Adversarial Machine Learning Attacks and Defences in Multi-Agent Reinforcement Learning

MAXWELL STANDEN, School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, Australia and Defence Science and Technology Group, Edinburgh, Australia
JUNAE KIM, Defence Science and Technology Group, Edinburgh, Australia
CLAUDIA SZABO, School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, Australia

Multi-Agent Reinforcement Learning (MARL) is susceptible to Adversarial Machine Learning (AML) attacks. Execution-time AML attacks against MARL are complex due to effects that propagate across time and between agents. To understand the interaction between AML and MARL, this survey covers attacks and defences for MARL, Multi-Agent Learning (MAL), and Deep Reinforcement Learning (DRL). This survey proposes a novel perspective on AML attacks based on attack vectors. This survey also proposes a framework that addresses gaps in current modelling frameworks and enables the comparison of different attacks against MARL. Lastly, the survey identifies knowledge gaps and future avenues of research.

CCS Concepts: • **Computing methodologies** → **Multi-agent reinforcement learning**; **Adversarial learning**; *Stochastic games*; *Partially-observable Markov decision processes*; Markov decision processes; **Neural networks**; **Multi-agent systems**; • **General and reference** → **Surveys and overviews**.

1 Introduction

Deep Reinforcement Learning (DRL) has made significant progress over the past decade, from its start playing Atari games [84], to beating humans in board games [116] and video games [92, 133], to now addressing important complex safety-critical challenges such as cybersecurity [117], power network management [82], and autonomous driving [102]. As the complexity of these tasks has increased, the focus has shifted from fully-observable single-agent environments to partially-observable multi-agent environments. However, to realise the potential positive societal impacts of Multi-Agent Reinforcement Learning (MARL), it is crucial to address its vulnerability to Adversarial Machine Learning (AML) attacks [123], which exploit the underlying machine learning techniques.

The vulnerability of MARL to AML attacks is challenging to predict due to cascading effects. Attacks against DRL may influence not only the immediate reward but also future rewards [77, 104, 120], which makes it difficult to accurately predict the vulnerability of a DRL agent. Attacks against Multi-Agent Systems (MAS) affect the interactions between agents and may impact agents that were not directly targeted [131]. MARL combines DRL and MAS, featuring both the complexities within those two fields and emergent complexities as a result of their combination. An attack against MARL may cause a cascade of minor effects across time and agents that eventually result in a major system failure. Due to this complexity, it is vital to understand the extent of the vulnerability of MARL to AML attacks across all possible perspectives such that they can be mitigated appropriately. To that end, our survey includes attacks against DRL and MAS due to their applicability to MARL.

Authors' Contact Information: Maxwell Standen, School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, South Australia, Australia and Defence Science and Technology Group, Edinburgh, Australia; e-mail: maxwell.standen@adelaide.edu.au; Junae Kim, Defence Science and Technology Group, Edinburgh, Australia; e-mail: junae.kim@defence.gov.au; Claudia Szabo, School of Computer and Mathematical Sciences, The University of Adelaide, Adelaide, South Australia, Australia; e-mail: claudia.szabo@adelaide.edu.au.

ACM 1557-7341/2024/12-ART
https://doi.org/10.1145/3708320

To effectively mitigate AML attacks against MARL, a thorough understanding of the available defences is required [88]. Some defences mitigate specific vulnerabilities, thus an effective general defence may require combining multiple defences. Defences for MARL may introduce new vulnerabilities or themselves degrade the performance of the algorithm compared to the undefended system [88]. Our work seeks to enhance the understanding of the impacts and limitations of defences for MARL and identify gaps in these defences that are yet to be addressed.

In this survey, we focus on the execution-time AML attacks that degrade the performance of a victim and defences to those attacks. This research direction is driven by our recognition of the importance of comprehending and addressing the complexities of adversarial attacks and corresponding defence strategies within the domain of MARL. These adversarial attacks, characterised by their ability to manipulate agent behaviour during run-time, exhibit significant influence across practical applications such as, robotics [94, 127], power management [93, 134], search and rescue [107], and cyber-security [85]. By narrowing the scope of our survey to this specific category of attacks, our objective is to provide an extensive and comprehensive analysis of execution-time attack strategies, their implications in MARL, and potential defensive techniques. This specialised approach enables us to conduct a deeper exploration of these specific attack and defence mechanisms, including a thorough examination of relevant case studies. Despite previous surveys on AML in the context of DRL [2, 18, 24, 51, 101], a notable gap exists in the coverage of AML attacks and defences for MARL, even within the specific context of execution-time AML vulnerabilities. Our work builds on these previous surveys to cover these critical gaps, thus providing coverage of AML attacks and defences for MARL.

In this work, we address the use of AML techniques on MARL, by employing a new attack perspective called attack vectors and proposing classifications for AML attacks and defences. Our taxonomy of AML attacks covers how an attack may be deployed, what information it uses, and the attack goal. Our taxonomy of AML defences covers the type of defence, the attack vectors a defence can counter, and when a defence is deployed. To address existing gaps in modeling formalisms, we propose a new framework for modelling a variety of AML attacks against MARL and DRL called Adversarial Partially Observable Stochastic Game (APOSG). The contributions we make in this work are fivefold:

- A survey of AML attacks and defences as applied to MARL, DRL, and Multi-Agent Learning (MAL) more broadly.
- A cyber-security inspired perspective on AML attacks against MARL, attack vectors.
- An improved categorisation of AML defences, which better categorises the different defences against execution-time attacks against MARL.
- A new framework for modelling AML attacks against MARL and DRL that describe combinations of attack vectors, attack magnitude, and tempo.
- An in-depth discussion of future research directions.

In Section 2, we provide a background on AML, DRL, and MARL, along with a review of related works. In Section 3, we discuss our methodology. In Section 4, we present our taxonomy of AML attacks against DRL. In Section 5, we discuss our proposed attack vectors and the attacks we found. In Section 6, we discuss AML defences for MARL, DRL, and MAS. In Section 7, we propose a new framework for modelling AML attacks against MARL and DRL. In Section 8, we discuss research gaps and recommend future work.

2 Background

To better capture the landscape of AML attacks on MARL, we focus in this section on AML, DRL, and MARL. Our background on AML focuses on the discovery of evasion attacks against deep learning algorithms, and the subsequent defences employed to mitigate that vulnerability. Our background on DRL covers the basics of

reinforcement learning and an example DRL algorithm. Our background on MARL covers the different types of reward structures, a focus on cooperative MARL, and the use of communication in MARL.

The field of AML uses a large amount of acronyms. To aid with readability, Table 1 features a list of acronyms used for AML attacks and defences that we found in our survey.

2.1 Adversarial Machine Learning

AML is concerned with identifying vulnerabilities and mitigations for machine learning algorithms. Execution-time AML attacks against supervised learning have largely focused on *Evasion* attack [124], which occur when slight human-imperceptible changes to inputs return significantly different outputs from neural networks [123]. AML defences for evasion attacks in supervised learning have been broader and include adversarial training [111], regularisation [137], adversarial detection [137], data preprocessing [115], and ensembles [124].

Evasion attacks may be *targeted* or *untargeted*. Untargeted evasion attacks, such as Fast Gradient Sign Method (FGSM) [35] and Projected Gradient Descent (PGD) [81], aim to cause the victim to predict a different class. Targeted evasion attacks, such as Jacobian Saliency Map (JSM) [96] and the Carlini and Wagner method (C&W) [14], aim to cause the victim to output a specific class. Targeted attacks can also be created using variations of untargeted attacks, for example, the One-step target class methods [68] alter the loss function in the FGSM attack.

Evasion attacks also require a certain amount of knowledge about a victim. Attacks that require full knowledge of a victim are known as *White-box* attacks. *Black-box* attacks only require the ability to query the victim. Transfer attacks [95] allow attacks that originally required white-box information to instead only require black-box information. Transfer attacks use black-box information to train a surrogate victim that replicates the behaviour of the original victim, and then use white-box techniques against the surrogate to discover attacks that are also effective against the original victim. There are also *grey-box* attacks, which only require partial knowledge of the victim [124].

To counter evasion attacks, defensive techniques are used to mitigate the vulnerability. Adversarial training [35] mitigates AML attacks by training a victim against both the original data and adversarially-perturbed data. However, adversarial training may fail to defend against other attacks that it was not trained against [111]. Regularisation [137] alters the training process of the algorithm to improve its robustness against AML attacks. Regularisation may include using additional terms, such as the Lipschitz constant [36], which aims to prevent sudden changes in the output caused by slight perturbations to the input. Preprocessing input data has been shown as an effective AML defence [115] and includes techniques such as autoencoders [29]. Input data can also be altered to remove adversarial perturbations [137], if they can first be detected, thus adversarial detection [137] is also a key AML defence. Ensembles [124] are also a type of AML defence, which trains multiple algorithms which collectively decide the output.

2.2 Deep Reinforcement Learning

DRL uses neural networks for reinforcement learning [1]. Reinforcement learning is a type of machine learning that is concerned with sequential decision-making, which may be modelled with frameworks such as a Markov Decision Process (MDP) (Definition 1). The goal of reinforcement learning is to learn a policy that selects actions that maximise the expected total reward.

Definition 1. *A Markov Decision Process (MDP) is defined by the 4-tuple* (S, A, T, R)*, in which* S *is the set of states,* A *is the set of actions,* T *is the state transition function that determines the probability of transitioning to the next state* $s_{t+1} \in S$ *given state* $s_t \in S$ *and action* $a_t \in A$*, and* R *is the reward function such that the reward* $r_t = R(s_t)$ *and* $r_t \in \mathbb{R}$*.*

DRL techniques such as Deep Q-Networks (DQNs) [84] use a neural network to learn a policy, which maps states to actions. DQNs use an ϵ-greedy policy, in which the agent will explore by randomly selecting an action

Table 1. Acronyms used in AML attacks and defences

Acronym	Definition	Acronym	Definition
A2PD	Adversary Agnostic Policy Distillation	MF-GPC	Model-Free Gradient Perturbation Controller
ACT	Adversarial Cheap Talk	MO-RL	Multi-Objective Reinforcement Learning
AD	Action Distortion	NAA	Naive Adversarial attack
ADMAC	Active Defense Multi-Agent Communication Framework	NR	Noisy-Robust
AdvEx-RL	Adversarial Behavior Exclusion for Safe RL	OS	Online Sequential
AME	Ablated Message Ensemble	OSFW(U)	"obs-seq-fgsm-wb" Universal
ARTS	Antagonist-Ratio Training Scheme	OWR	Optimized worst adversarial policy with regularization
ATLA	Alternating Training with Learned Adversaries	PA-AD	Policy Adversarial Actor Director
ATN	Adversarial Transformation Networks	PGD	Projected Gradient Descent
BCL	Bootstrapped Opportunistic Adversarial Curriculum Learning	PR	Probabilistically-Robust
CARRL	Certified Adversarial Robustness for Deep RL	PRIM	Power Regularization via Intrinsic Motivation
CBAP	Criticality-Based Adversarial Perturbation	RAD	Regret-based Adversarial Defense
CBTS	Criticality-Based Timing Selection	RAD-CHT	Regret-based Adversarial Defense Cognitive Hierarchical Theory
CIQ	Causal Inference Q-network	RARL	Robust Adversarial Reinforcement Learning
CPPO	CVaR-Proximal-Policy-Optimization	RMA3C	Robust Multi-Agent Adversarial Actor-Critic
CROP	Certifying Robust Policies for RL	ROMANCE	Robust Multi-Agent Coordination via Evolutionary Generation of Auxiliary Adversarial Attackers
C&W	Carlini and Wagner method	RORL	Robust Offline RL
DQWAE	Deep Q-W-network regularized with an Autoencoder	RS	Robust Sarsa
EA	Enchanting Attack	RTA	Recovery-Targeted adversarial
FGSM	Fast Gradient Sign Method	SA	State Adversarial
FD	Finite Difference	SAFER	robuSt vAriational ofF-policy lEaRning
GAN	Generative Adversarial Network	SBPR	Sample Based Power Regularization
GM	Grid Manipulation	SCR	Semi-Contrastive Representation
INRD	Identification of Non-Robust Directions	SFD	Adaptive sampling based FD
JSM	Jacobian Saliency Map	SRIM	Static Reward Image Map
LA	Learned Adversaries	ST	Strategically-timed
LSP	Long Short Periodic	TAS-RL	Task-Agnostic Safety for Reinforcement Learning
MACRL	Multi-Agent Communicative Reinforcement Learning	TF	Tentative Frame
MAD	Maximal Action Difference	TRC	Trust region-based safe RL method with CVaR constraints
MADRID	Multi-Agent Diagnostics for Robustness via Illuminated Diversity	UAP	Universal Adversarial Perturbation
MAGI	Multi-Agent Communication via Graph Information Bottleneck	VF	Value Function

with probability ϵ, otherwise it will select the action with the greatest Q-Value. A Q-Value is the expected total reward of a state-action pair, and incorporates both the immediate reward of an action and expected future rewards.

DRL has also been shown to be effective in partially observable environments through the use of recurrent networks. Partially-Observable MDPs (POMDPs) (Definition 2) are a common framework for representing partially-observable decision-making problems. An example of a real-world POMDP could be the treatment of a medical condition, in which the underlying cause is unknown and only the symptoms of the condition may be observed. To provide a concrete context, we will use this example throughout the paper, as it illustrates the key concepts and methodologies in a practical scenario, helping to bridge the gap between theoretical discussions and real-world applications. POMDPs separate the observation from the state and require a decision-making agent to construct a belief about the true state of the environment from a series of partial-observations. Deep Recurrent Q-Networks (DRQNs) [47] are able to construct a belief about the state in the form of a latent state representation from observations through the use of a Long Short Term Memory (LSTM) network.

Definition 2. *A Partially Observable Markov Decision Process (POMDP) is defined by the 6-tuple* (S, A, T, Ω, O, R)*, in which S, A, T and R are the same as defined in Definition 1,* Ω *is the set of observations, and O is the observation probability function that determines the probability of observing* $o_t \in \Omega$ *given the current state* $s_t \in S$ *and previous action* $a_{t-1} \in A$*.*

2.3 Multi-Agent Reinforcement Learning

MARL extends DRL to a multi-agent context. Multi-agent environments can be cooperative, competitive, or mixed based on the rewards provided to the agents [12]. *Cooperative environments* feature monotonic rewards, in which an increase in the reward of a single agent corresponds to an increase in the total reward of all agents. Decentralised POMDPs (Dec-POMDPs) [91] (Definition 3) are a common framework for representing cooperative multi-agent decision-making problems. The extension of a POMDP to a Dec-POMDP allows us to expand our medical treatment example from above to include multiple specialists treating a patient, where each specialist may have access to different skills but must cooperate to achieve the best outcome for the patient. Dec-POMDPs separate the single observation and action of a POMDP into a set of observations and set of actions, which are the observations and actions for each agent, respectively. Dec-POMDPs assume a single shared reward, and so are only suitable for fully cooperative problems. *Competitive environments* include but are not limited to zero-sum rewards, where an increase in one agent's reward directly reduces another agent's reward. *Mixed environments* may feature a combination of competitive and cooperative elements in their reward structure. An example of a mixed environment features two competing teams, with agents on the same team receiving the same reward and agents on opposing teams competing for a zero-sum reward. A Partially-Observable Stochastic Game (POSG) [45] (Definition 4) is a framework that is better suited to competitive and mixed multi-agent decision-making problems. Games, including POSGs, sometimes use varied terminology for agents, including players and controllers, however, we will use the term *agent* henceforth. A POSG allows us to expand our medical treatment example to include agents whose primary concern may not be the well-being of the patient but instead be for a different motivation such as profit. A POSG provides agents individual reward functions.

Definition 3. *A Decentralised Partially Observable Markov Decision Process (DEC-POMDP) is defined by the 7-tuple* $(I, S, \hat{A}, T, \hat{\Omega}, O, R)$*, in which I is the set of agents, S and R are as defined in Definition 1,* $\hat{A}$ *is the joint set of action sets,* A_i *is the action set of agent* $i \in I$*, T is the state transition function that determines the probability of transitioning to a next state* $s_{t+1} \in S$ *given a state* $s_t \in S$ *and joint action* $\hat{a}_t \in \hat{A}$*,* $\hat{\Omega}$ *is the joint set of observations,* Ω_i *is the set of observations of agent* $i \in I$*, O is the joint observation probability function that determines the probability of the joint observation* $\hat{o_t} \in \hat{\Omega}$ *given the current state* $s_t \in S$ *and the previous joint action* $\hat{a_{t-1}} \in \hat{A}$*.*

Definition 4. *A Partially Observable Stochastic Game (POSG) is defined by the 7-tuple* $(I, S, \hat{A}, T, \hat{\Omega}, O, \hat{R})$*, in which,* I*,* S*,* $\hat{A}$*,* $\hat{\Omega}$*,* O *are defined in Definition 3, and* $\hat{R}$ *is the set of reward functions* R_i *for agent* $i \in I$*, such that* $r_{i_t} = R_i(s_t)$*, where* $r_{i_t} \in \mathbb{R}$*.*

A unique feature of MARL is emergent communication [79], which occurs when agents learn to communicate. Communication is a common feature in cooperative partially-observable environments, because agents benefit from coordinating their actions and sharing information. There are two broad categories of communication, implicit communication and explicit communication. *Implicit communication* occurs when agents are able to view the state change caused by the actions of other agents [42]. For example, an agent may observe another agent moving toward a particular location and may be able to infer that the goal is at that location. *Explicit communication* occurs when an environment features a communication channel that allows agents to directly send messages to other agents [27]. Broadcasting is a common paradigm for explicit communication, in which agents produce a single message that is sent to all other agents in the system. Creating the message to send and interpreting messages from other agents are learnt behaviours of the agents.

2.4 Related Work

The combination of AML and MARL has seen little exploration in previous surveys that have covered AML for DRL [2, 18, 51]. Ilahi et al. [51] and Chen et al. [18] cover AML attacks against DRL and defences against those attacks, which includes some AML attacks against MARL. However, due to the focus of these surveys on DRL, they do not explore larger implications of combining AML and MARL. More broadly, there have been surveys such as Bai et al. [2] that focuses on the AML defence of adversarial training, and that by Ren et al. [111], that covers AML attacks and defences for deep learning. There has been no coverage of AML defences for MARL in any previous surveys to the best of our knowledge. Our work seeks to address the significant gap in coverage of AML attacks and defences for MARL.

3 Methodology

We used a literature review methodology based on the approach outlined by Kitchenham et al. [57]. We first identified and refined a number of research questions. From these questions, we identified keywords and constructed search strings. The search strings were used to find relevant papers published between 2010 and 2024 in top-tier conferences that publish AML, MARL, MAL, and DRL papers. We augmented this collection by forward snowballing [139] papers on the topic to discover very recent breakthroughs in the topic, for a total of 801 papers. From this collection, we rejected papers that did not match our accept/reject criteria, resulting in 93 papers. Finally, we extracted data from these papers using a set of guiding questions.

3.1 Research Questions and Search Strings

For a broad overview of AML attacks against MARL, MAL, and DRL, our research questions are:

- RQ1: What AML attacks exploit vulnerabilities in MARL, MAL, and DRL during execution?
- RQ2: What AML defences mitigate execution-time attacks against MARL, MAL, and DRL?

We used two search strings to find papers to address our research questions which were:

- `reinforcement AND (training OR learning) AND (adversarial OR attack)`
- `(multiagent OR multi-agent OR (multi AND agent)) AND (policy OR learning) AND (adversarial OR attack)`

We limited our search to high-quality conferences that published work in AML, MARL, MAL, and DRL since 2010. Our search strings were used in Scopus and external proceedings servers where necessary to search the following conferences[1]:

- Intl. Joint Conference on Autonomous Agents and Multiagent Systems
- Intl. Conference on Machine Learning
- Usenix Security Symposium
- Intl. Joint Conference on Artificial Intelligence
- Intl. Conference on Learning Representations
- National Conference of the American Association for Artificial Intelligence
- IEEE Symposium on Security and Privacy
- ACM Conference on Computer and Communications Security
- ACM International Conference on Knowledge Discovery and Data Mining
- Advances in Neural Information Processing Systems

Finally, we used several well known papers that address our research questions [8, 120, 131] and used a forward snowball search method [139] to find additional relevant papers. These papers were chosen because they made significant discoveries that are relevant to our search.

From our initial collection, we filtered out relevant papers using accept and reject criteria. For a paper to be excluded, it must either fail to meet all the accept criteria or meet any of the reject criteria. Our accept criteria were:

- Uses either an execution-time adversarial attack or a defence against an execution-time adversarial attack
- The attack or defence are used on either an offline deep reinforcement learning or a cooperative multi-agent decentrally-executable deep learning algorithm

Our reject criteria were:

- Number of pages is less than five
- Does not feature any experimentation

We define *adversarial attacks* as the intentional manipulation of an aspect of an environment, including actuators, sensors, and communication channels, to reduce the performance of a target agent by causing that agent to take irrational or self-defeating actions. We further limit the scope of our study to execution-time attacks. This definition excludes a number of AML attacks, including, Trojan implantation, data poisoning, and model inference attacks.

4 A Taxonomy of AML Attacks

The practical use of MARL and DRL requires an understanding of the vulnerabilities in the systems produced by those algorithms. As researchers, we aim to build this understanding by classifying known attacks. We have identified three general properties of AML attacks against MARL algorithms, namely, *Attack Vector*, *Information*, and *Objective* and present this taxonomy in figure 1.

Attack Vectors are the means by which an adversary performs an attack. We identify five attack vectors, namely *Action Perturbations*, *Observation Perturbations*, *Communication Perturbations*, *Malicious Communications*, and *Malicious Actions*. Attacks using action perturbations, observation perturbations, and communication perturbations introduce changes into the actions selected by agents, the observations of agents, and the messages sent between agents respectively. Attacks using malicious communications and malicious actions use non-cooperative agents in the environment to send messages to victim agents and take actions that affect the state transition respectively. We discuss these attack vectors in more detail in Section 5.

[1]This list does not cover 2010 to 2024 for the relevant conferences, as some did not run in some years

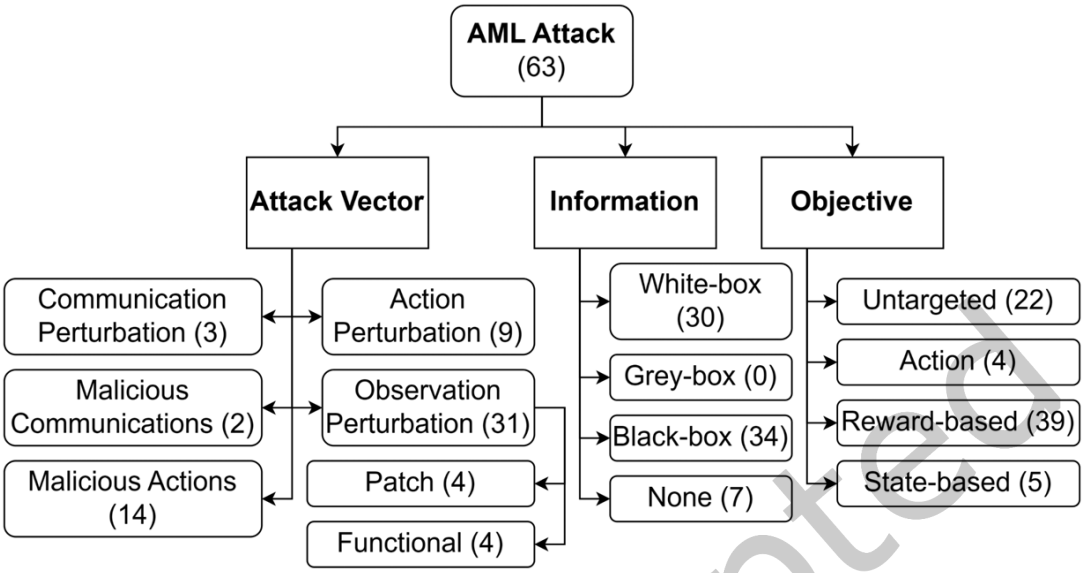

Fig. 1. Classification of Adversarial Machine Learning (AML) attacks against Deep Reinforcement Learning (DRL) and the number of papers that considered those categories.

Our classification considers the adversary's knowledge capability by classifying what *information* the adversary must know about the target algorithm in order to use an attack. We use the categories of *white-box* information and *black-box* information as presented by Ilahi et al. [51], *grey-box* information, and an additional category '*None*' for attacks that do not require any information about the victim to craft an attack. White-box information assumes an adversary has full access to the victim's system, including its architecture and parameters. Black-box information assumes an adversary has access to the inputs and outputs without knowing its internal details. Grey-box information, which is featured in taxonomies of AML attacks against supervised learning [124], assumes the adversary has partial information about the victim, such as its structure or training data. However, we did not discover any attacks that used grey-box information.

We also consider the *Objective* of an attack. A common objective is to alter the behaviour of a target. This objective, known as an *untargeted* attack, assumes that the victim agent is well-trained, thus altering its behaviour in any way will result in worse performance. This objective was originally identified in a taxonomy for AML attacks on supervised learning algorithms [96], which also identified the objective of a targeted attack, which aims to cause the supervised learning algorithm to output a specific label. We can interpret the targeted attack in a RL context as aiming to cause an agent to take a specific *action*. Other objectives unique to reinforcement learning methods are *state-based* and *reward-based*. State-based objectives aim to transition the environment to a specific state, which may correspond to a catastrophic failure of the system. Reward-based objectives aim to alter the reward returned by the environment during execution, and include goals such as maximising the adversary's reward or minimising the reward of the targeted agent.

5 Proposed AML Attack Vectors

Attack vectors are a cyber-security inspired perspective on AML attacks against MARL. An attack vector in cyber-security refers to how an adversary would exploit an attack surface to gain access to a network in order to achieve a goal. We can enumerate the 'attack surfaces' of our agent by looking at the inputs and outputs

of the agent during its execution, namely, actions, observations, and communications. From our literature review, we identify two methods of attacking these surfaces, namely, (i) directly perturbing the object or (ii) introducing a malicious agent into the environment. Perturbation refers to taking an existing input or output and altering it within a set magnitude. This concept also captures replacement of the original, if the magnitude is infinite. This leads us to propose three attack vectors, namely, *Action Perturbations*, *Observation Perturbations*, and *Communication Perturbations*. The second method of introducing a malicious agent refers to the adversary exploiting the multi-agent nature of the environment to introduce an adversarial agent that may then interact with the victim agent, either directly through *Malicious Communications*, or indirectly by taking *Malicious Actions* in the environment. By considering these five attack vectors, we are better able to analyse specific vulnerabilities in MARL algorithms. We discuss these attack vectors in the following subsections.

5.1 Action Perturbations

Action Perturbations are a form of AML attack unique to decision-making problems in which the adversary directly alters the action of an agent. An adversary exploits this attack vector by perturbing the action to cause an unexpected state transition, from which the target agent is unable to optimally act. Thus, by making minor changes to the agent's actions, an adversary can significantly degrade an agent's performance. An Action Perturbation attack is difficult for an adversary to achieve because it requires an adversary to directly affect an actuator either with physical access or a cyber-physical attack. We have identified eleven attacks that perform Action Perturbation that we present in Table 2 along with other key properties, including the information required for the attack, the attack objective, the framework used to originally construct the attack, and the type of action space under attack (discrete or continuous).

Table 2. Action Perturbation attacks

Attack Name	Information	Objective	Framework	Action Space
Targeted Adversarial [71]	Black-box	Reward-based	MDP	Continuous
Myopic Action Space [72]	White-box	Reward-based	MDP	Continuous
Look-ahead Action Space[72]	White-box	Reward-based	MDP	Continuous
Probabilistic Action [127]	None	Reward-based	PR-MDP	Continuous
Noisy Action [127]	None	Reward-based	NR-MDP	Continuous
EDGE [40]	Black-box	Untargeted	MDP	Both
Model-based Attack [138]	White-box	Reward-based	MDP	Continuous
MARLSafe Action Test [38]	Black-box	Reward-based	MMDP	Discrete
StateMask [21]	Black-box	Untargeted	E-MDP	Both
1-step adversarial power [74]	Black-box	Reward-based	Stochastic Game	Discrete
Evolutionary Generation of Attackers [153]	Black-box	Reward-based	LPA-Dec-POMDP	Discrete

MDPs were used as the framework for all the attacks except for those by Tessler et al. [127], which used Probabilistic Action Robust MDP (PR-MDP) and Noisy Action Robust MDP (NR-MDP). The PR-MDP [127] (Definition 5) adds an adversary to the standard MDP that probabilistically replaces the victim agent's action. The NR-MDP [127] (see Definition 6) adds an adversary to the standard MDP that is able to add a perturbation to the victim agent's action at every timestep. In our medical treatment example, the PR-MDP would represent an adversary that may be able to mix a different medication with the prescribed medication, whereas the NR-MDP

would represent an adversary that could slightly increase or decrease the amount of medication the patient is taking. The PR-MDP and NR-MDP are effective frameworks for modelling their respective attacks, namely, Probabilistic Action Perturbation [127] and Noisy Action Perturbation [127]. However, these frameworks are only suitable for modelling the decision-making of a single agent and the lack of a general framework for Action Perturbation attacks limits the ability to compare the effectiveness of the attacks.

Definition 5. *A Probabilistic Action Robust MDP (PR-MDP) is defined by the 6-tuple* (S, A, T, R, v, α)*, in which S, A, T, and R are as defined in Definition 1,* v *is the Probabilistically-Robust (PR) operator such that* $v(s_t) = a'_t$ *where* $a'_t \in A$ *is either an adversarial action with probability* α *or the agent's original action with probability* $1 - \alpha$ *that is used in the state transition function.*

Definition 6. *A Noisy Action Robust MDP (NR-MDP) is defined by the 6-tuple* (S, A, T, R, v, Δ)*, in which S, A, T, and R are as defined in Definition 1,* v *is the Noisy-Robust (NR) operator such that* $v(s_t, a_t) = a'_t$ *where* $a_t, a'_t \in A$ *and* $|a_t - a'_t| < \Delta$*,* a'_t *is the perturbed action that is used in the state transition function, and* Δ *is the magnitude of the perturbation.*

A key feature of Action Perturbation attacks is the type of action being perturbed, namely, continuous or discrete. Continuous Action Perturbation attacks [21, 40, 71, 72, 127, 138] reveal vulnerabilities caused by uncertain actuation, such as imperfections in a motor. This actuation uncertainty is common to the robotics domain applications, thus the simulation environment MuJoCo [129] was commonly used to demonstrate the applicability of the attack. Directly replacing an agent's action can be considered as a type of Action Perturbation with infinite magnitude. Replacement is necessary for attacks against discrete action spaces [21, 38, 40, 74, 153], and has also been demonstrated against continuous action spaces by the Probabilistic Action Perturbation attack [127].

Our survey found three Action Perturbation attacks against MARL [38, 74, 153]. The MARLSafe Action test [38] and the Evolutionary Generation of Attackers [153] attacks create a traitor agent, whose actions are replaced for an entire episode. MARLSafe Action test [38] trains an adversarial agent using deep learning [34] to select the worst-case action, whereas the Evolutionary Generation of Attackers [153] uses an evolutionary algorithm. Both of these attacks demonstrate their effectiveness on the StarCraft Multi-Agent Challenge (SMAC) [114] and measure the degradation of the reward and win-rate as their attack success metrics. The 1-step adversarial power [74] attack, instead considers the effect of replacing an action at a single timestep. The effectiveness of this attack is demonstrated in the two-player fully-cooperative Overcooked game and is measured using the degradation of reward.

The effectiveness of an Action Perturbation attack is dependent on its effects on the target. Most of the Action Perturbation attacks used a reward-based target, due to the immediate reduction in the optimal expected return of an agent. However, by causing the environment to transition to an unexpected state, an Action Perturbation attack may have a secondary effect on the victim agent. The vulnerability of algorithms to transitions to unknown states has explicitly been explored in both discrete [40] and continuous action spaces [127]. The metrics of average episodic reward and win rate that are currently used to evaluate the effectiveness of Action Perturbation attacks fail to differentiate between these two effects. A metric, such as counterfactual regret [159], may better evaluate the vulnerability of an agent to Action Perturbation attacks. The regret of an agent is the difference between the expected cumulative reward given an agent's policy and the best possible cumulative reward. The regret at the point of attack would indicate the level of vulnerability an algorithm has to the worst-case action, and the regret in the steps following an attack would indicate the vulnerability of the agent to the secondary effects of the attack.

5.2 Observation Perturbations

Observation Perturbation attacks, similar to Evasion attacks [124] in supervised learning, expose a vulnerability in an agent to adversarially altered inputs. For an adversary to alter the observation, the adversary must be capable of altering the environment [9] or an agent's sensors, via means such as a cyberattack [26]. The effect of an Observation Perturbation attack is to cause the agent to select a different action. Thus, Observation Perturbations indirectly lead to an Action Perturbation attack. We present many of these Observation Perturbation attacks in Table 3. This table covers the method used to craft the perturbation, if a transfer attack [95] was demonstrated, the method used to determine the timing of the attack, and the framework used by the paper to present the attack.

Observation Perturbation attacks require carefully crafted alterations to induce an adversarial effect. We present a number of perturbation crafting techniques in Table 4. Many of these perturbation crafting techniques were originally used against supervised learning algorithms [14, 35, 81, 96]. The first major category is white-box untargeted attack such as the FGSM [35], PGD [81], and UAP [86]. These attacks can be extended to target specific objectives by altering the loss function as has been demonstrated by it-FGSM [76], Two-Stage PGD [103], GM PGD [19], Directed FGSM [122], and RT-FGSM [134]. Targeted white-box attacks include JSM [96] and C&W [14] which are used to achieve action-based objectives. These white-box perturbation crafting methods can be extended to use black-box information through the use of a transfer attack [95]. A transfer attack uses black-box information to train a surrogate victim, then uses white-box information from the surrogate to craft perturbations, and then uses those perturbations against the original victim. Transfer attacks have been demonstrated against a number of DRL attacks [4, 17, 38, 49, 50, 52, 134, 135]. Alternatively, there are untargeted black-box attacks such as FD [7] and AdvGAN [143]. Some crafting methods were developed specifically for DRL, such as CopyCAT [50], which crafts observation-independent perturbations that cause the victim to follow a specific policy. Chan et al. [17] used a SRIM to generate an attack by evaluating the impact that perturbations of different features had on the cumulative reward of an agent. Alternatively, DRL enables the adversary to learn how to craft perturbations. This approach has been extended to use white-box information from the victim [112]. Other methods use the gradient of the Q-Value [97, 155] to craft reward-based perturbations.

Many of the Observation Perturbation attacks targeted unstructured observations, such as images. However, Wang et al. present an attack that targets the structured observations of an energy management system of an electric vehicle [135]. The data collected from the vehicle was presented to the agent as a vector of continuous values, which was smaller and contained a greater density of information than images. The empirical results showed the effectiveness of the attack, which suggests that attacks against other structured observations, such as those used in autonomous cyber defence [117], may also be effective. However, the observations in an autonomous cyber defence environment may be discrete, and it is unclear how effective these attacks would be against structured discrete observation spaces.

The timing of an attack has been shown to be a key factor in the success of that attack [49, 67, 77, 104, 120]. The simplest timing is the uniform attack [49], which attacks at every time step. However, frequent attacks are computationally expensive and easier to detect [77]. More efficient attacks wait until the value of a state or action is above a certain threshold. TF [104] selects a state to be attacked if the difference between the victim agent's Q-value estimation for its most preferred action and an adversarially selected action exceeds a set threshold. This technique only works for Q-value-based DRL algorithms and would not be effective against policy-based algorithms. Critical State [67] and CBTS [157] both consider an agent's preference for taking an action as the means to decide if a state is worth attacking. Other approaches consider the value of the state as the means to decide if it should be attacked. VF [65] selects a state if the estimation of the value of the current state is above a certain threshold. Critical Point [120] instead attacks a state if the value of the current state and the value of a future state caused by an attack exceeds a certain threshold. DRL has been used as an attack timing

Table 3. Observation Perturbation Attacks

Name of Attack	Perturbation Method(s)	Transfer Attack	Timing Method	Framework
Uniform Attack [49]	FGSM	✓	Uniform	None
Random Noise attack [65]	Random Noise	N/A	Uniform	None
VF Attack [65]	FGSM	×	VF	None
ST attack [77]	C&W	×	ST	MDP
EA [77]	C&W	×	Fixed	MDP
Policy Induction Attack [4]	FGSM, JSM	✓	Uniform	MDP
Contiguous attack [5]	FGSM	×	Uniform	MDP
NAA [97]	Optimal Noise	×	Uniform	None
Gradient-based attack [97]	Q-Value Gradient	×	Uniform	None
Sequential Attack [130]	ATN	×	Uniform	MDP
Optimal Attack [112]	White-box DRL	×	Uniform	Attack MDP
OS Attack [94]	SFD	N/A	Uniform	MDP
SRIM Attack [17]	SRIM	✓	Uniform	MDP
CopyCAT Attack [50]	CopyCAT	✓	Uniform	MDP
MO-RL attack [31]	DRL	N/A	Uniform	MDP
RS [155]	Q-Value Gradient	×	Uniform	SA-MDP
MAD [155]	Q-Value Gradient	×	Uniform	SA-MDP
Snooping attack [52]	FGSM	✓	ST	MDP
Critical Point Attack [120]	C&W	×	Critical Point	MDP
Antagonist Attack [120]	C&W	×	DRL	MDP
OWR [76]	it-FGSM, d-JSM	×	Uniform	DEC-POMDP
FGSM-based attack [135]	FGSM	✓	Uniform	MDP
FD Method [135]	FD	N/A	Uniform	MDP
Model Based Attack [138]	PGD	×	Uniform	MDP
Critical State Attack [67]	Q-Value Gradient, FGSM, Random Noise	×	Critical State	MDP
TF Attack [104]	DRL	N/A	TF	SS-MDP
Two-stage attack [103]	Two-Stage PGD	×	Uniform	SA-MDP
AD Attack [19]	AD PGD	N/A	Uniform	MDP
GM Attack [19]	GM PGD	N/A	Uniform	MDP
CBAP Attack [157]	FGSM	×	CBTS	MDP
PA-AD [122]	Directed FGSM	×	Uniform	PAMDP
Attacks on Beliefs [30]	FGSM	×	Uniform	PuB-MDP
LA [154]	DRL	×	Uniform	SA-MDP
OSFW(U) [126]	FGSM	×	Uniform	SA-MDP
UAP-S [126]	UAP	×	Uniform	SA-MDP
AdvRL-GAN [152]	AdvGAN	N/A	Uniform	MDP
RTA attack [134]	RT-FGSM, RT-JSM	✓	Uniform	MDP
MARLSafe State Test [38]	FGSM	✓	Uniform	MMDP
SCR [148]	PGD	×	Uniform	Goal-Conditioned MDP

Table 4. Perturbation Crafting Methods

Name of Method	Information	Objective
FGSM [35]	White-box	Untargeted
JSM [96]	White-box	Action-based
C&W [14]	White-box	Action-based
UAP [86]	White-box	Untargeted
FD [7]	Black-box	Untargeted
Random Noise [65]	None	Untargeted
ATN [3]	White-box	Reward-based
PGD [81]	White-box	Untargeted
Q-Value Gradient [97, 155]	White-box	Reward-based
Optimal Noise [97]	White-box	Untargeted
AdvGAN [143]	Black-box	Untargeted
White-box DRL [112]	White-box	Reward-based
DRL [31, 112]	Black-box	Reward-based
SRIM [17]	White-box	Reward-based
CopyCAT [50]	White-box	Action-based
it-FGSM [76]	White-box	Action-based
d-JSM [76]	White-box	Action-based
Two-Stage PGD [103]	White-box	Action-based
AD PGD [19]	Black-box	Untargeted
GM PGD [19]	Black-box	Action-based
Directed FGSM [122]	White-box	Action-based
SFD [94]	Black-box	Untargeted
RT-FGSM [134]	White-box	Action-based
RT-JSM [134]	White-box	Action-based

method by training an agent to select if the current state should be attacked [120]. A direct comparison of all of these methods has yet to be made. Instead, we consider the experimental data presented in the papers and conclude that the most efficient attack timing methods are Critical Point, DRL, and Critical state. These methods achieved an effective degradation of the victim's performance by attacking less than 5% of the steps under specific experimental conditions as shown in Table 5.

We found few of the Observation Perturbation attacks targeted MARL. The MARLSafe State Test [38] extends the Uniform attack [49] to attack all agents in the system. This attack provides some evidence that the Qmix MARL algorithm [110] may be more vulnerable to Observation Perturbation attacks than the MAPPO MARL algorithm [150]. The OWR attack uses DRL to find the worst policy that would minimise the victim's reward. OWR [76] then executes this policy with action-based targeted Observation Perturbations. Both of these attacks used Uniform timing, thus failing to consider the time aspect of AML attacks against MARL. OWR attacks a single agent in the SMAC [114], which is a mixed multi-agent environment. Qualitative analysis of the victim's behaviour shows the victim agent hiding from the enemy team and refusing to fight. This results in the victim's team being defeated by the enemy team. This behaviour suggests that the attack only affected the victim agent

[2]The attack frequency has been identified under specific experimental conditions and may not be directly comparable

Table 5. Observation Perturbation Timing Methods

Name of Method	Attack-Step Selection	Proportion of States Attacked
Uniform [49]	All steps are attacked	1
Fixed [77]	Steps from time t to time t+H are attacked	H/Episode Length
ST [77]	If the difference between an agent's most and least preferred actions exceed a threshold	0.25^2
VF [65]	If the value of the current state exceeds a threshold	0.1^2
Critical Point [120]	If the divergence between the current and future attacked state exceeds a threshold	$0.005\text{-}0.02^2$
DRL [120]	If a DRL-trained agent's 'perturb' value exceeds a threshold	$0.0075\text{-}0.03^2$
Critical State [67]	If the difference between most preferred and all other actions exceeds a threshold	$0.0002\text{-}0.0155^2$
TF [104]	If the difference between the Q-Values of the adversarially-induced action and the original action exceeds a threshold	$0.65\text{-}0.75^2$
CBTS [157]	If the difference between an agent's first and second preferred actions exceeds a threshold	$0.24\text{-}0.56^2$

and failed to cause a cascading failure in the other agents. However, due to the mixed nature of the environment, it is difficult to isolate these effects.

Observation Perturbation attacks often used MDPs as the framework for modelling the attack, along with POMDPs, DEC-POMDPs, and other MDP variations. However, these frameworks are unable to accurately model Observation Perturbation attacks. The State-Adversarial MDP (SA-MDP) [155] (Definition 7) was used by some Observation Perturbation attacks and is capable of modelling Observation Perturbation attacks by adding an adversary to an MDP that is able to alter the state-observation that the agent receives. However, SA-MDP can only model Observation Perturbation attacks in a single-agent fully-observable environment. In our medical treatment example, the SA-MDP would model the doctor being told the incorrect pathogen infecting the patient by a malicious actor.

A number of metrics are used to evaluate Observation Perturbation attacks. The reward of the system was the most common metric and is a direct indication of the effect of the attack on the performance of the system. The success or win rate of the system was also used by some of the papers, which is higher-level than reward and only reveals if the attack had a major effect on the performance of the system. For targeted attacks, papers measured the reward of the attacker [148] and the attack success rate [17, 50, 94]. Some of the papers measured other properties of the attack such as the time taken to create the attack [17, 19]. An understanding of the time requirements allows us to understand the risk of real-time attacks and the constraints that may be placed on an adversary.

Definition 7. *A State-Adversarial Markov Decision Process (SA-MDP) is defined by the 5-tuple* (S, A, T, R, v)*, in which S, A, T, and R are defined the same as Definition 1, and* v *is the adversarial state-perturbation function such that* $v(s_t) = s'_t$ *where* $s_t, s'_t \in S$*,* s_t *is the original state and* s'_t *is the state as observed by the agent. This state permutation is only a change to the perception of the agent and does not change the true state of the environment.*

We have identified two unique sub-categories of Observation Perturbation attacks based on the magnitude and localisation of an attack. *Patch attacks* target a small section of the observation, but may use a larger magnitude. *Functional attacks* use both a large magnitude and may affect the whole observation, but are restricted to 'natural'

alterations, such as rotations or colour swaps.

Patch Attacks occur when a small area of an observation is altered. They are easier to realise than other Observation Perturbation attacks, as an adversary only needs to change a small part of the observation. The effectiveness of Patch attacks have been demonstrated against supervised learning algorithms [9] and a physical patch can be used to attack an image recognition algorithm. This patch attack has also been applied against DRL [94].

We have found four papers that use this attack against DRL [10, 48, 80, 94]. A Targeted Physical attack [10] controls the appearance of a fixed-size square object that exists in the observation to affect the victim. UniAck [48] performs both an AML attack and evades AML defences. The attack generates a universal perturbation that alters the action selection and causes the interpretation module of a DRL algorithm to identify the non-perturbed region of an image as the reason behind the action selection. The ACT attack [80] exploits a communication channel that is controlled by the attacker and is directly input into a victim's observation.

Functional Attacks use 'natural' alterations to the observation to degrade the performance of an algorithm and were originally proposed against supervised learning models [69]. These attacks are untargeted and aim to change the observation in a way that a human could easily adapt. These attacks include changing colours, rotating, introducing compression artefacts, or altering the dimensions of the observation.

Functional attacks vary in their ability to be realised. Rotating a sensor to attack an agent may be easier to achieve than changing the colours that the sensor perceives. Functional attacks expose potential weaknesses in a DRL algorithm to basic changes in the environment to which a human would be easily able to adapt. We found four approaches to functional attacks [58–60, 63], that all used an MDP framework.

5.3 Communication Perturbations

Communication Perturbation attacks exploit vulnerabilities in the communications between agents. An adversary may exploit this vulnerability if they are capable of altering the messages sent between agents in a cooperative MAS. This attack vector only considers explicit communications, because implicit communication uses an agent's observation of the environment to communicate information, and so perturbation attacks against implicit communication would be considered a type of Observation Perturbation attack. A common approach to explicit communication is to include messages from other agents in an agent's observation. Thus, there is a similarity between Patch Observation Perturbation attack and Communication Perturbation attacks. An alternative perspective is to consider the overlap between Communication Perturbations and Action Perturbations, as messages sent from other agents may be considered actions. However, separately categorising Communication Perturbation from the other forms of perturbation reveals unique challenges of AML attacks against MAS.

Three papers consider the Communication Perturbation attack vector [121, 131, 144]. Xue et al. [144] present a reward-based black-box attack against MARL-Comms algorithms, namely, CommNet [119], TarMAC [22], and NDQ [136]. Sun et al. [121] present an attack that requires no information about the victim algorithm. Tu et al. [131] present the only attack against a multi-agent supervised learning algorithm discovered in our search. The supervised learning algorithm uses emergent communications to recognise an object with information from multiple perspectives [20]. An adversary is then introduced that perturbs the communications to reduce the effectiveness of the object recognition algorithm. Tu et al. [131] demonstrated both a white-box attack and transfer black-box attack against Multi-View ShapeNet.

AML attacks that perturb communications must consider how communication is being performed in the system. One aspect of the communication is which agent can send messages to which other agent. It is common for algorithms to use a broadcast paradigm, in which all agents send a message to all other agents. All the attacks that we found [121, 131, 144] targeted broadcast communication. Another key aspect of inter-agent communication

is what type of data is being communicated. We can separate this data into two types, namely, continuous and discrete. All the attacks we found [121, 131, 144] targeted continuous data.

Communication Perturbation attacks may be easier for an adversary to use in the real world than Observation Perturbation attacks. Cybersecurity attackers may use a technique called Meddler-In-The-Middle (MITM) [100] which allow them to intercept and alter messages that are sent over a network. Cybersecurity defences against MITM attacks, such as encryption, would be highly effective at stopping a Communication Perturbation attack. However, Communication Perturbation attacks should still be considered due to the risk of cybersecurity defences not being used, being used incorrectly, or if an attacker has a means of bypassing these defences, such as acquiring the encryption key.

Communication Perturbation attacks may also be less constrained than Action or Observation Perturbations. The messages sent by MARL agents may be indecipherable to humans, and so the constraints that are normally imposed, such as magnitude of attack or location of attack, may not apply. Without these constraints, an adversary could completely change the messages sent between agents. This change is similar to Malicious Communication, discussed in Section 5.4, except that the message is coming from a trusted source. None of the techniques that we found use an unlimited white-box attack against communications, however as communications may make up a small part of an agent's observation, we believe that its effectiveness may be similar to that of a Patch attack.

5.4 Malicious Communications

A *Malicious Communication attack* occurs when an adversary agent is able to send messages to a target agent. Adversarial messages are used by both Malicious Communications and Communication Perturbations. However, Malicious Communication attacks craft a new message instead of altering an existing message. This means that the attack may not exploit the trust between agents, or use information inside a message to shape the attack. We believe that defences against Malicious Communications may be more effective than against Communication Perturbations, as messages do not originate from a trusted source. However, Malicious Communications attacks may be more easily realised than Communication Perturbations, as the adversary does not need to intercept and alter an existing message. Malicious Communications are also highly similar to Malicious Actions, which we describe in Section 5.5, because it is possible to represent communications between agents in the action space.

Exploration of Malicious Communications in the literature has been limited [8, 105]. The attacks presented in these papers are limited to using black-box information, however we believe that white-box information could also be used to craft more devastating and effective attacks. Blumenkamp et al. [8] trained then fixed the policy of a MARL system and then trained an adversarial agent, with Malicious Communications capabilities, against that system. However, the primary goal of the adversarial agent was to maximise its own reward, with the reduction in performance of the cooperative system being a secondary effect. Qu et al. [105] also took the approach of maximising the adversary's reward. Their adversary attacked a system of decentralised sensors, with a central agent receiving data from those sensors. The cooperative sensors used a fixed policy to determine what information to communicate.

The communication paradigm employed should influence the effectiveness of Malicious Communications. Broadcast is the most common communication paradigm in MARL [158], in which all agents send a message that is received by all other agents. Neither of the papers we found used broadcast communications. Blumenkamp et al. [8] aggregated received messages and propagated this to nearby agents, such that those agents that were closer had a greater effect on the message that was received. Qu et al. [105] used a many-to-one communication where the many sensors were communicating back to the central agent. We believe that a broadcast malicious message should be highly effective because it is able to simultaneously affect all agents in the system.

5.5 Malicious Actions

Malicious Actions exploit a victim agent's vulnerability by finding naturally occurring environment states and observations that have an adversarial effect on the victim. Malicious Actions are valid actions that an adversary agent may take in mixed and competitive MARL to exploit weaknesses in an opponent's policy. Both Malicious Communication and Malicious Actions use valid actions in the environment to attack a target, however, Malicious Actions may only indirectly affect a target.

This type of attack is the most feasible of all the attack vectors considered, however it is also both the most difficult and least effective attack. The only requirement for an adversary is that it can take actions in an environment. However, this attack may be countered by both AML defences and improvements to the DRL agent generalisation.

An adversary must find states and observations that exist in the environment that have either not been encountered or generalised by the target policy. Several papers we found consider Malicious Actions and their attacks are presented in Table 6 along with the information required for the attack, the objective of the attack, if a transfer attack is used, and the framework used in the paper to present the attack.

A major type of Malicious Action attack was pioneered by Gleave et al. [34] and extended by Guo et al. [39]. These attacks used competitive MARL against a fixed target policy to find the actions that an adversary could perform in the environment that causes the opponent to lose. Gleave et al. [34] use a zero-sum reward to minimise the target's reward, and found effective strategies in MuJoCo. The extension by Guo et al. [39], that relaxed the zero-sum constraint, was highly effective in StarCraft II environments [114].

Table 6. Malicious Action Attacks

Name of Attack	Information	Objective	Transfer Attack	Framework
Failure Search [132]	Black-box	State-based	×	MDP
Adversarial Policy [142]	White-box	Untargeted	×	MDP
Adversarial Policy [34, 39]	Black-box	Reward-based	×	SG
White-box Adversary [93]	White-box	Reward-based	×	MDP
Black-box Adversary [93]	Black-box	Reward-based	✓	Failure-MDP
Antagonists [98]	Black-box	Reward-based	×	POSG
Evasive Malware Generation [85]	Black-box	Reward-based	×	MDP
Adversarial Dynamics Search [94]	Black-box	Reward-based	×	MDP
ISMCTS-BR [128]	Black-box	Reward-based	×	EFG
Adversarial Interference [107]	Black-box	Reward-based	×	DEC-POMDP
Adversarial Policies [15]	White-box	Reward-based	×	2-player MDP
Dynamic Adversary [54]	Black-box	Reward-based	×	DEC-POMDP
Non-Cooperative Behavior [156]	None	Untargeted	×	Agent Dynamic Model
MADRID [113]	Black-box	Reward-based	×	Underspecified Stochastic Games

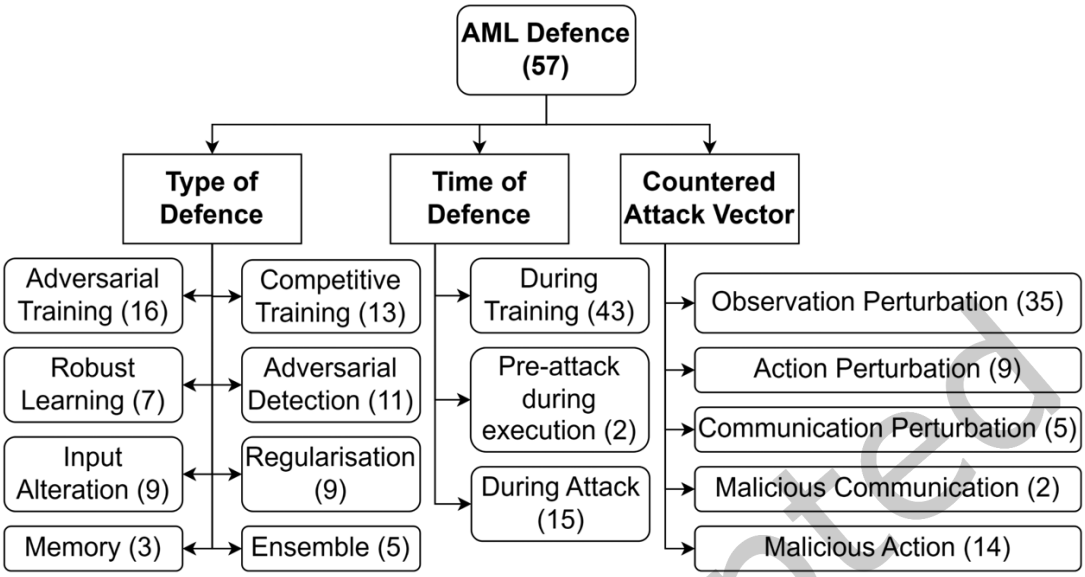

Fig. 2. Classification of AML defences for DRL and the number of papers that considered those categories.

6 AML Defences

We consider several categories in the classification of AML defences for MARL, DRL and MAL, namely, the type of defence, when the defence occurs, and what attacks the defence counters. We have identified several types of AML defences that are used to defend MARL and DRL algorithms from AML attacks. These are Adversarial Training, Competitive Training, Robust Learning, Adversarial Detection, Input Alteration, Memory, Regularisation, and Ensembles. When considering when a defence is applied, we identify four general times, namely, during training, during execution before an attack, and during an attack. Figure 2 shows our classification.

Adversarial Training [35, 81] retrains the original agent against the AML attack. Adversarial training occurs during the training phase of the machine learning pipeline, and its main purpose is to counter Observation Perturbations. It has also been shown to be effective in countering Communication Perturbations [131]. Table 7 presents the adversarial training techniques and the attack vector countered, the defence's impact on performance, and the framework the paper used to present the defence. The impact of a defence on the clean performance of a model shows potential benefits or drawbacks. We use five classifications, namely, *positive* and *negative*, for defences that improve or degrade clean performance respectively, *no change*, for defences that do not significantly change the performance, *mixed*, for defences that show a mixture of positive and negative changes for different evaluation conditions, and *not evaluated*, for papers that do not include enough information to allow a comparison of the clean performance of their defence.

We exclude online training from this category, despite its potential for performing adversarial training during execution. Online training against an adversary poses a risk as the adversary may influence the data collected from the environment. This adversary-influenced data will be used by the algorithm during the online training process. An adversary may intentionally poison this training data to cause the algorithm to learn a poor policy [44, 56]. Thus, online adversarial training needs to handle data poisoning attacks before it can be deployed as an effective defence.

Table 7. Adversarial Training Defences

Name of Defence	Countered Attack Vectors	Impact on Performance	Framework
Adversarial training [59, 62]	Observation Perturbations	Negative	MDP
Robustifying models [131]	Communication Perturbations	Positive	N/A
DQWAE [89]	Observation Perturbations	Positive	MDP
Adversarial training [5, 61, 65]	Observation Perturbations	Not evaluated	MDP
SA DRL [155]	Observation Perturbations	Positive	SA-MDP
RADIAL-RL [90]	Observation Perturbations	Mixed	MDP
Robust training [97]	Observation Perturbations	Positive	MDP
PA-ATLA [122]	Observation Perturbations	Mixed	MDP
CIQ [145]	Observation Perturbations	No Change	POMDP-IO
BCL [141]	Observation Perturbations	Negative	MDP
RMA3C [43]	Observation Perturbations	Not evaluated	SAMG
Semi-Contrastive Adversarial Augmentation [148]	Observation Perturbations	Not evaluated	Goal-Conditioned MDP
SAFER [78]	Observation Perturbations	Mixed	CMDP

Competitive Training [99] improves the robustness of the agents by training them against opponent agents. This defence mitigates Malicious Actions but has also been shown to be effective against Action Perturbations and Communication Perturbations. Table 8 presents the defences in this category, which attacks the defence counters, the impact on performance, and the framework used by the paper that presented the defence.

Robust Learning [25, 106] increases the robustness of an agent in an environment. These methods do not consider specific attacks and instead consider robustness against specific adversary capabilities, such as the magnitude of perturbation. Some of these approaches are also able to certify the robustness of agents' decisions against a certain level of adversarial capability [25, 140]. Table 9 shows robust learning techniques and their impact on performance.

A limitation of our work is that our data collection only found papers that considered the improved robustness in the context of defending against an AML attack. We believe that the robust learning category could be extended to consider approaches that do not consider specific AML adversary capabilities, but instead look at generally improving the robustness of an agent to other conditions, such as robustness against non-stationary environments [70].

Input Alteration [115] mitigates an AML attack by removing the adversarial component of the observation or communication. Table 10 shows the input alteration defences, their requirement for adversarial detection, which attacks they counter, their impact on performance, and the framework used by the paper that presented the defence.

Methods that do not require the detection of adversarial inputs alter the observation based on assumptions about an adversary's capabilities, such as the magnitude of perturbation. These defences have been considered in supervised learning contexts [37] and we found three Input Alteration defences for DRL [34, 40, 89]. These methods risk removing valuable data, causing a decrease in the performance of an agent, as shown by the observation masking defence [34]. However, using an auto-encoder did improve the performance over the original agent [89].

Table 8. Competitive Training Defences

Name of Defence	Countered Attack Vectors	Impact on Performance	Framework
RARL [99]	Malicious Actions	Positive	SG
PR Operators [127]	Action Perturbations	Not evaluated	PR-MDP
NR Operators [127]	Action Perturbations	Not evaluated	NR-MDP
Adversary Resistance [39]	Malicious Actions	Not evaluated	SG
Adversarial training [93]	Malicious Actions	Positive	MDP
ARTS [98]	Malicious Actions	Negative	POSG
MACRL [144]	Communication Perturbations	Not evaluated	Dec-POMDP-C
Up-training [85]	Malicious Actions	Positive	MDP
Adversarial Threat Mitigation [107]	Malicious Actions	Positive	Dec-POMDP
ATLA [154]	Observation Perturbation	Mixed	SA-MDP
WB-RARL [15]	Malicious Actions	Negative	2-player MDP
RAD-CHT [6]	Observation Perturbation	Negative	MDP with Adversarial Observations
PRIM [74]	Action Perturbations	Negative	Stochastic Game
ROMANCE [153]	Action Perturbations	Negative	LPA-Dec-POMDP

Table 9. Robust Learning Defences

Name of Defence	Impact on Performance
A2PD [106]	Positive
CARRL [25]	Not evaluated
CROP [140]	Not evaluated
CPPO [149]	Positive
TRC [55]	Positive
PATROL [41]	Positive
ReCePS [87]	Positive

Methods that detect adversarial inputs are able to more precisely blind the agent to those inputs or sanitise the observation to remove the malicious components. Message filtering was used against both Perturbed [144, 151] and Malicious Communications[83]. However, attacks have been shown to avoid detection [13] in a supervised learning context. We believe that this evasion of detection has yet to be considered in the context of RL.

Adversarial Detection [137] identifies either the existence or location of adversarial attacks. Tekgul et al. [126] considers adversarial detection, but does not consider how to remove or repair adversarial observations. CIQ [145] uses causal inference to identify an adversarial attack. AdvEx-RL [109] and TAS-RL [108] learn to detect the effects of an adversarial attack by a particular adversary. Decentralized Anomaly Detection [54] detects adversarial attacks against MARL-trained systems. INRD [64] uses changes in the neural network to detect adversarial attacks. Finally, LSP memory [156] aims to detect Malicious Action attacks in MAS by using a memory-based anomaly detection method.

Table 10. Input Alteration Defences

Name of Defence	Adversarial Detection Required	Countered Attack Vectors	Impact on Performance	Framework
DQWAE [89]	×	Observation Perturbation	Positive	MDP
Blinding [40]	×	Malicious Actions	Not evaluated	MDP
Message Filtering [83]	✓	Malicious Communications	Not evaluated	Dec-POMDP
Instance-based Defense [32]	✓	Observation Perturbation	Not evaluated	MDP
Message filter [144]	✓	Communication Perturbations	Not evaluated	Dec-POMDP-C
Observation masking [34]	×	Malicious Actions	Negative	Stochastic Game
Distance-based Defense [134]	×	Observation Perturbation	Negative	MDP
Policy Smoothing [66]	×	Observation Perturbation	Negative	Adversarial robustness
ADMAC [151]	✓	Communication Perturbation	Mixed	Dec-POMDP-C

Regularisation [36] is a technique commonly used in DRL to improve the training and generalisation of models. While its primary purpose is to reduce overfitting [16], regularisation has also been explored as a defence mechanism against AML attacks in DRL. Regularisation introduces additional terms to the loss function to improve the robustness of a DRL algorithm. We share regularisation defences in Table 11 and include which attack vectors the defence counteres, the impact on performance, and the framework used in the paper.
Memory is crucial in partially-observable environments and may also be used to defend against AML attack [112, 145, 156]. Deep Recurrent Q-Networks [46] have been developed, incorporating a Long Short-Term Memory (LSTM) component, enabling the retention of latent state representations. This approach has shown promise as a potential defence against AML [112]. CIQ [145] uses causal inference to alter its behaviour to defend against an attack. The algorithm uses labelled data to learn to predict if the current observation has been adversarially perturbed by considering the previous sequence of observations. CIQ then uses a secondary neural network that has been adversarially-trained to make a decision with the adversarially perturbed observation. We believe that further research is required into the potential risks of these defences, as they may themselves be vulnerable to attack. It also opens a new vulnerability, which is a single adversarial perturbation affecting multiple future steps.
Ensembles [124] train multiple algorithms which collectively vote for an action for the agent to take. We found four ensemble-based defences [108, 109, 121, 145, 146]. AME [121] defends against a Communication Perturbation attack in a mixed multi-agent environment. RORL [146] uses an ensemble of Q-functions to mitigate Observation Perturbation attacks. TAS-RL [108] and AdvEx-RL [109] both train an additional agent that selects safe actions, and executes those safe actions if they detect the effects of an adversarial attack.

7 Proposed Framework for Attack Vector Modelling

Existing works have used a variety of frameworks for modelling AML attacks against DRL. However, many of these frameworks are unsuitable for modelling attack vectors. Single agent and cooperative multi-agent

Table 11. Regularisation Defences

Name of Defence	Countered Attack Vectors	Impact on Performance	Framework
Lipschitz Constant [125]	Malicious Actions, Observation Perturbations	Not Evaluated	Semi Markov Decision Process
Worst-case-aware Robust RL [75]	Action Perturbations, Observation Perturbations	Mixed	MDP
MAGI [23]	Communication Perturbations	Positive	Dec-POMDP
MF-GPC [33]	Observation Perturbations	Positive	dynamical system with adversarial cost functions
RAD [6]	Observation Perturbations	Negative	MDP with Adversarial Observations
Sensitivity-Aware Regularizer [148]	Observation Perturbations	Not evaluated	Goal-Conditioned MDP
ERNIE [11]	Malicious Actions, Observation Perturbations	Positive	DEC-POMDP
RORL [146]	Observation Perturbations	Positive	MDP
SBPR [74]	Action Perturbations	Negative	Stochastic Game

frameworks, namely, MDPs, POMDPs, and DEC-POMDPs, are commonly used in the papers covered by this survey and are unable to formally model any of the attack vectors. A minority of papers used frameworks capable of modelling some of the attack vectors, such as the POSG, SA-MDP [155], PR-MDP [127], and the NR-MDP [127]. However, to the best of our knowledge, there currently exists no framework that allows the modelling of all attack vectors simultaneously. To address this gap, we propose a framework that extends and combines existing frameworks, that we call the Adversarial POSG (APOSG) (Definition 8). APOSG is capable of modelling all attack vectors in mixed multi-agent environments. This ability to model multiple attack vectors allows researchers to compare similar attacks, such as the timing methods used in Observation and Action Perturbation attacks, as well as consider the effect of an attack using multiple attack vectors, such as a low magnitude random Action Perturbation and a reward-targeted Malicious Communication.

Continuing our medical treatment example, the APOSG would allow us to model a number of specialists with diverse motivations, the presence of an adversary that may alter the data being obtained by the specialists and alter the medication being taken by the patient, including when the adversary affects the data and medication and how much the adversary can change. To conclude this macabre example, an adversary seeking the death of the patient may reduce the medication the patient received from altruistically motivated specialists and provide slightly incorrect data to profit-motivated specialists that causes a misdiagnosis and eventually death.

APOSG (Definition 8) features an action-robust operator v_i^a that perturbs the action of an agent i. This allows the action of each agent in the environment to be subject to different perturbations. The action-robust operator is conditioned by Θ_i^a and Δ_i^a. Θ_i^a is the action attack tempo and determines whether action a_i is perturbed at state s. Δ_i^a sets a maximum magnitude on the perturbation. We use the term tempo to capture the similarity between AML attacks and cyber attacks [73]. Broadly, the tempo of a cyber-attack describes how often an attacker takes an action, which varies based on a number of factors in the environmental state and has a significant impact on the success of an attack. Similarly, in Section 5.2, we discuss the time aspect of an AML attack against MARL

and DRL, and its impact on the success of an AML attack. To model the Observation Perturbation attack vector in multi-agent and partially observable environments, the APOSG uses an observation-adversarial function, v_i^o, for each agent i in the environment. We introduce the parameter Δ_i^o to model the magnitude of change that an adversary may inflict on an observation. The observation attack tempo function Θ^o, in the APOSG, determines if a particular state will be attacked and captures the concept of when an attack will occur.

Definition 8. *An Adversarial Partially Observable Stochastic Game (APOSG) is defined by the 13-tuple* $\{I, S, \hat{A}, T, \hat{\Omega}, \hat{O}, \hat{R}, \hat{\Theta^a}, \hat{v^a}, \hat{\Delta^a}, \hat{\Theta^o}, \hat{v^o}, \hat{\Delta^o}\}$*, in which I, S, Â, T, Ω̂, Ô, and R̂ are defined the same as Definition 4, and*

- $\hat{\Theta^a}$ *is the set of action attack tempos for all agents, such that* $\Theta_i^a \in \hat{\Theta^a}$ *determines the probability that the action of the agent i for a given state* s_t *will be perturbed by the function* v_i^a*,*
- $\hat{v_i^a}$ *is the set of action-robust operators, such that* $v_i^a \in \hat{v^a}$ *perturbs the action selected by agent i with probability* $\Theta_i^a(s_t)$*.*
- $\hat{\Delta^a}$ *is the set of action attack magnitudes, where* $\Delta_i^a \in \hat{\Delta^a}$ *is the maximum magnitude of change from the perturbed action* a'_{it} *to the original action* a_{it}*, such that* $|a'_{it} - a_{it}| \leq \Delta_i^a$
- $\hat{\Theta^o}$ *is the set of observation attack tempos for all agents, such that* $\Theta_i^o \in \hat{\Theta^o}$ *determines the probability that the observation of the agent i for a given state* s_t *will be perturbed by the function* v_i^o*,*
- $\hat{v^o}$ *is the set of adversarial observation-permutation functions, such that* $v_i^o \in \hat{v^o}$ *perturbs the observation of agent i with probability* $\Theta_i^o(s_t)$*.*
- $\hat{\Delta^o}$ *is the set of observation attack magnitudes, where* $\Delta_i^o \in \hat{\Delta^o}$ *is the maximum magnitude of change of the perturbed observation* o'_{it} *from the original observation* o_{it} *of the agent i, such that* $|o'_{it} - o_{it}| \leq \Delta_i^o$

We can demonstrate the applicability of APOSG on existing attacks. PAP and NAP [127] are single agent environments, thus $I = [0]$ and we omit subscript 0. The state, state-transition, and reward functions come from the particular environment. The attacks target fully observable environments, thus $\Omega = S$ and $O(s_t, a_{t-1}) = s_t$. PAP completely replaces the original action, thus for PAP: $\Delta^a = \infty$, whereas NAP perturbs the action by a certain amount called α in the original paper, thus for NAP $\Delta^a = \alpha$. Both PAP and NAP train an adversarial policy $\hat{\pi}$ that selects the worst case action $a' \in A$, thus for PAP: $v^a(o_t, a_t) = a'_t$ and for NAP: $v^a(o_t, a_t) = (1 - \alpha)a_t + \alpha a'_t$. PAP probabilistically replaces an action, thus for PAP: $\Theta^a(s) = p$, and NAP replaces every timestep, thus NAP: $\Theta^a(s_t) = 1$. PAP and NAP only perform an Action Perturbation attack and so $\Theta^o(s_t) = 0$, and $\hat{v^o}$, and $\hat{\Delta^o}$ are not used.

APOSG can also model multi-agent AML attacks. We demonstrate this by modelling the Adversarial Communication [144] attack. The multi-agent environment features n cooperative agents, thus $I = [0, ..., n-1]$. The state, state-transition, and reward functions come from the particular environment. The attacks target partially observable environments, thus, Ω and O come from the particular environment. The action space can be separated into environmental and message actions, thus $A = A_e \times A_m$, where A_e are the environmental actions that affect the state transitions of the environment and A_m are message actions that send a message from one agent to another. Adversarial Communication perturbs the message component of the action, thus $\Delta^a = \infty$. Adversarial Communication only affects a limited number of agents in the system. We can let I_{adv} represent those agents being attacked. Adversarial Communication trains an adversarial policy $\hat{\pi}$ that outputs a perturbed message but does not otherwise affect the environmental action selection, thus: $v_i^a(o_{it}, \{a_{it}^e, a_{it}^m\}) = \{a_{it}^e, a_{it}'^m\}$, where

$$a_{it}'^m = \begin{cases} \hat{\pi}(o_{it}) & if i \in I_{adv} \\ a_{it}^m & otherwise \end{cases}.$$

Adversarial Communication perturbs a message at every time-step, thus $\Theta_i^o(s_t) = 1 \forall i \in I$.

We demonstrate the applicability of APOSG for modelling existing attacks with the Uniform attack [49]. This attack targets single agent environments, thus $I = [0]$ and we omit subscript 0. The state, state-transition, and

reward functions come from the particular environment. The attack targets fully observable environments, thus $\Omega = S$ and $O(s_t, a_{t-1}) = s_t$. The Uniform attack considers various magnitudes, however a magnitude of 0.001 was found to decrease the target's performance by at least 50%. The Uniform attack uses FGSM to craft an adversarial policy, thus $v^o(o_t) = o_t + \Delta^o FGSM(o_t)$, where $FGSM = sign(\nabla_o J(\theta, o_t, softmax(Q_\theta(o_t))))$ for a victim DQN, such that θ are the parameters of the DQN, and $Q_\theta(o_t)$ is the predicted Q-values of the DQN. The Uniform attack attacks every timestep, thus $\Theta^o(s_t) = 1$.

We can also demonstrate the applicability of APOSG for modelling the CBAP attack [157]. Like the Uniform attack, this attack targets single agent environments, thus $I = [0]$, S, T, and R_0 come from the particular environment. The attack targets fully observable environments, thus $\Omega = S$ and $O(s_t, a_{t-1}) = s_t$. The CBAP attack uses $\Delta^o = 0.1$. The CBAP attack also uses FGSM, however it uses a one-hot encoding of the highest Q-value action instead of the softmax of the Q-values, thus $FGSM = sign(\nabla_o J(\theta, o_t, OneHotEncode(argmax(Q_\theta(o_t)))))$. The CBAP attack targets states where the difference between an agent's preferences for its most preferred and second-most preferred action exceed a threshold β, thus

$$\Theta^o(o_t) = \begin{cases} 1 & if\, p(o_t) > \beta \\ 0 & otherwise \end{cases},$$

where $p(o_t) = max_{a \in A} \pi(o_t|a) - max_{a \in A'} \pi(o_t|a)$, where A' is the action set A excluding the highest valued action $argmax_{a \in A} \pi(o_t|a)$.

Through modelling both the Uniform [49] and CBAP [157] attacks, we can observe some key similarities and differences between the attacks. Both attacks are untargeted and use FGSM to craft the perturbations, however there are slight differences to their approaches to using FGSM. The Uniform attack targets all timesteps, whereas the CBAP attack only targets some of the timesteps as determined by the Θ^o equation. We can use the APOSG to isolate and compare these differences.

We can also use APOSG to model Malicious Action attacks, and we demonstrate this by modelling the attack by Gleave et al. [34]. The multi-agent environments feature two agents, thus $I = [0, 1]$. The state, state-transition, and reward functions come from the particular environment. The attacks target partially observable environments, thus, Ω and O come from the particular environment. The Malicious Action attack does not perturb the observation, thus $\Delta^a = 0$ and $\Delta^o = 0$. Instead, the attack is modelled in the state transition function, reward function, and the observation probability function. All of these functions depend on the joint action selected by both agents. By using this model, we can see that Malicious Action attacks have three effects on the victim. First, they directly affect the reward. Secondly, they affect the state transition. Finally, they affect the observation received by the victim. For the observation component of the attack, the victim's observation $o_{0t} = O(s_t, a_{0t}, a_{1t})$, thus the attacker agent, selects the action a_{1t} which causes o_{0t} causing an adversarial effect on the victim.

APOSG can model some of the defences discussed in Section 6. Specifically, APOSG can model memory, input alteration, ensembles, and, to a limited extent, adversarial detection. Memory-based defences [112, 145, 156] may be modelled by extending the state to include agents' memories allowing investigation into the robustness of memory defences to state-based objective attacks such as EA [77]. The APOSG may model input alteration defences [34, 40, 66, 89, 134] using the adversarial observation-permutation functions. Formalising the observation change used in these defence would allow us to quantify information loss and explore the impact of such defences on the base performance of a system. Ensemble defences [108, 109, 121, 146] may be modelled as different agents in an APOSG, where the transition function to combine the joint-action of the ensemble. This approach would allow us to explore the robustness of different ensemble populations and voting mechanisms have to AML attacks. Adversarial detection defences that work with input alterations [32, 83, 144, 151] and ensembles [108, 109, 145] may be modelled by extending the observation-permutation functions or agent policies respectively. However, standalone adversarial detection defences [54, 64, 126, 156], that rely on human intervention post-detection are

not easily modelled by the APOSG. APOSG is also unable to model defences that occur during training, namely, Adversarial Training, Competitive Training, Robust Learning, and Regularisation. Despite these limitations, modelling some defences allows for a more robust comparison of those defences and opens new avenue of research.

8 Research Gaps and Challenges

We focus on new AML attacks that may be effective against MARL algorithms, and suggest approaches to AML defences that may improve algorithm robustness. We have identified potential AML attacks that are yet to be explored, namely, targeting multiple agents and targeting multiple attack vectors. We have also found a gap in the quantification of AML defences. Finally, we suggest a novel defence that could use knowledge of an attacker.

8.1 Attacks against Multiple Agents

A major research direction that is yet to be explored in literature is AML attacks against multiple agents. The development and understanding of these attacks allow us to better understand the vulnerability and risks of using MARL. MARL uses both explicit [27] and implicit communications [42] to coordinate the agent behaviours, and we believe that a cascading effect on the system may be produced by attacking a single agent. This failure induced by a single agent's actions is shown by works that present Malicious Communications [131, 144] and Malicious Action [30, 34, 39, 93, 98, 132] attacks. Certain individuals in social networks are able to unduly influence the behaviour of the whole network [147]. Likewise, an attack against an individual agent may disproportionately affect the performance of the whole system. An attack may be more effective based on factors, such as timing and communication protocols. The effects of these attacks may also cascade through time, causing delayed adversarial effects or an accumulating vulnerability. Measuring this impact is a very important research problem that could be used to improve robustness of MAS against attacks.

The influence of a single agent in a MAS changes over the course of an episode. We believe that investigating the effect of switching the targeted agent of an attack is an important research direction. Findings may be used to increase the effectiveness of attacks or demonstrate flaws in AML defences that assume all agents are being attacked [83, 144]. Attacks such as the Strategically Timed Attack [77] and the Critical Point Attack [120] consider the importance of an action at a particular time step during a game. Extending these to consider the agent importance at a particular time step would allow for more precise attacks against MAS.

MAS may be *homogeneous*, in which all agents share an identical policy, or *heterogeneous*, in which agents have different policies. The relative vulnerability of heterogeneous vs homogeneous systems has yet to be explored. We hypothesise that homogeneous systems are vulnerable based on the increased implicit communication [42] and the similarity of the agents leading to a higher risk of successful transfer attacks.

Attacks that use reward-based targeting rely on the target agent attempting to optimise a single reward. A single reward is often used in cooperative MARL for all agents. However, mixed MARL can produce cooperation and coordination due to the use of certain reward functions [53]. We believe that it would be valuable to compare the vulnerability of MARL algorithms that use shared rewards to those that use individual rewards, such as COMA [28].

8.2 Attack Vectors

The current state of AML attacks only target a single attack vector and there is yet to be an exploration of an AML attack that targets multiple attack vectors. The proposed APOSG framework allows research into combining different attack vectors that may produce more effective attacks than those that use a single attack vector. For example, allowing an adversarial agent to both play the opponent and discover Malicious Actions while simultaneously perturbing inter-agent communications could be much more effective than either of those

attacks individually. Combining attack vectors may also reveal vulnerabilities in real-world systems. An example of this is an adversary capable of introducing minor perturbations into the continuous steering actions of a self-driving vehicle through a cyber attack and a physical patch attack to degrade the performance of the vehicle to a catastrophic degree, where a single attack is unable to achieve any meaningful effect on the vehicle. This hypothetical effect would only be achievable because the combined attack was able to shift the vehicle steering action beyond the robustness threshold of the victim agent.

The study of the trade-off between the effectiveness of direct attack vectors vs the practicality of indirect attack vectors is critical. We hypothesise that the most effective attacks are those with direct input to the victim agents, namely, Observation and Communication Perturbations and Malicious Communications. These perturbations also benefit from the perturbed observations being outside of the victim's training set. Indirect attacks, namely action perturbations and malicious actions, may be less effective because they can affect the state of the environment, which affects the observation of the victim. The ability to manipulate the state may be limited by factors including the actions of other agents in the environment. Even then, it is possible that the victim has already trained against that observation and does not experience an adversarial effect. However, indirect attacks are easier to achieve than direct attacks in a real-world setting.

The enumeration of attack vectors has revealed potential gaps in AML attacks against MARL, namely, white-box Malicious Communications, patch attacks, action-based and state-based targeted Communication Perturbation attacks, untargeted Communication Perturbation attacks, action-based and state-based targeted Malicious Actions. Malicious Communications allow an adversary to send any data to a target which we believe would be devastating, as shown by the single-pixel attack in the supervised learning domain [118]. Similarly, the single-pixel attack could be investigated in the Patch Attack vector. Communication Perturbation has only been considered with reward-based targeting. However, no research has been done to our knowledge into targeting actions and states. Further, untargeted Communication Perturbation attacks have not been investigated. Malicious Actions have been used to target rewards, but no research has been done to see if Malicious Actions can manipulate a target into taking certain actions or moving towards certain states. Being able to convince an adversary to move towards specific states, or take certain actions, could have a significant impact on the field of competitive MARL.

8.3 Quantifying Defence Generalisations

There is a gap in the current approaches to measuring the effectiveness of a defence, specifically, many defences are only evaluated against a single specific attack type. Our survey has shown that many AML defences are only evaluated against the attack they were designed to mitigate. However, we have also shown that there exist a broad range of attacks against DRL trained agents. AML defences must be able to be deployed to protect these agents, and it is vitally important that the impact of AML defences be well understood before deployment otherwise there is a risk of increasing the overall system vulnerability. Evaluating AML defences against multiple attacks from a variety of attack vectors would provide an insight into practical AML mitigations. This evaluation would allow DRL practitioners to better understand if a defence is effective against multiple attacks, only effective against a single attack, or if it introduces new vulnerabilities into the system.

Input alteration defences may remove vital information while attempting to remove adversarial examples [34]. This defence may cause a significant loss in the performance of the system. An ideal defence would selectively sanitise an observation to remove adversarial aspects of the observation but retain other important and unattacked information. However, defences deployed during an attack are also vulnerable. Evasion of Adversarial Detection techniques has been demonstrated in a supervised learning context [13], although this is yet to be explored for DRL. To mitigate this potential attack, other methods such as adversarial training in conjunction with input alteration defences may provide better security than either option alone.

Adversarial training has been shown to reduce the performance of the model to benign examples [59], but it is yet to be explored if adversarial training increases the vulnerability of the system to other attacks such as malicious actions. Malicious actions indirectly attack the system using observations generated by the environment. If adversarial training does cause some degree of forgetting in the model, then we expect the vulnerability of the model to malicious actions to increase. This vulnerability may be mitigated by using other defences, such as competitive training which has been shown to improve the robustness of a model to malicious actions [99]. Combining these two training techniques, while maintaining effective benign performance may be a significant challenge. However, overcoming that challenge may lead to a more robust agent.

Defences against Action Perturbation attacks have the potential to address both the mitigation of Action Perturbation attacks and to explore resilience in the face of attacks from other vectors. That is, if a bad action occurs, then the agent should learn how to best recover. This is a vital property for any system that may be operating in the real world. However, we found a single defence against Action Perturbation attacks [127].

8.4 Employing Attacker Metrics in Defence

From our survey, we observe that there are yet to be AML defences that use information about an attacker. In cyber-security, intelligence about the adversary is vital to crafting an appropriate response. To this end, we believe that both identifying the presence of an attack and the properties of the attacker such as preferred attack vectors or tempo of attacks may produce more effective defences. We identified AML defences that identify the presence of an attack and even isolate the location of an attack [32, 54, 64, 83, 108, 109, 126, 144, 145, 151]. However, none of these made use of other attacker information, such as when an attack may occur.

AML attacks against MARL and DRL rely on timing [49, 67, 77, 104, 120]. Timing methods may use knowledge about the victim's action preference, which is a difficult task with black-box information. However, a defender has direct access to this information. Despite this, there are currently no defences that use knowledge about when an attack may occur. An Adversarial Detection defence may be extended to make use of information about the likelihood of an attack, based on the action preference of the agent it is defending.

9 Conclusion

There are significant gaps in the research around Adversarial Machine Learning (AML) attacks and defences for Multi-Agent Reinforcement Learning (MARL) that need to be addressed, including mitigating AML attacks against multiple agents, the combined effect of multiple AML attacks, quantifying the effectiveness of AML defences, and using knowledge about an attacker to improve AML defences. Our survey has identified numerous AML attacks and defences for single-agent Deep Reinforcement Learning (DRL) algorithms. However, we found relatively few attacks and defences for MARL. Beyond the much-needed systematisation of knowledge, a key contribution of this work is the perspective of attack vectors used to understand attacks by capturing the means by which an adversary might execute an attack, i.e., action, observation, communication perturbations, malicious communications, and malicious actions. Another key contribution of this work is the proposal of a new framework, Adversarial Partially Observable Stochastic Game (APOSG), which addresses a gap in the current frameworks being used to research AML in DRL, namely, the inability to model Communication Perturbations, the inability to model multiple simultaneous AML attack vectors, and the lack of detail around the tempo and magnitude of attacks. We identify the need for future work to study attacks against multiple agents, and on quantifying the effectiveness of defences through the use of various metrics.

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. 2017. A Brief Survey of Deep Reinforcement Learning. *IEEE Signal Processing Magazine* 34, 6 (Nov. 2017), 26–38.

[2] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. 2021. Recent Advances in Adversarial Training for Adversarial Robustness. In *International Joint Conference on Artificial Intelligence*. 4312–4321.
[3] Shumeet Baluja and Ian Fischer. 2018. Learning to Attack: Adversarial Transformation Networks. In *AAAI Conference on Artificial Intelligence*, Vol. 32. 2687–2695.
[4] Vahid Behzadan and Arslan Munir. 2017. Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks. In *Machine Learning and Data Mining in Pattern Recognition*. 262–275.
[5] Vahid Behzadan and Arslan Munir. 2017. Whatever Does Not Kill Deep Reinforcement Learning, Makes It Stronger.
[6] Roman Belaire, Pradeep Varakantham, Thanh Nguyen, and David Lo. 2024. Regret-based Defense in Adversarial Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. 2633–2640.
[7] Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. 2017. Exploring the Space of Black-box Attacks on Deep Neural Networks.
[8] Jan Blumenkamp and Amanda Prorok. 2021. The Emergence of Adversarial Communication in Multi-Agent Reinforcement Learning. In *Conference on Robot Learning*. 1394–1414.
[9] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial Patch.
[10] Prasanth Buddareddygari, Travis Zhang, Yezhou Yang, and Yi Ren. 2022. Targeted Attack on Deep RL-based Autonomous Driving with Learned Visual Patterns. In *International Conference on Robotics and Automation*. 10571–10577.
[11] A. Bukharin, Y. Li, Y. Yu, Q. Zhang, Z. Chen, S. Zuo, C. Zhang, S. Zhang, and T. Zhao. 2023. Robust Multi-Agent Reinforcement Learning via Adversarial Regularization: Theoretical Foundation and Stable Algorithms. In *Advances in Neural Information Processing Systems*, Vol. 36. 13 pages.
[12] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (March 2008), 156–172.
[13] Nicholas Carlini and David Wagner. 2017. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *ACM Workshop on Artificial Intelligence and Security*. 3–14.
[14] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy*. 39–57.
[15] Stephen Casper, Taylor Killian, Gabriel Kreiman, and Dylan Hadfield-Menell. 2022. White-Box Adversarial Policies in Deep Reinforcement Learning.
[16] Gavin C Cawley and Nicola LC Talbot. 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research* 11 (2010), 2079–2107.
[17] Patrick PK Chan, Yaxuan Wang, and Daniel S Yeung. 2020. Adversarial Attack against Deep Reinforcement Learning with Static Reward Impact Map. In *ACM Asia Conference on Computer and Communications Security*. 334–343.
[18] Tong Chen, Jiqiang Liu, Yingxiao Xiang, Wenjia Niu, Endong Tong, and Zhen Han. 2019. Adversarial attack and defense in reinforcement learning-from AI security view. *Cybersecurity* 2, 1 (March 2019), 1–22.
[19] Yize Chen, Daniel Arnold, Yuanyuan Shi, and Sean Peisert. 2021. Understanding the Safety Requirements for Learning-based Power Systems Operations.
[20] Ricson Cheng, Ziyan Wang, and Katerina Fragkiadaki. 2018. Geometry-aware recurrent neural networks for active visual recognition. In *Advances in Neural Information Processing Systems*, Vol. 31. 11 pages.
[21] Z. Cheng, X. Wu, J. Yu, W. Sun, W. Guo, and X. Xing. 2023. StateMask: Explaining Deep Reinforcement Learning through State Mask. In *Advances in Neural Information Processing Systems*, Vol. 36. 31 pages.
[22] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. 2019. TarMAC: Targeted multi-agent communication. In *International Conference on Machine Learning*, Vol. 2019-June. 1538–1546.
[23] S. Ding, W. Du, L. Ding, L. Guo, and J. Zhang. 2024. Learning Efficient and Robust Multi-Agent Communication via Graph Information Bottleneck. In *AAAI Conference on Artificial Intelligence*, Vol. 38. 17346–17353. https://doi.org/10.1609/aaai.v38i16.29682
[24] Oliver Eigner, Sebastian Eresheim, Peter Kieseberg, Lukas Daniel Klausner, Martin Pirker, Torsten Priebe, Simon Tjoa, Fiammetta Marulli, and Francesco Mercaldo. 2021. Towards Resilient Artificial Intelligence: Survey and Research Issues. In *IEEE International Conference on Cyber Security and Resilience*. 536–542.
[25] Michael Everett, Björn Lütjens, and Jonathan P How. 2021. Certifiable Robustness to Adversarial State Uncertainty in Deep Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* 33, 9 (2021), 4184 – 4198.
[26] Aidin Ferdowsi, Ursula Challita, Walid Saad, and Narayan B Mandayam. 2018. Robust Deep Reinforcement Learning for Security and Safety in Autonomous Vehicle Systems. In *International Conference on Intelligent Transportation Systems*. 307–312.
[27] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. 2016. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*. 2145–2153.
[28] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, Vol. 32. 2974–2982.
[29] Joachim Folz, Sebastian Palacio, Joern Hees, and Andreas Dengel. 2020. Adversarial Defense based on Structure-to-Signal Autoencoders. In *IEEE Winter Conference on Applications of Computer Vision*. 3568–3577.

[30] Ted Fujimoto and Arthur Paul Pedersen. 2022. Adversarial Attacks in Cooperative AI.
[31] Javier García, Rubén Majadas, and Fernando Fernández. 2020. Learning adversarial attack policies through multi-objective reinforcement learning. *Engineering Applications of Artificial Intelligence* 96, Article 104021 (Nov. 2020), 11 pages.
[32] Javier García and Ismael Sagredo. 2022. Instance-based defense against adversarial attacks in Deep Reinforcement Learning. *Engineering Applications of Artificial Intelligence* 107 (Jan. 2022), 104514.
[33] U. Ghai, A. Gupta, W. Xia, K. Singh, and E. Hazan. 2023. Online Nonstochastic Model-Free Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 36. 27 pages.
[34] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2020. Adversarial Policies: Attacking Deep Reinforcement Learning. In *International Conference on Learning Representations*. 16 pages.
[35] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*. 11 pages.
[36] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. 2021. Regularisation of Neural Networks by Enforcing Lipschitz Continuity. *Machine Learning* 110 (2021), 393–416.
[37] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. 2018. Countering Adversarial Images using Input Transformations. In *International Conference on Learning Representations*. 12 pages.
[38] Jun Guo, Yonghong Chen, Yihang Hao, Zixin Yin, Yin Yu, and Simin Li. 2022. Towards Comprehensive Testing on the Robustness of Cooperative Multi-Agent Reinforcement Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 115–122.
[39] Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing. 2021. Adversarial Policy Learning in Two-player Competitive Games. In *International Conference on Machine Learning*. 3910–3919.
[40] Wenbo Guo, Xian Wu, Usmann Khan, and Xinyu Xing. 2021. EDGE: Explaining Deep Reinforcement Learning Policies. In *Advances in Neural Information Processing Systems*, Vol. 34. 12222–12236.
[41] W. Guo, X. Wu, L. Wang, X. Xing, and D. Song. 2023. PATROL: Provable Defense against Adversarial Policy in Two-player Games. In *USENIX Security Symposium*, Vol. 6. 3943–3960.
[42] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, Vol. 10642 LNAI. 66–83.
[43] Songyang Han, Sanbao Su, Sihong He, Shuo Han, Haizhao Yang, and Fei Miao. 2024. What is the Solution for State-Adversarial Multi-Agent Reinforcement Learning? *Transactions on Machine Learning Research* 2024 (2024), 37 pages.
[44] Yi Han, David Hubczenko, Paul Montague, Olivier De Vel, Tamas Abraham, Benjamin I. P. Rubinstein, Christopher Leckie, Tansu Alpcan, and Sarah Erfani. 2020. Adversarial Reinforcement Learning under Partial Observability in Autonomous Computer Network Defence. In *International Joint Conference on Neural Networks*. 1–8.
[45] Eric Hansen, Daniel Bernstein, and Shlomo Zilberstein. 2004. Dynamic Programming for Partially Observable Stochastic Games. In *National Conference on Artifical Intelligence*. 709—-715.
[46] Matthew Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. In *AAAI Fall Symposium Series*. 29–37.
[47] Matthew John Hausknecht. 2016. *Cooperation and communication in multiagent deep reinforcement learning*. Ph. D. Dissertation. The University of Texas.
[48] Mengdi Huai, Jianhui Sun, Renqin Cai, Liuyi Yao, and Aidong Zhang. 2020. Malicious Attacks against Deep Reinforcement Learning Interpretations. In *KDD*. 472–482.
[49] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial Attacks on Neural Network Policies. In *International Conference on Learning Representations*. 7 pages.
[50] Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. 2020. CopyCAT: Taking Control of Neural Policies with Constant Attacks. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 2020-May. 548–556.
[51] Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. 2021. Challenges and Countermeasures for Adversarial Attacks on Deep Reinforcement Learning. *IEEE Transactions on Artificial Intelligence* 3, 2 (Sept. 2021), 90–109.
[52] Matthew Inkawhich, Yiran Chen, and Hai Li. 2020. Snooping attacks on deep reinforcement learning. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 2020-May. 557–565.
[53] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A. Ortega, D. J. Strouse, Joel Z. Leibo, and Nando de Freitas. 2019. Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning. In *International Conference on Machine Learning*, Vol. 2019-June. 5372–5381.
[54] K. Kazari, E. Shereen, and G. Dán. 2023. Decentralized Anomaly Detection in Cooperative Multi-Agent Reinforcement Learning. In *International Joint Conference on Artificial Intelligence*, Vol. 2023-August. 162–170.
[55] Dohyeong Kim and Songhwai Oh. 2022. TRC: Trust Region Conditional Value at Risk for Safe Reinforcement Learning. *IEEE Robotics and Automation Letters* 7, 2 (April 2022), 2621–2628.

[56] Panagiota Kiourti, Kacper Wardega, Susmit Jha, and Wenchao Li. 2020. TrojDRL: Evaluation of Backdoor Attacks on Deep Reinforcement Learning. In *ACM/IEEE Design Automation Conference*. 1–6.
[57] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology* 51, 1 (Jan. 2009), 7–15.
[58] Ezgi Korkmaz. 2020. Daylight: Assessing Generalization Skills of Deep Reinforcement Learning Agents.
[59] Ezgi Korkmaz. 2021. Adversarial Training Blocks Generalization in Neural Policies. In *NeurIPS Workshop on Distribution Shifts: Connecting Methods and Applications*. 6 pages.
[60] Ezgi Korkmaz. 2021. Assessing Deep Reinforcement Learning Policies via Natural Corruptions at the Edge of Imperceptibility.
[61] Ezgi Korkmaz. 2021. Investigating vulnerabilities of deep neural policies. In *Conference on Uncertainty in Artificial Intelligence*. 1661–1670.
[62] Ezgi Korkmaz. 2021. Non-Robust Feature Mapping in Deep Reinforcement Learning. In *Workshop on Adversarial Machine Learning*. 6 pages.
[63] E. Korkmaz. 2023. Adversarial Robust Deep Reinforcement Learning Requires Redefining Robustness. In *AAAI Conference on Artificial Intelligence*, Vol. 37. 8369–8377.
[64] E. Korkmaz and J. Brown-Cohen. 2023. Detecting Adversarial Directions in Deep Reinforcement Learning to Make Robust Decisions. In *International Conference on Machine Learning*, Vol. 202. 17534–17543.
[65] Jernej Kos and Dawn Song. 2017. Delving into adversarial attacks on deep policies. In *International Conference on Learning Representations*. 6 pages.
[66] Aounon Kumar, Alexander Levine, and Soheil Feizi. 2022. Policy Smoothing for Provably Robust Reinforcement Learning. In *International Conference on Learning Representations*. 29 pages.
[67] R Praveen Kumar, I Niranjan Kumar, Sujith Sivasankaran, A Mohan Vamsi, and Vineeth Vijayaraghavan. 2021. Critical State Detection for Adversarial Attacks in Deep Reinforcement Learning. In *IEEE International Conference on Machine Learning and Applications*. 1761–1766.
[68] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial Machine Learning at Scale. In *International Conference on Learning Representations*. 17 pages.
[69] Cassidy Laidlaw and Soheil Feizi. 2019. Functional Adversarial Attacks. In *Advances in Neural Information Processing Systems*. 11 pages.
[70] Erwan Lecarpentier and Emmanuel Rachelson. 2019. Non-stationary markov decision processes a worst-case approach using model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, Vol. 32. 10 pages.
[71] Xian Yeow Lee, Yasaman Esfandiari, Kai Liang Tan, and Soumik Sarkar. 2021. Query-based targeted action-space adversarial policies on deep reinforcement learning agents. In *International Conference on Cyber-Physical Systems*. 87–97.
[72] Xian Yeow Lee, Sambit Ghadai, Kai Liang Tan, Chinmay Hegde, and Soumik Sarkar. 2020. Spatiotemporally constrained action space attacks on deep reinforcement learning agents. In *AAAI Conference on Artificial Intelligence*. 4577–4584.
[73] Antoine Lemay and Sylvain Leblanc. 2019. Operational tempo in cyber operations. In *European Conference on Cyber Warfare and Security*. ACAD CONFERENCES LTD Location NR READING, 275–281.
[74] Michelle Li and Michael Dennis. 2023. The Benefits of Power Regularization in Cooperative Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. 457–465.
[75] Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. 2022. Efficient Adversarial Training without Attacking: Worst-Case-Aware Robust Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 35. 22547–22561.
[76] Jieyu Lin, Kristina Dzeparoska, Sai Qian Zhang, Alberto Leon-Garcia, and Nicolas Papernot. 2020. On the Robustness of Cooperative Multi-Agent Reinforcement Learning. In *IEEE Security and Privacy Workshops*. 62–68.
[77] Yen-Chen Lin, Zhang-Wei Hong, Yuan-Hong Liao, Meng-Li Shih, Ming-Yu Liu, and Min Sun. 2017. Tactics of Adversarial Attack on Deep Reinforcement Learning Agents. In *International Joint Conference on Artificial Intelligence*, Vol. 0. 3756–3762.
[78] Z. Liu, Z. Guo, Z. Cen, H. Zhang, Y. Yao, H. Hu, and D. Zhao. 2023. Towards Robust and Safe Reinforcement Learning with Benign Off-policy Data. In *International Conference on Machine Learning*, Vol. 202. 22249–22265.
[79] Ryan Lowe, Jakob Foerster, Y.-Lan Boureau, Joelle Pineau, and Yann Dauphin. 2019. On the Pitfalls of Measuring Emergent Communication. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 2. 693–701.
[80] C. Lu, T. Willi, A. Letcher, and J. Foerster. 2023. Adversarial Cheap Talk. In *International Conference on Machine Learning*, Vol. 202. 22917–22941.
[81] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*. 23 pages.
[82] Antoine Marot, Isabelle Guyon, Benjamin Donnot, Gabriel Dulac-Arnold, Patrick Panciatici, Mariette Awad, Aidan O'Sullivan, Adrian Kelly, and Zigfried Hampel-Arias. 2020. L2RPN: Learning to Run a Power Network in a Sustainable World NeurIPS2020 challenge design. , 26 pages.
[83] Rupert Mitchell, Jan Blumenkamp, and Amanda Prorok. 2020. Gaussian process based message filtering for robust multi-agent cooperation in the presence of adversarial communication.

[84] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. In *NIPS Deep Learning Workshop*. 9 pages.

[85] Christopher Molloy, Steven HH Ding, Benjamin CM Fung, and Philippe Charland. 2022. H4rm0ny: A Competitive Zero-Sum Two-Player Markov Game for Multi-Agent Learning on Evasive Malware Generation and Detection. In *IEEE International Conference on Cyber Security and Resilience*. 22–29.

[86] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal Adversarial Perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition*. 86–94.

[87] R. Mu, L.S. Marcolino, Y. Zhang, T. Zhang, X. Huang, and W. Ruan. 2024. Reward Certification for Policy Smoothed Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, Vol. 38. 21429–21437. https://doi.org/10.1609/aaai.v38i19.30139

[88] Micah Musser, Andrew Lohn, James X. Dempsey, Jonathan Spring, Ram Shankar Siva Kumar, Brenda Leong, Christina Liaghati, Cindy Martinez, Crystal D. Grant, Daniel Rohrer, Heather Frase, John Bansemer, Mikel Rodriguez, Mitt Regan, Rumman Chowdhury, and Stefan Hermanek. 2023. *Adversarial Machine Learning and Cybersecurity.* Technical Report. Center for Security and Emerging Technology.

[89] Kohei Ohashi, Kosuke Nakanishi, Wataru Sasaki, Yuji Yasui, and Shin Ishii. 2021. Deep Adversarial Reinforcement Learning With Noise Compensation by Autoencoder. *IEEE Access* 9 (2021), 143901–143912.

[90] Tuomas Oikarinen, Wang Zhang, Alexandre Megretski, Tsui-Wei Weng, and Luca Daniel. 2021. Robust Deep Reinforcement Learning through Adversarial Loss. In *Advances in Neural Information Processing Systems*, Vol. 34. 26156–26167.

[91] Frans A. Oliehoek and Christopher Amato. 2016. *A Concise Introduction to Decentralized POMDPs.* Springer Cham.

[92] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning.

[93] Alexander Pan, Yongkyun Lee, Huan Zhang, Yize Chen, and Yuanyuan Shi. 2021. Improving Robustness of Reinforcement Learning for Power System Control with Adversarial Training.

[94] Xinlei Pan, Chaowei Xiao, Warren He, Shuang Yang, Jian Peng, Mingjie Sun, Jinfeng Yi, Zijiang Yang, Mingyan Liu, Bo Li, et al. 2022. Characterizing Attacks on Deep Reinforcement Learning. In *International Conference on Autonomous Agents and Multiagent Systems*. 1010–1018.

[95] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples.

[96] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. 2016. The Limitations of Deep Learning in Adversarial Settings. In *IEEE European Symposium on Security and Privacy*. 372–387.

[97] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2018. Robust Deep Reinforcement Learning with adversarial attacks. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 3. 2040–2042.

[98] Thomy Phan, Thomas Gabor, Andreas Sedlmeier, Fabian Ritz, Bernhard Kempter, Cornel Klein, Horst Sauer, Reiner Schmid, Jan Wieghardt, Marc Zeller, et al. 2020. Learning and testing resilience in cooperative multi-agent systems. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 2020-May. 1055–1063.

[99] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, Vol. 6. 4310–4319.

[100] Damian Poddebniak, Fabian Ising, Hanno Böck, and Sebastian Schinzel. 2021. Why TLS is better without STARTTLS: A Security Analysis of STARTTLS in the Email Context. In *USENIX Security Symposium*. 4365–4382.

[101] Amanda Prorok, Matthew Malencia, Luca Carlone, Gaurav S. Sukhatme, Brian M. Sadler, and Vijay Kumar. 2021. Beyond Robustness: A Taxonomy of Approaches towards Resilient Multi-Robot Systems.

[102] Óscar Pérez-Gil, Rafael Barea, Elena López-Guillén, Luis M. Bergasa, Carlos Gómez-Huélamo, Rodrigo Gutiérrez, and Alejandro Díaz-Díaz. 2022. Deep reinforcement learning based control for Autonomous Vehicles in CARLA. *Multimedia Tools and Applications* 81, 3 (Jan. 2022), 3553–3576.

[103] You Qiaoben, Chen Ying, Xinning Zhou, Hang Su, Jun Zhu, and Bo Zhang. 2024. Understanding Adversarial Attacks on Observations in Deep Reinforcement Learning. *Science China Information Sciences* 67 (2024), 15 pages.

[104] You Qiaoben, Xinning Zhou, Chen Ying, and Jun Zhu. 2021. Strategically-timed State-Observation Attacks on Deep Reinforcement Learning Agents. In *Workshop on Adversarial Machine Learning*. 8 pages.

[105] Ao Qu, Yihong Tang, and Wei Ma. 2023. Adversarial Attacks on Deep Reinforcement Learning-based Traffic Signal Control Systems with Colluding Vehicles. *ACM Transactions on Intelligent Systems and Technology* 14, 6 (2023), 1–22.

[106] Xinghua Qu, Abhishek Gupta, Yew-Soon Ong, and Zhu Sun. 2023. Adversary Agnostic Robust Deep Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* 34, 9 (2023), 6146–6157.

[107] Aowabin Rahman, Arnab Bhattacharya, Thiagarajan Ramachandran, Sayak Mukherjee, Himanshu Sharma, Ted Fujimoto, and Samrat Chatterjee. 2022. AdverSAR: Adversarial Search and Rescue via Multi-Agent Reinforcement Learning. In *IEEE International Symposium*

on Technologies for Homeland Security. 7 pages.
[108] M.A. Rahman and S. Alqahtani. 2023. Task-Agnostic Safety for Reinforcement Learning. In *ACM Workshop on Artificial Intelligence and Security*. 139–148.
[109] M.A. Rahman, T. Liu, and S. Alqahtani. 2023. Adversarial Behavior Exclusion for Safe Reinforcement Learning. In *International Joint Conference on Artificial Intelligence*, Vol. 2023-August. 483–491.
[110] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, Vol. 21. 1–51.
[111] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial Attacks and Defenses in Deep Learning. *Engineering* 6, 3 (March 2020), 346–360.
[112] Alessio Russo and Alexandre Proutiere. 2021. Towards Optimal Attacks on Reinforcement Learning Policies. In *American Control Conference*. 4561–4567.
[113] Mikayel Samvelyan, Davide Paglieri, Minqi Jiang, Jack Parker-Holder, and Tim Rocktäschel. 2024. Multi-Agent Diagnostics for Robustness via Illuminated Diversity. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. 1630–1644.
[114] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft multi-agent challenge. In *International Conference on Autonomous Agents and MultiAgent Systems*, Vol. 4. 2186–2188.
[115] Uri Shaham, Kelly P Stanton, Jun Zhao, Huamin Li, Khadir Raddassi, Ruth Montgomery, and Yuval Kluger. 2017. Removal of batch effects using distribution-matching residual networks. *Bioinformatics* 33, 16 (Aug. 2017), 2539–2546.
[116] David Silver, Aja Huang, Christopher Maddison, Arthur Guez, Laurent Sifre, George Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529 (Jan. 2016), 484–489.
[117] Maxwell Standen, Martin Lucas, David Bowman, Toby J Richer, and Junae Kim. 2021. CybORG: A Gym for the Development of Autonomous Cyber Agents. In *International Workshop on Adaptive Cyber Defense*. 7 pages.
[118] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation* 23, 5 (Oct. 2019), 828–841.
[119] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2016. Learning Multiagent Communication with Backpropagation. In *Advances in Neural Information Processing Systems*. 2252–2260.
[120] Jianwen Sun, Tianwei Zhang, Xiaofei Xie, Lei Ma, Yan Zheng, Kangjie Chen, and Yang Liu. 2020. Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*. 5883–5891.
[121] Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang. 2023. Certifiably Robust Policy Learning against Adversarial Communication in Multi-agent Systems. In *International Conference on Learning Representations*. 30 pages.
[122] Yanchao Sun, Ruijie Zheng, Yongyuan Liang, and Furong Huang. 2021. Who Is the Strongest Enemy? Towards Optimal and Efficient Evasion Attacks in Deep RL. In *International Conference on Learning Representations*. 40 pages.
[123] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*. 10 pages.
[124] Elham Tabassi, Kevin J. Burns, Michael Hadjimichael, Andres D. Molina-Markham, and Julian T. Sexton. 2019. *A taxonomy and terminology of adversarial machine learning*. preprint 8269. National Institute of Standards and Technology. 35 pages.
[125] Xiaocheng Tang, Zhiwei Qin, Fan Zhang, Zhaodong Wang, Zhe Xu, Yintai Ma, Hongtu Zhu, and Jieping Ye. 2019. A deep value-network based approach for multi-driver order dispatching. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1780–1790.
[126] Buse GA Tekgul, Shelly Wang, Samuel Marchal, and N Asokan. 2022. Real-time Adversarial Perturbations against Deep Reinforcement Learning Policies: Attacks and Defenses. In *European Symposium on Research in Computer Security*. 384–404.
[127] Chen Tessler, Yonathan Efroni, and Shie Mannor. 2019. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*. 6215–6224.
[128] Finbarr Timbers, Nolan Bard, Edward Lockhart, Marc Lanctot, Martin Schmid, Neil Burch, Julian Schrittwieser, Thomas Hubert, and Michael Bowling. 2022. Approximate Exploitability: Learning a Best Response. In *International Joint Conference on Artificial Intelligence*. 3487–3493.
[129] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 5026–5033.
[130] Edgar Tretschk, Seong Joon Oh, and Mario Fritz. 2018. Sequential Attacks on Agents for Long-Term Adversarial Goals. In *ACM Computer Science in Cars Symposium*. 9 pages.

[131] James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. 2021. Adversarial Attacks On Multi-Agent Communication. In *IEEE/CVF International Conference on Computer Vision*. 7748–7757.
[132] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. 2018. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, Vol. 11. 7995–8007.
[133] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (Nov. 2019), 350–354.
[134] Xu Wan, Lanting Zeng, and Mingyang Sun. 2022. Exploring the Vulnerability of Deep Reinforcement Learning-based Emergency Control for Low Carbon Power Systems. In *International Joint Conference on Artificial Intelligence*. 3954–3961.
[135] Pengyue Wang, Yan Li, Shashi Shekhar, and William F. Northrop. 2020. Adversarial Attacks on Reinforcement Learning based Energy Management Systems of Extended Range Electric Delivery Vehicles.
[136] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. 2020. Learning Nearly Decomposable Value Functions via Communication Minimization. In *International Conference on Learning Representations*. 15 pages.
[137] Yulong Wang, Tong Sun, Shenghong Li, Xin Yuan, Wei Ni, Ekram Hossain, and H Vincent Poor. 2023. Adversarial Attacks and Defenses in Machine Learning-Powered Networks: A Contemporary Survey.
[138] Tsui-Wei Weng, Krishnamurthy (Dj) Dvijotham, Jonathan Uesato, Kai Xiao, Sven Gowal, Robert Stanforth, and Pushmeet Kohli. 2020. Toward Evaluating Robustness of Deep Reinforcement Learning with Continuous Control. In *International Conference on Learning Representations*. 13 pages.
[139] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *International Conference on Evaluation and Assessment in Software Engineering*. 1–10.
[140] Fan Wu, Linyi Li, Zijian Huang, Yevgeniy Vorobeychik, Ding Zhao, and Bo Li. 2022. CROP: Certifying Robust Policies for Reinforcement Learning through Functional Smoothing. In *International Conference on Learning Representations*. 34 pages.
[141] J. Wu and Y. Vorobeychik. 2022. Robust Deep Reinforcement Learning through Bootstrapped Opportunistic Curriculum. In *International Conference on Machine Learning*, Vol. 162. 24177–24211.
[142] Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. 2021. Adversarial Policy Training against Deep Reinforcement Learning. In *USENIX Security Symposium*. 1883–1900.
[143] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. 2018. Generating adversarial examples with adversarial networks. In *International Joint Conference on Artificial Intelligence*, Vol. 2018-July. 3905–3911.
[144] Wanqi Xue, Wei Qiu, Bo An, Zinovi Rabinovich, Svetlana Obraztsova, and Chai Kiat Yeo. 2022. Mis-spoke or mis-lead: Achieving Robustness in Multi-Agent Communicative Reinforcement Learning. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, Vol. 3. 1418–1426.
[145] Chao-Han Huck Yang, I-Te Danny Hung, Yi Ouyang, and Pin-Yu Chen. 2022. Training a Resilient Q-network against Observational Interference. In *AAAI Conference on Artificial Intelligence*, Vol. 36. 8814–8822.
[146] R. Yang, C. Bai, X. Ma, Z. Wang, C. Zhang, and L. Han. 2022. RORL: Robust Offline Reinforcement Learning via Conservative Smoothing. In *Advances in Neural Information Processing Systems*, Vol. 35. 16 pages.
[147] Yunyun Yang and Gang Xie. 2016. Efficient identification of node importance in social networks. *Information Processing & Management* 52, 5 (Sept. 2016), 911–922.
[148] X. Yin, S. Wu, J. Liu, M. Fang, X. Zhao, X. Huang, and W. Ruan. 2024. Representation-Based Robustness in Goal-Conditioned Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, Vol. 38. 21761–21769. https://doi.org/10.1609/aaai.v38i19.30176
[149] Chengyang Ying, Xinning Zhou, Dong Yan, and Jun Zhu. 2021. Towards Safe Reinforcement Learning via Constraining Conditional Value at Risk. In *Workshop on Adversarial Machine Learning*. 3673–3680.
[150] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, Vol. 35. 24611–24624.
[151] L. Yu, Y. Qiu, Q. Yao, Y. Shen, X. Zhang, and J. Wang. 2024. Robust Communicative Multi-Agent Reinforcement Learning with Active Defense. In *AAAI Conference on Artificial Intelligence*, Vol. 38. 17575–17582. https://doi.org/10.1609/aaai.v38i16.29708
[152] Mengran Yu and Shiliang Sun. 2022. Natural Black-Box Adversarial Examples against Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, Vol. 36. 8936–8944.
[153] L. Yuan, Z. Zhang, K. Xue, H. Yin, F. Chen, C. Guan, L. Li, C. Qian, and Y. Yu. 2023. Robust Multi-Agent Coordination via Evolutionary Generation of Auxiliary Adversarial Attackers. In *AAAI Conference on Artificial Intelligence*, Vol. 37. 11753–11762.
[154] Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. 2021. Robust Reinforcement Learning on State Observations with Learned Optimal Adversary. In *International Conference on Learning Representations*. 16 pages.

[155] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. 2020. Robust Deep Reinforcement Learning against Adversarial Perturbations on State Observations. In *Advances in Neural Information Processing Systems*, Vol. 33. 21024–21037.
[156] Mingyue Zhang, Nianyu Li, Jialong Li, Jiachun Liao, and Jiamou Liu. 2024. Memory-Based Resilient Control Against Non-cooperation in Multi-agent Flocking. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. 2075–2084.
[157] Yan Zheng, Ziming Yan, Kangjie Chen, Jianwen Sun, Yan Xu, and Yang Liu. 2021. Vulnerability Assessment of Deep Reinforcement Learning Models for Power System Topology Optimization. *IEEE Transactions on Smart Grid* 12, 4 (2021), 3613–3623.
[158] Changxi Zhu, Mehdi Dastani, and Shihan Wang. 2022. A Survey of Multi-Agent Reinforcement Learning with Communication.
[159] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. Regret Minimization in Games with Incomplete Information. In *Advances in Neural Information Processing Systems*, Vol. 20. 8 pages.

Received 5 December 2023; revised 8 November 2024; accepted 22 November 2024