochastic two-player games with near-optimal time and sample complexity, 2019.

Jayakumar Subramanian and Aditya Mahajan. Reinforcement learning in stationary mean-field games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 251–259, Richland, SC, 2019. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450363099.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249.

Yichuan Charlie Tang. Towards learning multi-agent negotiations via self-play, 2020.

Zheng Tian, Shihao Zou, Ian Davies, Tim Warr, Lisheng Wu, Haitham Bou Ammar, and Jun Wang. Learning to communicate implicitly by actions, 2019.

Ermo Wei, Drew Wicke, David Freelan, and Sean Luke. Multiagent soft q-learning, 2018.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning, 2019.

Yingce Xia, Tao Qin, Wei Chen, Jiang Bian, Nenghai Yu, and Tie-Yan Liu. Dual supervised learning, 2017.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning, 2018.

Bora Yongacoglu, Gürdal Arslan, and Serdar Yüksel. Learning team-optimality for decentralized stochastic control and dynamic games, 2019.

S. Zazo, S. Valcarcel Macua, M. Sánchez-Fernández, and J. Zazo. Dynamic potential games with constraints: Fundamentals and applications in communications. *IEEE Transactions on Signal Processing*, 64(14):3806–3821, 2016.

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Fully decentralized multi-agent reinforcement learning with networked agents, 2018.

Kaiqing Zhang, Erik Miehling, and Tamer Başar. Online planning for decentralized stochastic control with partial history sharing, 2019a.

Li Zhang, Wei Wang, Shijian Li, and Gang Pan. Monte carlo neural fictitious self-play: Approach to approximate nash equilibrium of imperfect-information games, 2019b.

Yiren Zhao, Ilia Shumailov, Han Cui, Xitong Gao, Robert Mullins, and Ross Anderson. Blackbox attacks on reinforcement learning agents using approximated temporal information, 2019.