Security and Privacy Issues in Deep Reinforcement Learning: Threats and Countermeasures

KANGHUA MO, Guangzhou University, China and Guangdong Provincial Key Laboratory of Blockchain Security, China
PEIGEN YE, Beijing Institute of Technology, China
XIAOJUN REN, Guangzhou University, China and Guangdong Provincial Key Laboratory of Blockchain Security, China
SHAOWEI WANG and WENJUN LI, Guangzhou University, China
JIN LI, Guangzhou University, China and Guangdong Provincial Key Laboratory of Blockchain Security, China

Deep Reinforcement Learning (DRL) is an essential subfield of Artificial Intelligence (AI), where agents interact with environments to learn policies for solving complex tasks. In recent years, DRL has achieved remarkable breakthroughs in various tasks, including video games, robotic control, quantitative trading, and autonomous driving. Despite its accomplishments, security and privacy-related issues still prevent us from deploying trustworthy DRL applications. For example, by manipulating the environment, an attacker can influence an agent's actions, misleading it to behave abnormally. Additionally, an attacker can infer private training data and environmental information by maliciously interacting with DRL models, causing a privacy breach. In this survey, we systematically investigate the recent progress of security and privacy issues in the context of DRL. First, we present a holistic review of security-related attacks within DRL systems from the perspectives of single-agent and multi-agent systems and review privacy-related attacks. Second, we review and classify defense methods used to address security-related challenges, including robust learning, anomaly detection, and game theory approaches. Third, we review and classify privacy-preserving technologies, including encryption, differential privacy, and policy confusion. We conclude the survey by discussing open issues and possible directions for future research in this field.

CCS Concepts: • **Security and privacy**; • **Computing methodologies** → *Control methods*;

Additional Key Words and Phrases: Deep reinforcement learning, adversarial attack, defense

ACM Reference Format:
Kanghua Mo, Peigen Ye, Xiaojun Ren, Shaowei Wang, Wenjun Li, and Jin Li. 2024. Security and Privacy Issues in Deep Reinforcement Learning: Threats and Countermeasures. *ACM Comput. Surv.* 56, 6, Article 152 (February 2024), 39 pages. https://doi.org/10.1145/3640312

This work was supported by the National Natural Science Foundation of China for Joint Fund Project (No.U1936218), the National Natural Science Foundation of China (No.U23A20307, No.62372120, No.62102108), and the Natural Science Foundation of Guangdong Province of China (No.2022A1515010061).
Authors' addresses: K. Mo, X. Ren (Corresponding author), and J. Li, Guangzhou University, Guangzhou, Guangdong, China and Guangdong Provincial Key Laboratory of Blockchain Security, Guangzhou, Guangdong, China; e-mails: mokanghua@gmail.com, renxiaojun@gzhu.edu.cn, jinli71@gmail.com; P. Ye (Corresponding author), Beijing Institute of Technology, Beijing, China; e-mail: ypgmhxy@gmail.com; S. Wang and W. Li, Guangzhou University, Guangzhou, Guangdong, China; e-mails: wangsw@gzhu.edu.cn, wenjun1999@e.gzhu.edu.cn.

ACM 0360-0300/2024/02-ART152
https://doi.org/10.1145/3640312

1 INTRODUCTION

In the realm of **Artificial Intelligence (AI), Deep Reinforcement Learning (DRL)** has notably emerged as a rapidly advancing technique in recent years. Its development is increasingly regarded as a cornerstone for the realization of **General Artificial Intelligence (GAI)**, offering new prospects in this ambitious domain [129]. DRL represents an innovative amalgamation of **Reinforcement Learning (RL)** and **Deep Learning (DL)**, two pivotal areas in AI. Historically, RL has encountered challenges in addressing complex real-world scenarios, notably in continuous systems [4] and in the context of intricate optimization problems [87, 172]. However, the synergistic combination of RL with **deep neural networks (DNNs)** within DRL has unlocked the potential to effectively engage with these realistic and multifaceted tasks, marking a significant leap in the field. DRL has found applications in numerous fields, including language assistants [8, 100], stock trading [47, 173], recommendation systems [25, 76, 187], autonomous driving [20, 66], cybersecurity [37, 95, 136, 137], and manufacturing control [96, 118, 176].

However, recent research has highlighted the vulnerability of DRL to adversarial attacks, also known as security-related attacks. These attacks can potentially manipulate the decision-making process of the DRL system, thus undermining its integrity. In addition, privacy-related attacks on DRL pose considerable risks by potentially compromising the confidentiality of sensitive information, including personally identifiable data or DRL's functional models, which may lead to privacy breaches. These privacy-related attacks could also serve as stepping stones for other attacks. Therefore, it is imperative to thoroughly understand the risks and countermeasures associated with DRL before deploying DRL-based critical systems. Despite noteworthy contributions such as the initial survey on security concerns [13, 56, 62], trustworthiness [169], there remains a need for surveying security and privacy issues specifically in the context of DRL.

This article provides a narrative review of the prevalent security and privacy-related attacks and their corresponding defenses within DRL. We have structured our discussion into three key sections: (1) Threats: This segment surveys potential security and privacy-related threats, considering scenarios involving single and multiple agents; (2) Countermeasures: This section surveys a range of defenses designed to combat these security threats, focusing on robust learning, anomaly detection, and the game-theory approach; (3) Privacy-preserving methods: In this section, we review techniques designed to maintain privacy, focusing on methods such as differential privacy, cryptography, and policy confusion.

1.1 Contributions

The main contributions of this article can be summarized as follows:

(1) We present the first comprehensive survey of attacks on the security and privacy of DRL.
(2) We provide a novel taxonomy to categorize the latest attacks, including threats from single-agent and multi-agent systems and security-related and privacy-related attacks.
(3) We offer a comprehensive review of countermeasures against DRL attacks, including both security-related defenses and privacy-preserving methods. We evaluate these works from the perspective of taxonomy, prerequisite, methodology, and effectiveness.
(4) We identify and discuss several important open issues, providing directions for future research.

1.2 Overview

In this article, we review state-of-the-art methods regarding the security and privacy of DRL. As shown in Figure 1, this article focuses on security issues during the training and testing phases, such as data poisoning, adversarial, and communication attacks. Additionally, this article covers

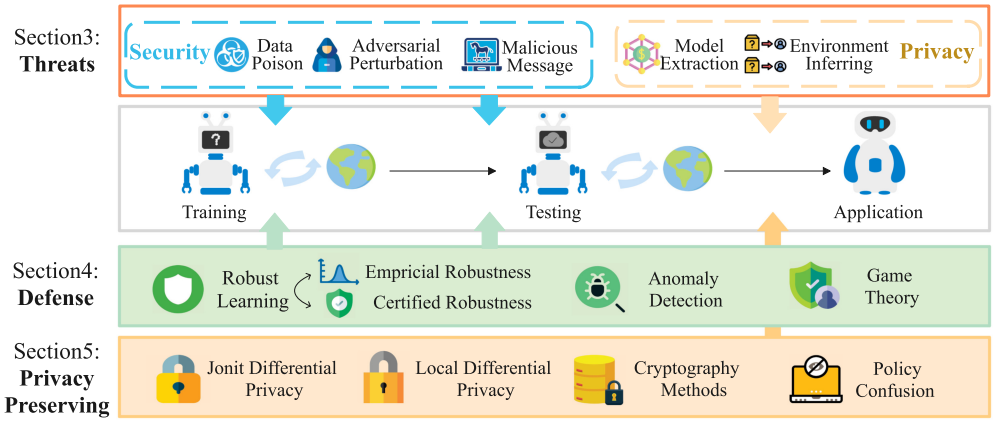


Fig. 1. Overview of threats and countermeasures in deep reinforcement learning.

privacy issues during the deployment and application phase, including model, environment, and reward privacy attacks. Then, we introduce the defenses and privacy-preserving methods against these threats, respectively. The remainder of this article is organized as follows: Section 2 presents the essential background knowledge of DRL and briefly states the typical security and privacy issues of DL. In Section 3, we classify existing security and privacy attacks on DRL systems. Section 4 focuses on the defense methods of DRL, followed by privacy preservation methods in Section 5. In Section 6, we present open issues and research challenges, suggesting promising directions for future research on this topic. We draw our conclusions in Section 7.

2 PRELIMINARIES

2.1 Deep Reinforcement Learning

RL involves agents interacting with dynamic environments, using their experiences to optimize decision-making policies. These agents utilize a trial-and-error process to optimize their policies [135], thereby maximizing the cumulative rewards from the environments. Formally, a generic RL problem is modeled as a **Markov Decision Process (MDP)**, which is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$. In this tuple, $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P}$ denotes the transition probability, r denotes the reward function, and γ denotes the discount factor. We briefly introduce these and other commonly used RL concepts as follows:

- **Action space.** It refers to the entire set of actions (decisions) the environment allows an agent to make, which may be discrete, continuous, or mixed.
- **Environment.** This term refers to the simulator with which an agent interacts, representing the rules or mechanisms of the interaction process. The environment provides observations to the agent, returns a reward for the agent's actions, and transitions to the next state according to $\mathcal{P}$. Based on observations, environments can be divided into two categories, i.e., fully observable and partially observable. In a fully observable environment, an agent can observe all the information in the state. For partially observable environments, a part of the state is invisible to the agent, so this agent can only learn based on partial observations.
- **Reward.** A value returned by the environment after an agent takes an action. The cumulative reward is the sum of all the rewards during an episode.
- **Episode.** A sequence of interactions between an agent and its environment, starting from an initial state and ending at a terminal state.

— **Policy.** It determines how an agent will behave in the environment, which is a teleologically oriented subset of all possible behaviors. A policy is optimal if it enables the agent to achieve the maximum possible cumulative reward. Policies are further divided into two types: deterministic policy and stochastic policy. A policy is called deterministic if the actions taken by the agent are deterministic. However, when actions are sampled from a conditional probability distribution of actions in a given state, a policy is said to be stochastic.

At each timestep $t \in T$, based on the state $s_t \in \mathcal{S}$, the agent chooses an action $a_t \in \mathcal{A}$ according to its policy $\pi(s_t)$. Then, a new state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$ and a reward $r(s_t, a_t)$ from the environment are given to the agent. The return R of a policy π is computed as $R = \sum_{t=0}^{T-1} \gamma^t r(s_t, \pi(s_t))$. The value function $V_\pi(s)$, defined as $V_\pi(s) = E[R \mid s, \pi]$, is the expected return when starting in state s and following π thereafter. Further, the Q value function is proposed to measure the value of each action at each state, which is defined as $Q_\pi(s, a) = E[R \mid s, a, \pi]$.

To maximize the cumulative reward, the core objective of RL is to obtain an optimal policy by updating its policy to satisfy the following condition:

$$\pi^* = \arg\max_\pi Q_\pi(s, a), \forall s \in S, \forall a \in A. \tag{1}$$

However, due to the difficulty of learning in complex environments, RL algorithms still face some limitations in real-world applications. Combining the advancement of DL, the key idea of DRL is applying DL technology to approximate the value function or Q value function, which significantly enhances agent's capabilities in complex environments. DRL methods can be divided into two categories according to the number of agents in the environment: single-agent RL and multi-agent RL.

2.1.1 Single-agent Reinforcement Learning. In single-agent RL, an agent learns a policy by interacting with the dynamic environment through a trial-and-error procedure. Most existing single-agent frameworks assume that the agent can fully observe the state of the environment, thereby satisfying the Markov property. In this case, the transition probability can be determined solely based on observation and actions taken without considering the history of previous states, i.e., the transition probability is conditionally independent of the transition history. In situations where the agent cannot fully observe the state of the environment, the agent can estimate the representation of the true state based on past observed states. Hence, the Markov property of an MDP is based on a statistically sufficient representation, known as **partially observable Markov decision processes (POMDP)**.

Over the past few years, with the advances in DL, a multitude of single-agent DRL algorithms has been presented. Depending on whether users have access to an environmental model, these algorithms are broadly categorized into model-based and model-free types.

Model-based algorithms construct environmental models, providing the transition probability and the associated reward before or during the training phase. Commonly used model-based algorithms include DYNA [134], AlphaZero [128], and MuZero [120]. In contrast, model-free algorithms, with no prior knowledge about the environment model, learn the optimal policy by directly optimizing the policy or other value parameters.

Model-free algorithms aim to directly learn the value function or policy during the interactions with the environment. Value function methods exploit the Bellman equation [14] to learn the Q function, which has the following recursive form:

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r(s_t, a_t) + \gamma Q_\pi(s_{t+1}, \pi(s_{t+1}))]. \tag{2}$$

This implies that Q_π can be improved by bootstrapping, i.e., the current values of Q^π can enhance its estimate.

Policy-based methods, instead of estimating the value function, utilize a policy that is parameterized by a neural network to search for an optimal policy. This policy represents a probability distribution of actions over states and is directly optimized by gradient-based for reaching the maximum returns.

2.1.2 Multi-agent Reinforcement Learning. In a single-agent setting, the agent is the sole decision-maker in the environment. Therefore, transitions can be attributed to the agent, making the agent's learning problem stationary. In contrast, in a multi-agent domain, agents may update their policies during the learning process simultaneously, such that the environment appears non-stationary from the perspective of a single agent [93]. So, it becomes necessary to consider the impact of other agents. Formally, multi-agent RL can be modeled as Markov Games. The Markov Games, extending game theory to MDPs, were introduced by Reference [79] to generalize MDPs to multiple agents interacting within a shared environment and with each other. The Markov Games are formalized by the tuple $\langle \mathcal{N}, \mathcal{X}, \{\mathcal{S}\}, \mathcal{P}, \{r_i\}, \gamma \rangle$, where $\mathcal{N} = \{1, \ldots, N\}$ denotes the set of interacting agents and $\mathcal{S}$ is the set of states observed by all agents. The action space of multiple agents is called the joint action space, which is denoted by $\mathcal{U} = \mathcal{U}^1 \times \cdots \times \mathcal{U}^N$. $\mathcal{P}$ is the transition probability function. Each agent owns an associated reward function $r^i : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow \mathbb{R}$ and γ describes the discount factor.

At timestep t, each agent $i \in N$ selects and executes an action depending on the individual policy π_i. The environment evolves from state S_t under the joint action t concerning the transition probability function $\mathcal{P}$ to the next state s_{t+1} while each agent receives its reward according to the reward function r_i. Similar to the single-agent problem, each agent aims to change its policy in such a way as to maximize the received rewards from a long-term perspective.

According to the tasks of multiple agents, multi-agent RL can be divided into three types: fully cooperative, fully competitive, and mixed. In a fully cooperative setting, all agents share the same reward, that is, $\mathbb{R} = \mathbb{R}^i, i \in N$. Agents are seen as a group and motivated to cooperate for maximum rewards. Cooperative settings are those where agents are encouraged to cooperate without owning an equally shared reward. A fully competitive setting is described as a zero-sum Markov Game, where the sum of the rewards of both parties is zero. In this setting, agents are encouraged to contend with others for the limited reward. In a wide sense, competitive games also comprise agents urged to outperform opponents while the total rewards do not equal zero. The mixed setting, also known as the general-sum game, is neither fully cooperative nor fully competitive and, thus, does not incorporate restrictions on agent goals.

2.2 Security of Machine Learning

In recent years, **machine learning (ML)** has achieved notable success and benefits in many fields, including image classification, speech recognition, and natural language processing. However, it is also fragile and susceptible to being fooled or attacked easily. In this section, we describe some typical attack and defense methods in ML.

2.2.1 Taxonomy of Machine Learning Attack. Many ML technologies in use today are vulnerable to various adversarial attacks, causing numerous DL-based applications to face serious security problems. According to the attacks' stages, the existing attacks can be divided into two types: training-phase attacks and testing-phase attacks.

Training phase. ML models, especially DL models, are trained on collected data through a task-oriented training mechanism during the training phase. As data is the driving force behind the DL, some attacks aimed at introducing deviations or errors into the collected data, such as the malicious data attack and the sensor spoofing attack. Attackers may publish malicious data on the Internet for crawlers to collect. If this data is collected without checking, then it will affect

the quality of the whole data collection. The sensor spoofing attack is when the attacker performs the sensor attack by tampering with the data provided by sensors [126, 131]. As a typical post-processing operation after data collection, image scaling could be maliciously utilized as an attack mechanism, i.e., the scaling attack [110, 161]. The attacker misuses the scaling algorithm to craft a camouflage image of a normal unscaled one, leading to a dramatic change in visual semantics before and after image scaling. Another type of attack against training data is the poisoning attack, wherein an attacker poisons the training set to manipulate the inference behavior of the DL model.

Testing phase. In the testing phase, the most common security threat of DL, an adversarial example, can mislead the model f to produce incorrect predictions. An adversarial example consists of a clean sample x with $f(x) = y_{real}$ and an imperceptible perturbation δ, which is calculated as $x^{'} = x + \delta$. According to the outputs of $f(x^{'})$, adversarial examples are divided into the untargeted attack and the targeted attack. In terms of the untargeted attack, the $x^{'} = x + \delta$ will mislead the DL model to output a wrong class $f(x^{'}) \neq y_{real}$. In terms of the targeted attack, the model will be fooled to output a special class $f(x^{'}) = y_{target}$ as the attacker wants.

There are various types of calculating the perturbation δ. A famous method proposed by Goodfellow et al. [43], namely, the **Fast Gradient Sign Method (FGSM)**, calculates the perturbation in the white-box setting as follows:

$$\delta = \epsilon \cdot sign\left(\nabla_x J_\theta(x, y_{real})\right), \tag{3}$$

where ϵ corresponds to the magnitude of perturbation, θ is the parameters of a model, and $J_\theta(\cdot)$ is the cost function. Based on this, an iterative version of FGSM proposed by Kurakin et al. [68] generated more aggressive perturbations. A universal perturbation was generated by Moosavi-Dezfooli et al. [91], which can be added to some clean samples instead of only one. In addition, there are many efficient algorithms to generate adversarial examples, including ZOO [23] and DeepFool [92].

2.2.2 Defenses of Machine Learning. Research on defense strategies is rapidly developing as countermeasures against ML attacks. Existing defense strategies can be roughly divided into heuristic and certificated defenses.

Heuristic defense. Some empirical heuristic defense methods show great ability to mitigate the effect of adversarial attacks, including adversarial training, data randomization, and denoised methods. Adversarial training is one of the most successful ways to improve the robustness of ML models. The principle is to use a large number of adversarial examples crafted by some selected adversarial attacks with correct labels, mixing them with the normal data to train the neural networks. Models trained this way will exhibit remarkable resistance to the adversarial attacks used for training. However, the limitation is that the improvement of robustness is limited to the adversarial examples used in adversarial training, while the number of adversarial examples is infinite and will continue to improve with the development of attack methods, and there are still new adversarial examples that can deceive the network again [149]. The data randomization methods [105, 162] refer to some randomization operations on the input samples, such as panning, rotating, clipping, scaling, filling, and so on, which could mitigate the impact of adversarial perturbations in the input/feature domain to some extent. The randomization-based defense is simple to implement and can achieve good performance in some settings of black-box and gray-box attacks, but this approach can only defend against some weak attacks and is not effective for most well-designed attacks. In addition, denoising [26, 163] is a relatively simple but effective defense method. Previous work has pointed to two directions for this defense design: input denoising [26] and feature denoising [163]. The former attempted to partially or completely remove adversarial perturbations

from the input, while the latter tried to mitigate the impact of adversarial perturbations on the high-level features learned by DNNs.

Certificated defense. The aforementioned defense methods are evaluated based on analysis and iterative experiments on existing adversarial attacks, and they may be bypassed once new attacks emerge. Thus, certificated defense methods were introduced, which provide theoretical guarantees against adversarial attacks. These provable defense methods always maintain some accuracy under a well-defined class of attacks [114]. A famous certificated method is proposed in Reference [154], which was designed for training provably robust deep ReLU classifiers. The classifiers are guaranteed to be robust against any norm-bounded adversarial perturbations on the training set. However, this kind of method has high computational complexity and needs to be designed manually for heterogeneous DNNs. For deep ReLU classifiers, several computationally efficient methods [152] have been proposed. In addition, some efforts [150, 168, 184] have been made to certify the robustness of neural networks with general activation functions.

2.3 Privacy of Machine Learning

With the rapid development of big data, DL models have been widely used in various fields, such as intelligent medicine, face recognition, and natural language processing. However, in recent years, researchers have found many privacy risks in the current mainstream AI models, which will limit the further development of AI technology. We describe privacy attacks and defense methods in ML.

2.3.1 Taxonomy of Privacy Attacks. In privacy attacks of ML, the goal of the attacker is to acquire knowledge that **ML as a service (MlaaS)** does not intend to share, such as knowledge about training data D or information about models M, and even extract information about data properties, such as unintentional coding biases. According to the difference in privacy knowledge attained by stealing, the existing privacy attacks can be divided into four types: membership inference attack, property inference attack, reconstruction attack, and model extraction attack.

Membership inference attack. The **membership inference attack (MIA)** attempts to determine whether a specific input sample x belongs to the training set D. The victim model is often trained with sensitive training data. Through this attack, the attacker can infer the membership information of the training set of the target model, resulting in the disclosure of the privacy of sensitive data: for example, if the data collected from patients with a certain disease is the training set of an ML model. Once it is known that certain data belongs to the training data of the model, the attacker can immediately know the health status of the owner of the data. Specifically, given a specific selected input sample x and a target model M (victim model) trained with the training set D, the attacker inputs x into M to obtain the corresponding prediction result $\hat{y}$. Then, the attacker can infer whether x belongs to D according to the $\hat{y}$. Shokri et al. [127] first introduced a membership inference attack against ML models in 2017 and proposed a membership inference method by building an attack model with the help of the shadow model. Since then, the game between attack and defense has continued to heat up.

MIA frequently occurs in supervised ML models and can be executed in the case of black boxes, where the attacker only has knowledge of the model's prediction vector (black box). In contrast, a white-box attack is also a threat, especially in a collaborative environment, where attackers can launch passive and active attacks. The access to model parameters and gradients allows for a more effective white box MIA in terms of attack accuracy. In addition to supervised models, generative models such as GAN and VAE are also vulnerable to MIA. In this case, the attacker has resorted to different degrees of knowledge of the data generation elements to retrieve the training data information.

Property inference attack. The property inference attack aims to steal the sensitive privacy attributes of model training data. For example, in a malware classifier model that discriminates between malicious and benign software execution traces, the attacker may want to use this model to learn the attributes of the test environment to avoid detection or recognition. The test environment will affect all trace tracking so it can be regarded as the attribute of the entire training set rather than the attributes of a single data. Another example is that recent studies have found that some special classes of people, such as women and ethnic minorities, may be underrepresented in various training datasets. There are corresponding differences in the performance of common classifiers at all levels. Therefore, it is also an important application scenario of a property inference attack to infer whether the dataset used for training the model has a certain type of preference, which is also one of the global attributes of the dataset.

The principle of the property inference attack is roughly the same as that of the membership inference attack. The difference, however, is that the membership inference attack model aims to classify whether a sample belongs to the training set, while the property inference attack tends to distinguish whether a sample contains a certain feature or belongs to a certain category. In a property inference attack, shadow models are trained on the datasets with or without certain properties. Moreover, the attack model changes from the classification of membership inference attack of whether a sample belongs to the training set to the classification of whether a sample contains/belongs to a certain feature/category. To infer global properties of the training datasets, Ganju et al. [35] first proposed a property inference attack method for **fully connected neural networks (FCNNs)**, which simplifies the training process of property inference attack based on meta classifier [6].

Reconstruction attack. The reconstruction attack is also called the model inversion attack, which attempts to reconstruct the training samples or the corresponding labels rather than determine some abstract information of the dataset. The initial reconstruction attack was based on the assumption that the adversary could access model M, the distribution of features that are a priori known to be sensitive and insensitive for a specific input x, and the output $\hat{y}$ of the model. The attack is based on estimating the value of sensitive features, giving the value of nonsensitive features and output labels [33]. Data reconstruction is obviously a more direct, threatening, and difficult attack method than membership inference and property inference attacks. At present, for different training scenarios, there are many attack schemes of dataset reconstruction that have achieved significant results. For example, in federated learning, the attacker is able to precisely reconstruct the content of the training set by only stealing the communication content from each round of client and server [51, 151, 193].

Model extraction attack. The model extraction attack is a kind of malicious behavior that steals model information. The attacker attempts to access the target model, create a pseudo model with similar functions, or imitate the decision boundary. Stolen models are often of great commercial value or used in security applications and are regarded as confidential. Once the information about the model is leaked, the attacker can avoid paying or opening up third-party services to obtain commercial benefits, which will damage the rights and interests of the model owner. More seriously, if the model is stolen, then the attacker can further deploy a white-box adversarial attack to cheat this online model. At this time, the leakage of the model will greatly increase the success rate of the attack. Specifically, given a carefully selected input sample x, an enemy query target model M (victim model), the corresponding prediction result $\hat{y}$, the opponent can then infer or even extract the whole model M being used. For artificial neural network $y = wx + b$, the model-stealing attack can approximate the values of w and b to some extent. Trame'r et al. [139] took the lead in investigating model theft attacks in 2016 and proposed a model theft method for equation solving.

2.3.2 Privacy Preservation of Machine Learning. To reduce the risk of privacy leakage that may be caused by AI models, including the leakage of data information caused by the update of model parameters in the training stage, the leakage of model and data privacy caused by the return of query results in the testing stage, and the data privacy leakage indirectly caused by the normal use of these AI models, academia and industry have made many efforts from various perspectives. To solve this kind of data leakage, the main idea is to reduce or confuse the effective information contained in this kind of interactive data as much as possible without affecting the effectiveness of the AI model. The following types of data privacy protection measures can be adopted: model structure defense, information confusion defense, and query control defense.

Model structure defense. The model structure defense method deliberately adjusts the training strategy to reduce the sensitivity of the model output predictions corresponding to different samples. Model-oriented defense [117, 127] is to reduce the model information leakage or the degree of overfitting by making appropriate modifications to the model structure to complete the protection of model leakage and data leakage. In addition, some works [103, 156] also combine ML and encryption technology to protect the model's privacy.

Information confusion defense. The information confusion defense method destroys the effective information included in the confused interactive data (e.g., model output and parameter update) as much as possible while ensuring the model's effectiveness. Data-oriented defense involves applying fuzzy operations to the model's input samples or prediction results. Through these fuzzy operations, on the premise of ensuring the correctness of the output results of the AI model, the effective information included in the output results is interfered with, such that the leakage of private information is reduced. These data confusion operations mainly include two types: (1) truncation confusion [63, 127, 144], that is, rounding the model output vector by removing information after a certain decimal point; (2) noise confusion [49, 60], that is, adding small noise to the input samples, the information in the training process, or the output probability vector to interfere with the accurate information.

Query control defense. The query control defense method can prevent data leakage by detecting query operations and rejecting malicious queries in time. Specifically, the defender can extract features according to the user's query behavior and then conduct the defense against privacy disclosure attacks by detecting abnormal queries. To perform a privacy disclosure attack, an attacker needs to launch a large number of queries on the target model and even needs to modify his input vector specifically to speed up the implementation of the privacy disclosure attack. According to the characteristics of user query behavior, it is able to distinguish which users are attackers and then restrict or deny service to the attacker's query behavior to achieve the purpose of defense attacks. Query control defense mainly includes two types: abnormal sample detection [63, 64, 174] and query behavior detection [49]. Generally speaking, according to the characteristics of a model leakage attack, the defender can identify the model-stealing behavior by detecting the query of abnormal samples. According to the characteristics of a data leakage attack, the defender can limit the number of queries in the sample input stage according to the behavior characteristics of user queries.

3 THREATS TO DRL

In this section, we analyze the potential threats to DRL in light of the objectives of the attacks. Attacks that alter or interfere with DRL systems are referred to as security threats, because they affect the availability and integrity of information systems. We refer to risks to confidentiality and privacy in information systems as privacy threats when they involve obtaining information from DRL systems that was not meant to be shared. The taxonomy of threats to the DRL system is illustrated in Figure 2.

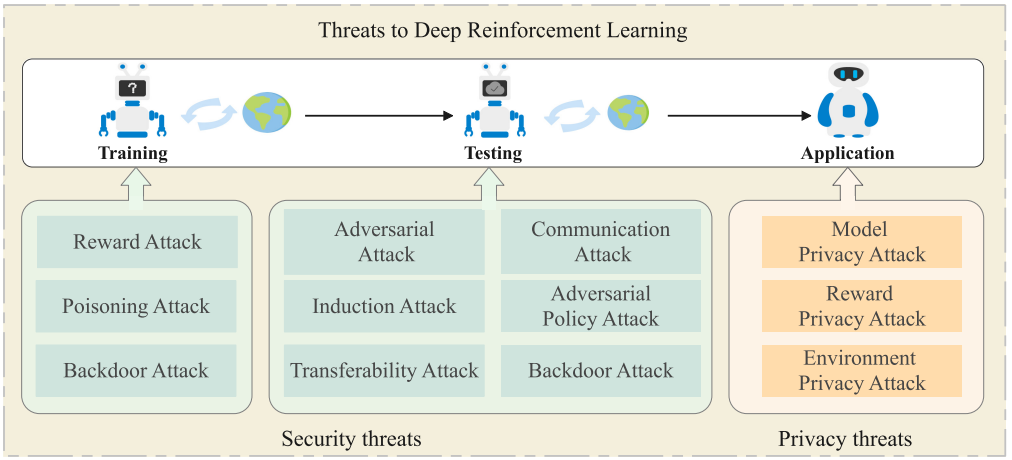


Fig. 2. Illustration of threats to DRL.

3.1 Attack Methods on Single-agent DRL

In this subsection, we elaborate on various attack methods on single-agent DRL and organize them into two categories: training-phase and testing-phase attacks, according to the attack stage. For each category, we categorize the attacks into different types based on the fragile components of the DRL process.

3.1.1 Training-phase Attacks. We divide this subsection according to the target component of the attacker.

Reward attack. As rewards formally characterize an agent's purpose, altering the rewards logically changes the learned policies of the agents. A practical attack method, called reward-poisoning attack, was proposed in batch RL [182, 183], where rewards were stored in an unlocked, pre-collected dataset. This provided the attacker with the opportunity to directly change the reward in the dataset. The trained agent's performance was somewhat negatively affected by this attack, but it also involved some serious assumptions about the adversary's knowledge. Studies [54, 112] have investigated online reward-poisoning attacks in the white-box setting, where the attacker is assumed to have full knowledge of the MDP or the learning algorithm. They investigated the cost of these poisoning attacks with the goal of developing an attack model with the least costly attack.

Rakhsha et al. [112] investigated an approach to secretly alter the rewards or the transition dynamics in the learning environment to compel the agent to carry out a target policy. An optimization framework was proposed to identify various optimal stealthy attacks under different measures of attack cost. According to experimental findings, the attacker might easily use this reward-poisoning attack to teach the victim agent any target policy. Rakhsha et al. [113] and Cai et al. [16] studied the reward-poisoning attack in black-box settings. Their attacks made assumptions about the knowledge of the adversary: (1) no preliminary knowledge of the environment or the learner; and (2) can not observe the learner's internal mechanisms. The attack in Reference [113] was implemented in two phases, i.e., the exploration phase and the attack phase. During the exploration phase, the confidence set M was established to estimate the parameters of environment MDP. These sets served as the exploration phase's "stop sign" and were used to assess whether the amount of data gathered was sufficient for the attack phase. With the help of those sets, this reward-poisoning approach might achieve a level of effectiveness close to the optimal white-box

attack. Cai et al. [16] built a deep neural network to learn the **successor representation (SR)** value for each state, in which the SR implicitly represented the expected discounted sum of access frequencies for each successor state from the current state in the future. Then, the attacker determined the time of the attack based on the SR value. The experimental results demonstrated that this black-box attack algorithm could effectively compromise agents with fewer adversarial examples.

Poisoning attack. The goal of the poisoning attack is to dramatically degrade the DRL agent's performance by manipulating the learned DRL model parameters. Huai et al. [52] designed an attack that manipulated the learned model parameters to degrade the performance of the DRL interpretation methods. They proposed an optimization framework based on which the optimal adversarial attack strategy can be derived. The experimental results demonstrated that interpretations could be successfully misled across various settings. For example, a sub-optimal action taken by an agent, which is actually caused by a top-left corner crafted universal perturbation, will be attributed to the bottom region by a Jacobian interpreter.

Xu et al. [167] studied the environment-dynamics poisoning attacks that seek the minimal environment dynamics manipulation that would prompt the momentary policy of the agent to change in a desired manner. They modeled this problem as a sequence optimization problem and used **Deep Q-learning (DQN)** algorithm to solve this problem. They also studied the transferability of these attacks. Experimental results showed that attack strategies can be trained using a white-box proxy agent and transferred to poison a black-box victim's policy. Xu et al. [166] further studied this problem, assuming that the attacker could only alter the environment hyper-parameters. They proposed a Double-Black-Box EPA framework, which incorporated an inference module to capture the internal information of an unknown RL system to learn an adaptive strategy based on an approximation of our attack objective. They argued that this attack achieved comparable performance to the white-box attack in the grid world.

Backdoor attack. Panagiota et al. [65] studied the backdoor attack in DRL and presented the TrojDRL framework. This framework constructed Trojan-infected states by overlaying triggers on the original states and modified the reward of these Trojan-infected states to launch the backdoor attack. Once the backdoor agent observes any state containing the trigger, it will take the target action, which the attacker determines. They pointed out that deciding when to poison the input data and manipulate the associated rewards is central to the attack's success. Experimental results showed that these attacks require only the poisoning of as little as 0.025% of the training data.

Ashcraft et al. [5] introduced an alternate backdoor attack using in-distribution triggers and training methods for embedding backdoors into RL agents. They claimed that these attack methods are challenging to detect due to the nature of these triggers. Yu et al. [177] exploited the partial state observability of DRL to hide malicious behaviors for backdoors. They proposed a temporal-pattern backdoor attack to DRL, whose trigger is a set of temporal constraints on a sequence of observations. They argued that this backdoor attack could be applied in many real-world DRL applications, since observations in these applications are partial, and our triggers can be easily hidden in unobservable temporal observation.

3.1.2 Testing-phase Attacks. We divided this subsection based on the attacker knowing information about the target agent.

Adversarial attack. It was reported [56] that adversarial examples could deceive a well-trained DRL agent into taking a specific action during the testing phase. However, unlike DNNs, attacking DRL agents not only fools the agent into performing a wrong action but also a series of wrong actions. Huang et al. [53] proposed an initial attack in which the attacker calculated the adversarial perturbations for each state at each step. This type of attack is also known as the uniform attack,

in which the DRL would perform malicious actions during the episode. Lin et al. [77] argued that the uniform attack ignored the relevance of the state in DRL and that frequent attacks likewise increased the likelihood of being detected. The conclusion was that adversarial attacks launched at different steps have different effects and that attacking at particular timesteps was a more dependable strategy. Key point attacks involve targeting specific objectives. Lin et al. [77] proposed a strategically timed attack to minimize the agent's reward with as few attacks as possible. This strategically timed attack could achieve the same effect as the uniform attack by launching four times fewer attacks.

Kos et al. [67] compared attacks launched using random noise with those using adversarial examples generated by FGSM, demonstrating that the FGSM method is much more effective than random noise in deceiving DRL agents. They proposed the concept of using value functions to guide against perturbation injection. They conducted comparative experiments in three different settings and demonstrated that an attacker could use value functions to perform more effective attacks than traditional attack methods.

Lee et al. [71] noted that traditional adversarial attacks primarily focus on manipulating the state space of RL agents, while attacks on the action space of RL agents have received relatively less attention. They proposed a white-box **myopic action space (MAS)** attack algorithm that distributed the attacks over different action space dimensions as a constrained decoupled optimization problem. Based on this, the optimization problem was reconstructed with the same objective function but with a temporal coupling constraint on the attack budget to account for the approximate dynamics of the agent. A white-box **look-ahead action space (LAS)** attack algorithm was also proposed, which distributed the attacks over action and time dimensions. Their experiments demonstrated that LAS attacks had a much larger impact on agent performance than MAS attacks when using the same amount of resources. Through the LAS attack, an adversary could exploit the agent's dynamics to perform malicious attacks with limited resources, ultimately leading to agent failure.

Chan et al. [18] showed that the traditional attack approach somewhat reduced the agents' performance but did not reveal how changes in features affected the cumulative reward obtained by the agents or why certain features are more critical in the decision-making process. They claimed that certain features were inherently more important than others in the decision-making process, providing more critical information. They introduced the concept of a static reward impact map to quantify the impact of each feature on the reward. Based on this, the counter-attack approach targeting DRL to minimize the cumulative reward is proposed. Attackers would likely select features with a more considerable reward impact. Their experiments demonstrated that the counter-attack method targeting DRL, aiming to minimize the cumulative reward, outperformed existing one-time and random attack methods in both white-box and black-box settings, especially in terms of cumulative reward and successful attack rates.

Sun et al. [132] argued that strategically timed attacks only considered the effect of one step of the attack and ignored the impact on subsequent states and operations, for which they introduce two novel adversarial attack techniques that are stealthy and effective against DRL agents. By using these two techniques, the adversary was able to harm the agent the most while injecting adversarial examples at the least critical moments. The first technique was the critical point attack, where the attacker built a model to predict the future state of the environment and the agent's actions, evaluated the damage of each possible attack strategy, and then chose the optimal attack strategy. The second technique was the antagonistic attack, where the attacker automatically learned a domain-agnostic model to discover the critical moment to attack the agent in a given episode. Experimental results demonstrated that the critical point technique required only

1 (for the TORCS) or 2 (for the Atari Pong and Breakout) steps, while the antagonist technique required less than 5 steps (4 for MuJoCo tasks), which was a significant improvement over the state-of-the-art approach.

Induction attack. Intuitively, reward-based attacks can only be applied in the training phase, since the agent's decisions in the testing phase are limited to the trained policy function. Thus, to reduce the agent's performance during testing, the attacker could only focus on some states or actions. The induced assault was proposed as a way to get around this limitation by using the planning algorithm to produce a series of successful attacks that would induce the agent into the specified target state. The induced attack manipulated the reward indirectly, which was the product of combining the planning algorithm with the generative model.

Vahid Behzadan et al. [12] conducted a thorough investigation into the susceptibility of DQN agents to policy-induced attacks. They presented a novel counter-example attack method and undertook a detailed evaluation of several basic DQN architectures, as initially conceptualized by Mnih et al. [89]. Their research provided significant insights, confirming that policy-induced attacks can effectively compromise the performance of RL systems through minimal alterations to the environmental or sensory inputs.

Lin et al. [77] proposed enchanting attacks where the attacker aimed to lure the agent to a designed target state. The generative model predicted the next state according to the current state, while the planning algorithm generated a preferred sequence of actions to lure the agent. Then, a series of adversarial examples was designed to lure the agent into taking the preferred sequence of actions. Their experiments demonstrated that even in the most-used reinforcement learning algorithms, including **Advantage Actor-Critic (A3C)** and DQN, the enchanting attack could achieve a success rate of over 70%.

Edgar Tretschk et al. [140] demonstrated that imposing arbitrary adversarial rewards on a victim policy network through a series of attacks was possible. They used an adversarial attack technique, namely, the **adversarial transformer network (ATN)**, to learn how to generate attacks that were easy to integrate into the policy network. The ATN received input frames, computed perturbations in a feed-forward manner, and then added the perturbations back to the input, fed into the victim's DQN. As a result of the imposed attacks, the victim was misled to optimize the adversarial reward over time.

Weng et al. [153] argued that previous research on RL agents has focused on model-free counter-attacks and agents with discrete actions and mostly made unrealistic assumptions about the attacker's high accessibility to the agent's training information. They studied the problem of adversarial attacks on continuously controlled RL agents using a model-free attack baseline based on random search and heuristics. They proposed a two-step algorithm based on learning model dynamics that could be directly applied to two commonly used threat models, including observation manipulation and action manipulation. The dynamics model predicted the next attacker's desired state. They demonstrated that the proposed framework was more effective and efficient than model-free attack baselines in reducing agent performance and inducing agents into insecure states.

Most existing adversarial attacks target the agent's state space, and Hussenot et al. [55] stated that the requirement for writing access to the agent's internal workings to alter the agent's state directly was too severe in a realistic setting. They proposed an attack, namely, CopyCAT, which was a targeted attack capable of matching the behavior of a neural policy with that of an arbitrary policy, consistently luring an agent to follow an outsider's policy. It was precomputed and easily extrapolated in situations that involved real-time data. In this scenario, the attacker could only change the agent's observation.

Mo et al. [90] further studied and explored the existing attacks and found that the damage metrics or thresholds for the design of the critical point attack and policy time attack heuristics relied

on domain knowledge and manual debugging and the potential of DL technologies may not be fully utilized. What is more, the computation of all possible strategies in the critical point attack and the computation of online perturbations for back-propagation gradients required enormous computational resources. They claimed that the antagonist attack did not fully consider the characteristics of the attack problem, especially the sparse nature of perturbation injection. They proposed a **decoupled adversarial policy (DAP)** to attack the DRL mechanism. This attack decomposed the adversarial policy into two independent sub-policies: (1) switch policy, which decided whether the attacker should initiate an attack, and (2) lure policy, which decided the action the attacker induces the victim to take. The switch policy could attack DRL in real-time scenarios with fewer steps, benefiting from higher efficiency and utility.

Transferability attack. Huang et al. [53] explored the transferability of adversarial examples across policies and training algorithms. They discovered that the effectiveness of adversarial examples is significantly influenced by knowledge of the target policy, including details about its training algorithms and hyperparameters. Transferability across algorithms is less effective at decreasing agent performance than transferability across policies, which is less effective than when the adversary does not need to rely on transferability.

Yang et al. [171] proposed a black-box approach in which an attacker trained policy agents with opposite reward goals to achieve transferability attacks by observing the rewards, the victim DRL network's actions, state, and environmental information. A population-based adversarial policy agent (ASA) based on **parametric exploration policy gradient (PEPG)** was proposed to optimize the black-box system. The PEPG-ASA algorithm could dynamically select sensitive time frames for interference in physical noise patterns and minimize the system's total reward from offline observation of input-output pairs without accessing the actual parameters of a given DRL framework.

Inkawhich et al. [57] studied the attack problem, assuming the attacker cannot interact with the agent, but rather can only eavesdrop on the action and reward signals exchanged between the agent and the environment. They utilized the transferability of adversarial examples and proposed the snooping attack. This attack could drastically decrease the performance of target agents using adversarial examples from other models. These adversarial examples were calculated based on an agent within a similar task. The experimental result showed that even an attacker acting in a threat model with severe constraints could still carry out devastating attacks on the target agent by putting the agent model through a relevant task and using the transferability of adversarial examples.

We have emphasized various attack methods on single-agent DRL, ranging from altering the state space to interfering with the action space. However, in recent years, researchers have also turned their attention to the security and robustness of **large language models (LLM)**, particularly those fine-tuned through **reinforcement learning with human feedback (RLHF)**. These models excel in natural language processing tasks but have shown sensitivity to adversarial attacks. Wang et al. [146] studied the robustness of RLHF across varying stages, perturbation levels, and sizes of LLMs and subsequently demonstrated fluctuations in their robustness. They discovered a decline in model robustness throughout the RLHF process. Some literature [158, 175, 192] pointed out that these LLMs are susceptible to adversarial attacks, leading to erroneous answers. Their discovery raises great importance to improving the RLHF training paradigm to ensure the stability of LLM.

To summarize the attacks proposed against single-agent DRL, Table 1 presents the representative studies and analyses in terms of attack type, the timing of the attack, and the targeted element under attack.

Table 1. Summary of Single-agent Attacks

Attack Method	Attacking phase		Settings		Attacking Element	Representative Studies
	Training	Testing	White-box	Black-box		
Reward Poisoning Attack	•		•		Reward	[54, 112]
Reward Poisoning Attack	•			•	Reward	[16, 113]
Model Poisoning Attack	•		•		Neural network	[52]
Data Poisoning Attack	•		•		Observation	[5]
Uniform Attack		•	•		Observation	[52, 53, 55]
Key Point Attack		•	•		Observation	[67, 77, 115, 171]
Action Space Attack		•	•		Action	[71, 138]
Realistic Attack		•		•	Environment	[160]
Induction Attack		•	•		Policy	[12, 77, 140]
Metastability Attack		•	•		Observation	[53]
Transferability Attack		•		•	Observation	[171] [57]

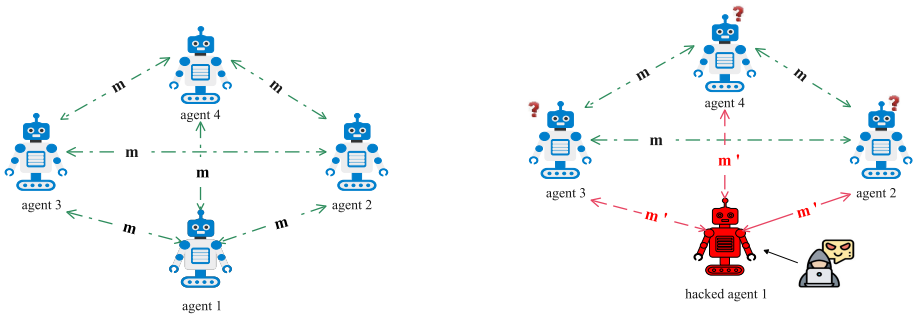
3.2 Attack Methods on Multi-agent DRL

Advances in DRL have achieved remarkable success not only in single-agent scenarios but also in multi-agent domains. Currently, massive multi-agent systems embedded in society are already partly driven by **multi-agent deep reinforcement learning (MADRL)** algorithms. However, the security of MARL, unlike that of MADRL algorithms, has not attracted significant attention and is currently in its infancy. This section provides an overview of the latest research in multi-agent DRL security.

3.2.1 Attack through Communication. Communication among cooperative agents in a partially observable environment enables the exchange of information, facilitating collaborative actions to achieve common objectives. There has been extensive research on communication between agents, including who [61], what [58], and how [186] to communicate. However, if some agents are adversarial and send maliciously designed messages, as described in Figure 3, then multi-agent coordination will rapidly disintegrate as these messages propagate.

Agents will communicate by transmitting feature maps, such as intermediate representations. Tu et al. [141] considered a setting where multiple homogeneous agents with the same neural network perform their tasks by sharing observations from different viewpoints encoded via a learned intermediate representation. Tu et al. claimed that malicious agents can send indistinguishable adversarial messages to benign agents, which will degrade all the agents' performance. An adversarial perturbation generation approach was proposed for computing those adversarial messages.

Another way to communicate is to propagate messages. Agents aggregated incoming messages from others by the communication protocol and then treated them as part of the state. Blumenkamp et al. [15] considered a situation where a self-interested agent may communicate erroneous information to others to increase its own access to the limited rewards. They proposed Self-interested Learning that trained the self-interested agent to maximize its reward regardless of team reward. They trained the self-interested agent with the actor-critic algorithm while other agents' policies were fixed. They analyzed the messages sent through self-interested agents by post



(a) Normal communicative multiple agents (b) Attacked communicative multiple agents

Fig. 3. An example of communication attacks in a communicative MARL system. (a) Multiple agents work well by delivering the right messages m. (b) One of the agents was hacked, and it sent out wrong messages m' to interfere with the cooperation.

hoc interpretability. The results demonstrated that the self-interested agent has learned to send devious encodings in messages to others, in other words, lying about its state and observations to mislead other agents.

Xue et al. [170] proposed an adversarial communication in MACRL. They considered a scenario in which adversarial agents hid among many benign agents, manipulating them to take suboptimal actions by posting malicious messages. Meanwhile, the adversarial agents try to behave like benign agents, hoping not to be detected. They modeled the policy of sending malicious messages as a multivariate Gaussian distribution determined by a DNN. They utilized **Proximal Policy Optimization (PPO)** [122] to optimize this policy to minimize the accumulated team rewards.

3.2.2 Attack through Adversarial Policy. Competitive agents aim to maximize their own individual rewards while minimizing the rewards of their opponents. In recent years, competitive algorithms have emerged, including self-play [10], well-design rewards [80], meta-learning [3], and opponent modeling [111]. What is more, there are some approaches that enable agents to win by exploiting the potential vulnerabilities of opponents cleverly, not by becoming generally strong agents. To avoid confusion, we use an adversarial agent(s) and victim agent(s) to represent agent(s) launching the attack and agent(s) being attacked, respectively.

Gleave et al. [38] found that agents win by falling to the ground in contorted positions rather than running or kicking like normal agents in competitive boxing environments. These weird results show that simpler ones could fool some complex policies. Gleave et al. [38] demonstrated that apparently strong self-play policies still harbor serious but hard-to-find failure modes, and those defects are difficult to eliminate completely. Wu et al. [157] proposed a higher efficiency and less sensitivity framework to find this vulnerability. The key idea of this framework is to maximize the deviation of the actions taken by the opponent agent at valuable steps with minimal observation change. A valuable step is defined as when the victim agent policy network pays sufficient attention to the features corresponding to the adversarial agent's actions. To this end, they utilized an explanation AI technique to check each step's value and adjust the action deviation coefficient in the adversarial agent's objective function.

The above-mentioned approaches rely upon the zero-sum assumption made in the two-player competitive games. Guo et al. [45] extended attacks to non-zero-sum competitive scenarios. The Non-zero-sum assumption is that maximizing the attacker's own reward does not certainly min-

imize the victim's reward. Guo et al. [45] proposed an objective function that could be divided into two terms—original adversarial reward and reward—that impose negative influence upon the victim agent. They argued that this objective function is monotonic in the entire adversarial policy learning process. Experiment results demonstrated that this adversarial had learned more adversarial policy than adversarial agents in Reference [38], such as abusing the unfairness of the target game.

Getting rid of the knowing victim's reward function, Fujimoto et al. [34] studied the effectiveness of the adversarial policy, which only learned from the victim's actions in the cooperative environment. They proposed Reward-free Attacks, whose key idea was maximizing the entropy of the victim's actions by the adversarial policy. To this end, they applied batch RL to learn the adversarial policy, which regarded the empirical entropy of the victim's policy as the reward function. Experiments demonstrated that the adversarial policy caused the victim's behavior to be more erratic and can negatively affect the victim.

3.2.3 Attack through Data Manipulation. In this subsection, we discuss the adverse consequences when the attacker has the ability to manipulate the data, i.e., observations, rewards, and actions.

Lin et al. [75] studied the robustness of cooperative MARL when an attacker manipulates an agent's observation. The attacker manipulated this agent to induce it to take specific actions by adding perturbations. The authors claimed that team reward estimation, non-differentiability of models, measuring the effect of misprediction of an agent, and low-dimensionality of the feature space prevent known attack algorithms applied in such settings efficiently. Lin et al. [75] tackled these challenges by proposing a two-step attack method. The first step was learning an adversarial policy that helped the attacker select adversarial actions using the DQN algorithm. The second step was crafting perturbations that lured the agent into taking adversarial actions by gradient-based targeted adversarial example method, namely, **Iterative target-based FGSM method (itFGSM)** and **Dynamic budget JSMA attack (d-JSMA)**. The authors observed that step 2 sometimes fails to achieve an adversarial action from step 1 when there is a sizable Q-value gap between the adversarial action and the original action. They handled this by adding regularization terms, which measure the Q-value gap in step 1. The experimental results on the StartCraft II multi-agent platform showed that their attack could decrease the reward from 20 to 9.4 and the winning rate from 98.9% to 0%.

Pham et al. [104] took further insight into training adversarial policy and selecting which agent to attack within the cooperative agents. They proposed a model-based adversarial policy training method that forced the victim agents to move closer to a damaging failure state in the next timestep. The decision of which 'vulnerable' agents to attack was modeled as an optimization problem, effectively solved using the PGD algorithm. Under the same experiment settings, this attack's amount of team reward reduction is 42% more than the Lin et al. [75] one.

Wang et al. [148] asserted that attackers could introduce covert triggers into the policies of competitive agents by manipulating the training data and process. These inserted agents would behave normally when the backdoor remained dormant but fail rapidly when the backdoor was activated, akin to a fast-failing policy triggered by the backdoor. The insertion of the backdoor was carried out in two steps: first, training the fast-failing policy, which would exhibit failure within a few steps using adversarial training [38]; second, merging the fast-failing and normal policies through behavior cloning. This backdoor attack was assessed in the MuJoCo environment with victim policies based on **Long Short-Term Memory (LSTM)**. The results demonstrated that upon activation of the backdoor, the winning rate of the victim agents decreased by 17% to 37% compared to when the backdoor was not activated.

Table 2. Summary of Multi-agent Attacks

Attack Method	Settings		Type		Attacking Element	Representative Studies
	White-box	Black-box	Cooperation	Competition		
Communication Attack	•		•		Intermediate feature	[141]
Adversarial Communication attack	•		•		Adversarial message	[15, 170]
Reward-free Attack		•	•		Policy	[34]
Adversarial Policy Attack		•		•	Policy	[38, 45, 157]
Adversarial Attack	•		•		Observation	[44, 75, 104]
Backdoor Attack	•			•	Training data	[148]

Guo et al. [44] proposed attacking the states, rewards, and actions of cooperative multi-agents to test robustness comprehensively. In their state-based attack, they utilized the FGSM method to significantly reduce the probability of the original optimal action, akin to adversarial attacks in supervised learning. This approach led to a drastic decrease in the agents' winning rate, from nearly 100% to less than 15%. The reward-based attack involved inverting the sign of the top k% rewards, misleading the policy during training and effectively reducing the winning rate to almost 0%. Furthermore, in the action-based attack, an agent labeled a 'traitor' was trained via adversarial methods [38] to minimize team reward. Their experiments, conducted on StarCraft II using QMIX and MAPPO algorithms, revealed that the QMIX algorithm exhibited greater robustness than MAPPO, particularly in the context of action-based attacks.

To summarize the up-to-date attacks proposed against multi-agent DRL security, Table 2 presents the representative studies analyzed in terms of settings, type, and the targeted element under attack.

3.3 Privacy Threats to DRL

With the growing utilization of DRL, an increasing number of privacy concerns have surfaced. In response to potential information disclosure, this section categorizes contemporary privacy attacks into three distinct categories: environment, model, and reward privacy attacks.

3.3.1 Environment Privacy Attack. Addressing the data privacy leakage issue in DRL, Pan et al. [101] formulated a problem to infer environmental dynamics in various scenarios. Pan et al. mainly proposed two methods to explore such privacy breaches: environment dynamics search via genetic algorithm and candidate inference based on shadow policies. In the first scenario, the attacker had no prior knowledge of the training environments but made the assumption that these environments adhered to typical constraints. Pan et al. proposed to use a genetic algorithm to recover the original transition map in a Grid World environment. In the second scenario, the attacker has access to a set of potential candidates of environment dynamics. Pan et al. proposed an algorithm to infer which candidate environment is used to train a given policy. Extensive experiments showed that DRL was vulnerable to potential privacy-leaking attacks, and specific information about the training environment dynamics can be recovered with high accuracy.

Additionally, Gomrokchi et al. [42] were the first to study the membership inference attack within the context of DRL. By accessing the off-policy RL model trained on a given input trajectory in a black-box way, the membership of the target input trajectory can be judged according to the output trajectory. It showed that reinforcement learning is more vulnerable to membership inference attacks in collective settings than individual membership inference attacks. To

successfully infer the membership of the target input trajectory, Gomrokchi et al. used the shadow model training technology to obtain the data needed to train the attack classifier. The training dataset of this attack classifier consisted of both the training trajectory and the output trajectory belonging or not belonging to the same training model.

Zhou et al.'s work [189] showed that trajectory analysis could extract information from observations. They proposed a variable information inference attack to steal the observation space of target DRL models. This attack was evaluated in different continuous control environments with various observation spaces, such as length, force, and mass. The experimental results showed that DRL models were vulnerable to privacy inference attacks and that specific information about the training observation could be inferred.

3.3.2 Model Privacy Attack. Model extraction attacks, which seek to reproduce (i.e., steal) an ML model, have drawn much interest from the research community since **ML as a service (MLaaS)** became available.

Inspired by the model extraction attack in supervised ML, the feasibility and impact of model extraction attack on DRL agent have been studied in Reference [11]. The problem of model extraction attack in DRL can be expressed as the replication of DRL policy based on the observation of behavior (i.e., action) in response to changes in the environment (i.e., state). In Reference [11], Behzadan et al. initially demonstrated the feasibility of using imitation learning technology for launching model extraction attacks on DRL agents and developed a proof of concept attacks, which could realize black-box attack on the integrity of DRL policy.

In addition, aiming at the model extraction attack of DRL, Chen et al. [22] first constructed a classifier to reveal the training algorithm family of the target black-box DRL model only according to its predicted actions and then used **generative adversarial imitation learning (GAIL)** to carry out DRL model extraction attack, which trained a discrimination model and a generic model to imitate the behavior of the target DRL policy. The competition between these two models can ensure that replication and target have very similar behavior in the same environment. This method can extract models with high similarity of training algorithm families, behaviors, and performance as the targeted models. A large number of experiments showed that attackers could achieve very high accuracy and fidelity for various tasks and algorithms.

3.3.3 Reward Privacy Attack. In addition to performing privacy attacks on general RL models, some researchers have also studied privacy attacks on privacy-preserving reinforcement learning models. Prakash et al. [108] argued that it is possible to infer the reward functions of privacy-preserving RL agents. They proposed the reward reconstruction attack, which sought to reconstruct the original reward from a privacy-preserving policy using the Inverse RL algorithm. They applied the reward reconstruction attack in various privacy-preserving RL agents, which was enhanced by Bellman update **differential privacy (DP)**, Rényi-DP, and Functional noise DP. Experimental results showed they could clearly identify rewarding blocks within the Frozen-Lake environments under some settings. They claimed that there is a significant gap between the current privacy-preserving offered and the privacy standard needed to protect RL reward functions.

4 DEFENSE METHODS FOR SECURITY-RELATED ATTACKS

In this section, we provide a comprehensive review of the latest defense methods in DRL. We classify existing defense methods into robust learning, adversarial detection, and game-theoretic approach. The defense's taxonomy is shown in Figure 4.

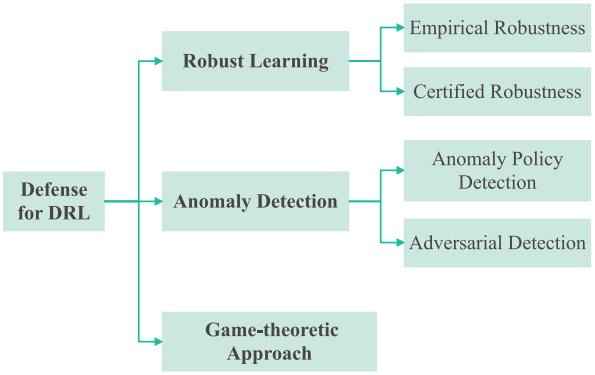


Fig. 4. Taxonomy of the defense.

4.1 Robust Learning

Robust learning, a mechanism to improve the robustness of DL models during the training phase, is structured according to empirical and certified approaches.

4.1.1 Empirical Robustness. Empirical robust learning refers to the use of empirical methods, such as heuristics or evaluations, to ensure the robustness of the model. Mandlekar et al. [85] introduced **Adversarially Robust Policy Learning (ARPL)**, which leverages the active computation of physically plausible adversarial examples during the training phase to enable robust policy learning heuristically under both random and adversarial input perturbations. They pointed out that the addition of adversarial perturbations was not limited to the image area but extended to the entire state of uncertainty in physical parameters such as mass, friction, and inertia. They applied the FGSM algorithm to construct those adversarial perturbations, demonstrating that training with ARPL improved the model's resistance to changes in observations and system dynamics. It must be noted that this approach may degrade the agent's performance in cases without any perturbations.

Tessler et al. [138] considered robustness with respect to alternative adversarial action and added action perturbations. They, respectively, modeled those two criteria for robustness as **probabilistic action robust MDP (PR-MDP)** and **noisy action robust MDP (NR-MDP)**. They solved these MDPs by using a variant of DDPG called Action Robust DDPG. Empirical results in various MuJoCo environments showed that their methods produced robust policies under abrupt perturbations and mass uncertainty. In addition, they argued that these methods might increase the agent's performance in the absence of these perturbations.

Zhou et al. [191] studied the robustness of multi-agent systems under state perturbations. They proposed a robust learning framework for **Mean-Field Actor-Critic Reinforcement Learning (MFAC)**, known as RoMFAC, renowned in the multi-agent field. The key idea of RoMFAC was maximizing the expected cumulative discount reward in the worst case. They tackled this learning problem by re-designing an objective function containing an action loss function representing the difference between actions taken on clean and adversarial states. They also proposed a repetitive regularization method to adjust the weight factor of the action loss. Experimental results conducted on MAgent demonstrated better robustness than naive MFAC algorithms. Moreover, they argued that this method would improve performance even in environments without perturbations.

The aforementioned robust learning considered the robustness of the state or action. From the perspective of reward, Yuan et al. [178] proposed a multi-step state-action value algorithm to over-

come the challenge of reward hacking by using a new return function that alters the discount of future rewards. This approach was evaluated in an austere environment with an effective defense. Wang et al. [147] considered robustness from a susceptible reward perspective: inherent noise, application-specific noise, and adversarial noise. They proposed a robust RL framework that enabled agents to learn in noisy environments where only perturbed rewards are observed. They generalized an unbiased estimator for true rewards. Using Q-Learning as an example, they also guaranteed the convergence to the optimal policy with the acceptable samples needed. Experimental results in Atari showed that their proposed method obtained 84.6% and 80.8% improvements on the average reward when the error rate is 10% and 30% with the PPO algorithm, respectively.

Shen et al. [124] proposed a learning framework, namely, **Smooth Regularized Reinforcement Learning (SR^{2}L)**, which achieved robustness by smoothing the DRL policy. SR^{2}L used a smoothness-inducing regularization to encourage the output of the policy not to change much when injecting small perturbations to the input of the policy. They measured this change in the outputs by Jeffrey's divergence, which added to the policy update, encouraging the policy to be smooth within the neighborhoods of all states on all possible trajectories regarding the sampling policy. They applied this smoothness-inducing regularizer to TRPO [121] and DDPG [74] algorithms. Experimental results showed that SR^{2}L improved robustness against adversarial perturbations to the state, sample efficiency, and training stability.

4.1.2 Certified Robustness. The concept of "certified robustness" originates from supervised learning. A classifier is claimed to be certifiably robust if, for any input x, one will get a guarantee that the classifier's prediction is constant inside some set around x [29]. In RL, the existing work mainly considered the certified robustness from a per-state action and cumulative reward. The former guaranteed that the learned policy would predict the same action before and after attacks under certain conditions. The latter guaranteed a lower bound on the cumulative reward for the learned policy under attacks.

Lütjens et al. [83] proposed an add-on certified defense, namely, **Certified Adversarial Robustness for RL (CARRL)**, to solidify the Q value in case of adversarial attacks. By considering the worst-case perturbations of states, the lower bound of the Q value could be calculated by the method in Reference [184]. Hence, the agent was supposed to take the most robust action, which maximized the lower bound. Experimental results in a collision-avoidance system and a classic control task (Cartpole) showed that the proposed method outperformed nominal DQN on benchmarks with perturbations.

Smirnova et al. [130] proposed a distributionally robust policy iteration approach to prevent the agent from executing sub-optimal actions, which may lead to unsafe states of the system. The proposed approach was based on robust Bellman operators, which provide a lower bound guarantee on policy state values. The experimental results on continuous control tasks showed that distributional robustness is able to improve the stability of training and ensure the safe behavior of RL algorithms.

Zhang et al. [181] modeled the perturbation on state observations as a MDP, which was called **state-adversarial MDP (SA-MDP)**. On this basis, the robustness learning problem can be transformed to find an optimal policy under the optimal adversary. Under the adversary's stationary, deterministic, and Markovian assumptions, they bounded the loss in performance by a theoretically principled robust policy regularizer. It was related to the total variation distance or KL divergence on perturbed policies, which could be applied in various traditional DRL algorithms, including DDPG, PPO, and DQN. Experiments on the MuJoCo environments showed significant robustness improvements compared to vanilla DDPG and PPO under five attacks, which included proposed **Robust Sarsa (RS)** and **Maximal Action Difference (MAD)** Attacks.

Furthermore, Zhang et al. [180] proposed a framework of **alternating training with learned adversaries (ATLA)**, which trains an adversary online together with the agent using a policy gradient following the optimal adversarial attack framework. They applied DRL algorithms to train this adversary agent to minimize the victim's expected cumulative reward. Experimental results showed that this method is more robust than SA-MDP [181]. In addition, they claimed that this orthogonal approach could be combined with state-adversarial regularization to achieve SOTA robustness under strong adversarial attacks. Tuomas et al. [98] proposed a framework to train RL agents with improved robustness against L_p-norm bounded adversarial attacks. They modified the standard loss function of different RL algorithms, adding adversarial loss functions by leveraging existing neural network robustness formal verification bounds to certify the robustness. They reported that this framework's total runtime only takes 17 hours compared to SA-MDP's 35 hours on the same hardware.

Sun et al. [133] proposed a certifiable defense to guarantee the robustness of multiple communicative agents, in which no more than half of the agents' communications were attacked. They considered the attacker might arbitrarily change the communications of victim agents instead of the alterations being subject to some constraint, i.e., L_2 or L_∞ ball. To eliminate attacked communications, a message-ensemble policy called **Ablated Message Ensemble (AME)**, which aggregated multiple randomly ablated message sets, was proposed to enable agents to make decisions based on the consensus of the benign messages. The lower bound of cumulative reward was certificated in both discrete and continuous action space. Experimental results showed that AME maintained its performance under various communication perturbations.

Focusing on Q-learning, Wu et al. [155] defined criteria of robustness certification from the perspective of each state's action and lower bound of cumulative reward. The robustness certification for the state's action is under maximum perturbation magnitude $\bar{\epsilon}$, i.e., for any perturbation within $\mathcal{B}^{\bar{\epsilon}}$, the predicted action will not change under any perturbation. The robustness certification for cumulative reward is the lower bound of perturbed cumulative reward under perturbations that were applied to all timesteps. They certified the per-state action and cumulative reward by action-value functional smoothing and global smoothing on the state trajectory, respectively. They applied these certification algorithms to certify nine RL methods, including empirical and certified methods in Atari games, control environments, and autonomous driving environments. Experimental results showed that SA-MDP [181] achieved high certified robustness. In addition, a large smoothing variance could improve certified robustness significantly in the autonomous driving environment. Compare with the standard testing process, the runtime of this framework takes 20–50 times longer.

4.2 Anomaly Detection

Anomaly detection aims to detect adversarial events that do not conform to an expected policy or cumulative reward.

4.2.1 Anomaly Policy Detection. To mitigate the impacts of adversarial state attacks, J. Havens et al. [119] introduced a **Meta-learned Advantage Hierarchy (MLAH)** framework to detect corrupted states via a supervisory master agent by leveraging the advantage function. In MLAH, there were two sub-policies, nominal and adversarial, which were separated and learned online. The master agent monitored these two sub-policies and decided whether to switch the acting sub-policy by measuring the returns of the sub-policies within certain timesteps. Thus, the supervisory master agent could detect the presence of adversarial examples due to unexpected sub-policies. Lee et al. [72] conducted grid-world experiments to better understand MLAH capabilities and limitations. They found that the master agent could select a sub-policy that learned to map an

adversarial observation to action that led to nominal rewards. The sub-policies themselves could adapt when the master policy fails to select the sub-policy.

Another defense method is to embed the watermark into the agent policy during training and detect the existence of the watermark in the test phase to determine whether there is an attack. Chen et al. [21] introduced a temporal-based watermarking methodology for DRL policies. They introduced damage-free states, from which the DRL system can still be safe and reliable when there was a deviation of action probability, designed an algorithm to identify such states, and used them for the watermarks. They utilized statistical tests of action probability distribution to verify the ownership of the target model with only black-box accesses. Experimental results revealed that this watermarking scheme could satisfy functionality, state, and damage-free requirements under different environments and system settings.

4.2.2 Adversarial Detection. Lin et al. [78] pointed out that the adversarial examples detection methods developed in the context of image classification would ignore the temporal coherence of multiple frames. Hence, they proposed an encoder-decoder visual foresight module that leveraged the temporal coherence of multiple previous frames and the executed actions to predict the next frame. By computing the similarity between the expected action distributions of predicted and observed observations, the adversarial example will be determined if the similarity fails to meet the pre-defined conditions. They argued that this defense method enabled agents to maneuver under adversarial attacks. Xiang et al. [159] proposed an adversarial examples detector for path-finding tasks. They calculated the probabilities of interference points by a **principal component analysis (PCA)**-based model, predicting the adversarial examples by selectively adding interference points based on their probabilities. Experimental results on a simple grid environment concerning Q-learning showed that their method gained a precision of around 70% detection. Similarly, Hickling et al. [50] developed adversarial example detectors within the supervised learning paradigm, utilizing insights from explainable DRL methods. They applied the DeepSHAP algorithm to a well-trained DRL model to generate individual input-level explanations. Then, they generated adversarial examples and mixed them with normal states with calculated SHAP values to build the detector training datasets. They claimed the LSTM-based detector achieved an accuracy of 91% with much faster computing times compared to the 80% accuracy CNN-based detector.

Xiong et al. [165] enhanced the robustness of DRL by a detect-and-denoise schema. A detector was proposed to identify anomalous observations generated in an adversarial way while a denoiser processed these observations to reverse the effect of the attacks. The detector and denoiser were modeled with Gated Recurrent Unit variational Auto-Encoders. They argued that this schema was trained with offline data augmentation to satisfy it in non-adversarial environments. The experimental results on five continuous control tasks with respect to PPO policies and TD3 policies showed better robustness than Zhang et al. [180]. They reported the worst accuracy of the detectors was 95%.

Xue et al. [170] proposed a two-stage defense scheme against the communication attack between agents. An anomaly detector was proposed to detect messages that were likely to be malicious. If so, then the malicious messages would be recovered before distributing them to the corresponding agents by a message reconstructor. The training strategy of the anomaly detector and the message reconstructor was formalized as a sequential decision-making problem, which was solved by the PPO algorithm. They tested this defense method on some communicative environments, including Predator Prey, Traffic Junction, and StarCraft II, in which agents' behavior and attack policies were frozen. Experiments showed that this method could effectively restore agent communication under attack.

4.3 Game-theoretic Approach

To learn a robust policy, Pinto et al. [106] introduced **robust adversarial reinforcement learning (RARL)**, where an agent is trained to operate in the presence of a destabilizing adversary that applies disturbance forces to the system. They formulated policy learning as a zero-sum minimax objective function. The adversarial agent, incentivized by specially designed rewards, aimed to identify state space trajectories that resulted in the worst rewards. Experimental results showed that trajectories with the worst rewards lead to a more robust control policy.

In the work of Reference [97], for designing robust policies, Ogunmolu et al. proposed an iterative minimax dynamic game framework with adversarial inputs, such as disturbance or uncertainties. They quantified a given policy's robustness by challenging it with disturbing inputs, allowing the measurement of robustness based on the degradation of the policy's performance. They evaluated their proposed framework on a Mecanum-Wheeled robot, and this agent aimed to find a locally robust optimal multistage policy that achieves a given goal-reaching task. They claimed that this framework was simple and adaptable for designing meta-learning/deep policies that are robust against disturbances.

Xue et al. [170] enhanced the robustness of the aforementioned defense scheme under the situation that the attacker is aware of the defense. They formulated the attack-defense adversarial communication problem as a two-player zero-sum game and then applied the **Policy Space Response Oracle (PSRO)** algorithm to approach Nash equilibrium. Experimental results demonstrated that this game-theoretic robustness training approach consistently outperformed the conventional method across all algorithms and environments.

Table 3 summarizes the security-related defenses. We utilize abbreviations[1] to signify the environments in which these defenses were evaluated. We can see that most of the defenses were evaluated in the MuJoCo and Atari environments. However, the settings and attacks considered by these defenses are not exactly the same. Therefore, there is still a need for representative evaluation environments to adequately assess common defense methods.

5 PRIVACY-PRESERVING METHODS

In this section, we first present the concept of privacy protection in RL, followed by rigorous RL privacy protection techniques, and finally discuss the privacy protection schemes suitable for DRL. Based on the characteristics of privacy-preserving approaches, the existing privacy defense technologies can be summarized into the following categories: differential privacy, cryptography methods, and policy confusion.

It is important to note that existing privacy-preserving efforts in ML have primarily focused on safeguarding the privacy of training data, the learned model, inputs to the model, and the model's output. In the context of RL, privacy concerns can be described as follows: (1) pieces or trajectories of training data composed of s_t, a_t, r, and s_{t+1}, where a complete episode naturally reveals more private information; (2) environmental information, including the transition function and settings; (3) training settings, including the algorithms of the RL and the hyperparameters of the RL's model; (4) the policy function or value function of the trained agent; (4) the inputs and outputs of the DRL's model. Note that the privacy information reflected by these elements is different in various tasks and agents.

[1]The abbreviation listed under the "Environment" column:
Continuous action space. H: Hopper, HC: Half Cheetah, W: Walker, A: Ant, P: Pendulum, IP: Inverted Pendulum, S: Swimmer, HU: Humanoid, MWR: Mecanum Wheeled Robot, FC: Food Collector, IM: Inventory Manager.
Disperse action space. B: Battle, Pur: Pursuit, M: Mappy, MC: Mountain Car, Pon: Pong, PP: Predator Prey, TJ: Traffic Junction, SC: StarCraft II, CP: CartPole, LL: Lunar Lander, Fr: Freeway, Br: Breakout.

Table 3. Summary of Security-related Defenses

Defense	Protect element	Against	Motivation	Environment[1]
APRL [85]	Observation	FGSM	Training the policy with known attack	H, HC, W, IP
PR(NP)-MDP [138]	Action	Noise	Formulating the criteria of robustness and incorporating it as an integral component of policy optimization	H, HC, W, IP
RoMFAC [191]	Observation	PGD	Maximizing the expected cumulative reward under the worst case	B, Pur
Expected n-step SARSA [178]	Reward	Reward hacking	Altering the discount of future rewards to select actions	M, MC
Reward Robust RL [147]	Reward	Reward hacking	Estimating a reward confusion matrix and defining a set of unbiased surrogate rewards	CP, P, Pon
SR^{2}L [124]	Observation	Random disturbance	Enhancing the stabilization of policies against potential attacks	S, HP, H, W, A
CARRL [83]	Observation	FGST	Computing the assured lower bounds of performance under worst-case scenarios	CP
SA-MDP [181], ATLA [180]	Observation	RS, MAD Attack	Finding the optimal policy under the optimal adversary	H, W, A, HU, IP
AME [133]	Communication	Adversarial communication	Taking actions based on the consensus derived from the benign messages	FC, IM
COPA [155]	Offline data	Poisoning attack	Certifying the number of poisoning trajectories	Fr, Br
MLAH [119]	Observation	Stochastic attack	Detecting corrupted states via a supervisory master agent	IP, MC, H
Watermarking [21]	Ownership verification	Model stealing	Incorporating the watermark into the policy	CP, LL
Detection and Denoising [165]	Observation	Opposite attack	Identifying and reversing the anomalous observations	H, W, HU, A, HC
R-MACRL [170]	Communication	Adversarial Communication	Identifying the anomalous messages and reversing	PP, TJ, SC
RARL [106]	Adversarial disturbances	Adversarial Policy	Formulating the policy learning as a zero-sum, minimax objective function	H, W, HU, A, IP
iDG [97]	Observation	Mismatching	Enhancing policy stability against uncertainty	MWR

5.1 Differential Privacy

Starting with Reference [142], a straight line of work gives RL algorithms verifiable privacy guarantees and performance bounds. Many studies investigated an episodic RL setting in which an agent interacts with K users who come sequentially over K episodes. The agent interacts with

user k over a fixed horizon of H timesteps in each episode k. At each timestep in the episode, the current user exposes their state to the agent, and the agent then suggests an action to take, which creates a reward for the user. The agent's objective is twofold: to maximize the cumulative reward across all users and to minimize the regret relative to the optimal policy.

However, each user's state and reward sequence may contain sensitive information. Although users might be willing to share such information with the agent in exchange for services or recommendations, there remains a risk of inadvertent disclosure of a user's private information when interacting with other users.

In an episodic RL setting, K users arrive sequentially, where each user's sensitive data corresponds to the sequence of states and rewards experienced in a single episode. While focusing on the central model, users are willing to share sensitive information with a trusted agent in exchange for a service or advice, but they do not want their information to be leaked to third parties. Consequently, one important goal is to avoid inference about the user's information while engaging with the RL agent.

5.1.1 Methods for Conventional RL. **Differential privacy (DP)** [32] is a widely used method for preserving data privacy, offering protection for user data. According to the (ϵ,δ)-DP definition, it formally guarantees that details about the data used to train a model can not be deduced from the model's parameters or predictions. These methods cannot be simply transferred to RL, because DP assumes that data are **independently and identically distributed (i.i.d.)**, which does not hold in RL due to the time-series nature of the data.

Joint differential privacy (JDP) is a variation of DP that has been adapted for RL settings. It considers the entire sequence of interactions (state, action, reward) between the agent and the environment and applies privacy-preserving noise throughout the process to ensure that no single interaction can be inferred from the learned policy. JDP thus provides a more suitable privacy-preserving framework for RL. JDP ensures that for any given user u, the information exposed to all other users does not reveal significant details about u's private data. This means that even if all other users collude (for example, by jointly analyzing the agent's policies) to uncover information about user u, the privacy of u's data will still remain secure.

Private episodic RL was initially explored in Reference [142], where the first private algorithm for regret minimization in finite-horizon problems was introduced. They proposed the **Private Upper Confidence Bound algorithm (PUCB)**, a JDP algorithm that guarantees both PAC and regret. In other words, PUCB can be seen as a private version of the UBEV algorithm [30], which obtained the policy using an optimistic strategy and provided both PAC and regret guarantees. They demonstrated that, under JDP constraints, regret increases solely by an additive term that is logarithmic in the number of episodes. As the first work on RL with differential privacy, they considered the well-studied tabular setting.

In a JDP setting, the agent can directly observe user states and trajectories carrying sensitive data, implying that the data is not confidential and may be leaked. **Local differential privacy (LDP)** is a more stringent concept of privacy that requires the user's data to be safeguarded during the collection period before the agent gains access to it. For stronger privacy guarantees, Garcelon et al. [36] developed the first LDP algorithm for regret minimization in RL, named **Local Differential Private Optimistic Backward Induction (LDP-OBI)**. Specifically, this algorithm used a general privacy-preserving mechanism to perturb information associated with each trajectory.

Expanding beyond the previously discussed tabular setting, Chowdhury et al. [28] investigated a specific non-tabular RL problem—the adaptive control of **linear quadratic (LQ)** systems. In this case, the state transition was a linear function, and the immediate reward was a quadratic function of the current state and action. To achieve regret minimization in this issue, they presented the

Table 4. Overview of Differential Privacy Applications in Regret Minimization Research

Reference	Year	DP		Tabular	Function Approximation	Value-based	Policy-based
		JDP	LDP				
Vietri et al. [142]	2020	•		•		•	
Garcelon et al. [36]	2021		•			•	
Chowdhury et al. [27]	2021	•	•	•		•	•
Liao et al. [73]	2021		•		•		
Zhou et al. [190]	2022	•			•		
Luyo et al. [84]	2021	•	•		•		

Private-OFU-RL algorithm, which extended the binary counting mechanism to guarantee differential privacy.

The methods mentioned in References [36, 142] could be categorized as value-based algorithms, and comparable performance was obtained for policy-based algorithms in tabular episodic RL under the central and local models [27].

Recent work by Chowdhury et al. [27] took the first steps toward private policy optimization in the tabular setting. They introduced a general framework called PRIVATE-UCB-PO for constructing private policy-based optimistic RL algorithms and assisting in the establishment of the regret bounds for PO under both JDP and LDP requirements. Additionally, Chowdhury et al. [27] revisited private optimistic value-iteration in tabular MDPs by presenting a generic framework, PRIVATE-UCB-VI, which used a unified analysis method to enhance the existing regret bounds under both JDP and LDP constraints. Another significant conclusion was that, compared to non-private regret, the cost of the JDP guarantee was simply a lower-order additive term, whereas the cost of LDP suffering was multiplicative and of the same order.

Table 4 presents the representative studies to summarize the aforementioned privacy-preserving methods for regret minimization in finite-horizon problems.

5.1.2 Methods for Function Approximation-based RL. Conventional tabular RL algorithms become computationally inefficient or intractable when the state and action spaces are huge or infinite. To get over this restriction, modern RL algorithms with function approximation are presented, which commonly use feature mappings to convert states and actions to a low-dimensional space. Recently, some efforts [73, 84, 190] investigated the private episodic RL with linear function approximation under both central and local models.

Liao et al. [73] proposed a locally differentially private technique for learning linear mixture MDPs, in which the transition probability kernel is a linear function of a specified d-dimensional feature mapping over the state-action-next-state triple. The key idea is to inject Gaussian noise into private data in the UCRL-VTR backbone [7]. From this vantage point, this algorithm can be viewed as a private variation of UCRL-VTR.

In addition to LDP, Zhou [190] explored MDPs with linear function approximation, particularly linear mixture MDPs, within the context of **joint differential privacy (JDP)**. In particular, this article developed two private RL algorithms based on value iteration and policy optimization, respectively, demonstrating that they both achieved sub-linear regret performance and privacy protection.

Luyo et al. [84] addressed continuous sequential decision-making tasks using linear parametric representations and introduced a novel approach that provides efficient guarantees for exploration while preserving privacy. Regarding linear mixture MDPs, they provided a unified framework that enables the analysis of joint and local DP and the demonstration of regret bounds for both LDP and JDP.

Ngo et al. [94] observed that the approach used in Reference [84] involves a polynomial number of triggered updates, with a non-adaptive update schedule. Consequently, Ngo et al. enhanced regret bound by updating its underlying policy whenever it noticed a significant change in the acquired data. With this approach, the amount of noise required to maintain the same privacy parameters is significantly reduced, enhancing regret. In practical applications, Reference [94] could be viewed as a privacy-enhanced variant of the **Least-Squares Value Iterations (LSVI)** offering a desirable accuracy-privacy tradeoff.

Protecting the reward function. There is a field of research on algorithms that safeguard privacy by protecting the reward function [9, 145]. Balle et al. [9] proposed the first private algorithm for policy evaluation with linear function approximation. Regarding the setup, they considered a one-step MDP on which the value function was trained. Based on the first visit to Monte Carlo estimation, they presented two differential privacy policy evaluation algorithms, with output perturbation serving as the privacy assurance; to put it more precisely, first, executing an existing (non-private) least-squares policy and producing a real-valued vector, then adding random noise to each element of the vector. Moreover, utility guarantees for this technique showed that when training batches grew more prominent, the cost of privacy decreased. However, when a new state is queried, the privacy guarantee will no longer be valid, because the private value function was trained using a fixed set of trajectories.

To mitigate the issue in Reference [9], Wang et al. [145] developed a rigorous and efficient algorithm for differentially private Q-learning in continuous state space. They utilized the Gaussian process technique [46], which introduces functional noise into the value function approximation, so the function can be evaluated at an infinite number of states while preserving privacy. In other words, even when new states are accessed, the value function is protected after each update. They also acquire insights into the utility analysis by demonstrating the utility guarantee in tractable discrete state space scenarios. The experiments support their theoretical conclusions and demonstrate how they outperform current approaches. Under similar settings, Abahussein et al. [2] considered privacy-preserving in DRL with Double Deep-Q-Network in continuous space. Concretely, they injected noise into the gradient through a differentially private SGD technique.

Protecting transition probabilities. In addition to those already stated, some studies concentrate on preserving the transition probabilities [40, 48]. The primary distinction is that transition probabilities are members of the probability simplex and must be handled differently to maintain their non-negative nature and sum to one nature. This means that several widely used DP mechanisms, such as the Laplace and Gaussian processes, cannot be used to protect transition probabilities. Recently, the Dirichlet mechanism [41] has been used to overcome this problem, which was subsequently introduced into MDP [40]. Gohari et al. [40] proposed a policy synthesis algorithm that preserves the privacy of the transition probabilities in a Markov decision process. Initially, the algorithm utilized the Dirichlet mechanism to perturb the transition probabilities. Then, based on the privatized transition probabilities, it synthesized a policy using dynamic programming.

The method in Reference [40] works with conventional MDPs, i.e., it searches through all actions for each future state, which is computationally demanding because of the exponential growth of future states. To address this problem, it is possible to convert a standard MDP into a **linearly solvable MDP (LS-MDP)** under suitable action costs. Additionally, LS-MDPs can be approximated to conventional MDPs with accuracy, since they are reduced to linear eigenvalue problems that can

be resolved analytically using an optimal policy. Based on this idea, Hassan et al. [48] introduced DP in an LS-MDP framework, which is used to preserve user privacy by injecting noise from a specific Dirichlet distribution to private default transition probabilities.

Protecting the data transfer. The performance of RL on one task can be substantially improved by leveraging information (e.g., via pre-training) on other related tasks [39, 70]. Lebensold et al. [70] put out a method for transferring knowledge in scenarios where agent trajectories involve private or sensitive data, such as in the healthcare industry. Their method used a differentially private policy evaluation algorithm to initialize an actor-critic model and boost learning efficiency in downstream tasks. Their research showed that this method boosts sample effectiveness in resource-constrained control problems while protecting the confidentiality of trajectory data gathered in upstream activity. In fact, Reference [70] can be viewed as Reference [9] expanded to an actor-critic algorithm with differentially private critics.

The algorithm in Reference [70] can be seen as providing a differentially private critic, while Reference [123] presented a differentially private actor. Seo et al. [123] claimed that an actor that records and changes its activities needs more information than a critic who merely evaluates policies. Therefore, if RL uses confidential and sensitive data, then it must be ensured that the actor's parameters and eligibility trace are kept a secret throughout training. To achieve this, Seo et al. presented a technique to safeguard the confidentiality of private information pertaining to an actor and its eligibility track while training in the actor-critic approach. To be more precise, they leveraged the gradient perturbation method introduced by Reference [1] to the off-policy actor-critic [31] with some modification.

In addition, there are some researches focused on private offline RL [109], private off-policy RL [164] and private distributed RL [99, 107].

Offline RL utilizes a static dataset to learn a nearly optimal strategy in an unknowable environment. Qiao et al. [109] presented the first efficient algorithms for offline RL with differential privacy. Concretely, they designed two novel pessimism-based algorithms, DP-APVI and DP-VAPVI, one for the tabular situation (finite states and actions) and the other for the case with linear function approximation (under linear MDP assumption). Instance-dependent learning boundaries and DP guarantee (pure DP or zCDP) are characteristics shared by both algorithms, and the cost of privacy is expressed as lower-order terms. According to their theory and simulation, for a medium-sized dataset, the privacy guarantee has no negative impact on utility compared to the non-private equivalent.

To create a robust policy that works well in distributed private environments, Ono et al. [99] studied locally differentially private RL algorithms. Specifically, local agents update the model while interacting with their environment, reporting noisy gradients that are created to satisfy LDP, which provides a strict local privacy guarantee. Then, a central aggregator updates its model and distributes it to several local agents using a collection of reported noisy gradients.

5.2 Cryptography

Sakuma et al. [116] argued that in distributed RL, agent perceptions, including state, reward, and actions, were not only distributed but also private. To safeguard the data privacy of each agent, the authors employed an additional homomorphic cryptosystem to create and implement a cryptographic solution that includes an optimal policy.

Liu et al. [81] explored RL applied to **dynamic treatment regions (DTRs)** and implemented personalized treatment by providing a series of treatment strategies for individual patients' time-varying clinical status. However, the health status of patients was compassionate in the process of dynamic strategy-making. Therefore, Liu et al. proposed a privacy protection RL framework, namely, Preyer, which can spontaneously formulate patient-centered treatment strategies and

protect the privacy of patients' health status and treatment decisions. The Preyer framework was deployed in the cloud computing environment. To ensure the security of cloud storage, Liu et al. designed a new encrypted data storage format, which could effectively implement secure non-integer processing across multiple encryption domains, and a new secure plaintext length control protocol to reduce the length of the original plaintext and alleviate the problem of plaintext overflow. In addition, considering the delay of RL incremental update, Liu et al. proposed a secure greedy algorithm and a secure Q-learning algorithm based on experience playback for secure RL.

Park et al. [102] considered the scenario of using RL technology for cloud computing services. Due to the need to exchange privacy-related user data between users and cloud computing platforms for RL-based services, serious data privacy problems may occur. Therefore, the authors considered using a **homomorphic encryption (HE)** scheme and proposed a **privacy-preserving reinforcement learning (PPRL)** framework for cloud computing platforms, which enables cloud computing platforms to carry out RL without decrypting the ciphertext. To ensure more efficient communication overhead, the author proposed a **secure, centralized computing PPRL (SCC-PPRL)** algorithm using a **full homomorphic encryption (FHE)** scheme in the cloud computing environment. The algorithm can limit the error growth of the FHE scheme without a bootstrap algorithm by eliminating the error growth in the iterative Q-value calculation process. By applying RSA encryption to the data exchange encrypted with a shared FHE key in the multi-user cloud environment, the confidentiality between users sharing an FHE key is ensured.

The client of the cloud computing system can not eliminate the information abuse or disclosure problem. Relevant research work uses the idea of **Secure Multi-Party Computation (SMC)** to distribute learning data on multiple servers to alleviate the privacy disclosure problem in cloud computing systems. Considering the problem of how to realize privacy security in a cloud computing system with RL whose learning data is not so obvious, Miyajima et al. [88] proposed an SMC algorithm for Q-learning in the simulation model and verified its effectiveness through numerical simulation.

In addition, considering the malicious third party stealing data privacy during RL training, Jesu et al. [59] developed a simple extended MDP framework, which provides state encryption. Jesu et al. conducted empirical research on different encryption schemes in two environments. Concretely, they reported an early experimental study of the performance of a DQN agent trained on encrypted states in discrete and continuous state space contexts. The findings show that even in the presence of non-deterministic encryption, the agent can still learn in small state spaces, but performance deteriorates in more complicated settings.

5.3 Policy Confusion

Instead of learning the optimal policy, learning a sub-optimal policy that is enough for the task is an effective privacy-preserving measure. This policy, namely, confused policy, may provide meaningless information to an external observer, increasing the difficulty of conjuring private information.

To protect against external observers cloning our proprietary policies, Zhan et al. [179] trained an ensemble of near-optimal policies to prevent an external observer from using behavior cloning. The evaluated results in grid-world experiment showed that adversarial behavior is not feasible in this method, while this ensemble policy achieved a high reward.

To protect the privacy of the reward function, Liu et al. [82] defined a deceptive RL problem, which made it difficult for the observer to determine the truth of the reward function in the reward function set when forming the policy of maximizing the expected return over trajectories. To model the observer's intention recognition, the author proposed two methods: one is based on ambiguity, in which the agent selected actions that maximize the entropy from the observer's point

of view; the other was based on Masters and Sardina's model [86] for intention recognition using irrationality, which took action selection as a weighted sum of honest and "irrational" behavior. These methods used pre-trained Q functions. Since Q functions provide measures of the expected future rewards for each action, they make general representations of the likelihood of action selection. The experimental results show that the resulting policy is deceptive, and the participants' decision on the real reward function is less reliable than that of honest agents.

6 OPEN ISSUES AND RESEARCH CHALLENGES

In the previous sections, we discussed and presented a comprehensive review of security and privacy issues in DRL. Although the threats mentioned above and countermeasures have made solid progress, there are still some open issues and challenges.

Comprehensive attack and defense studies for multi-agent DRL. Compared to single-agent systems, multi-agent DRL presents an extensive opportunity for further progress in understanding its security aspects. The security concerns primarily reside in the training design and implementation mechanism. Mainstream mechanisms such as **Centralized Training Centralized Execution (CTCE)** and **Distributed Training Decentralized Execution (DTDE)** do not inherently factor in security issues. When agents fail to perform as anticipated, the blame [17, 185] often falls on non-stationarity or scalability, potentially overlooking potential attacks. We currently lack a comprehensive understanding of the implications of a single agent or a part of agents being compromised. Consequently, it is crucial to deepen our understanding of multi-agent DRL security.

Comprehensive privacy evaluations for DRL. The exploration of privacy in DRL is still in its early stages, with most existing solutions focusing solely on the privacy issues in RL. It is essential to scrutinize privacy issues specific to DRL and the transferability of privacy attacks and preservation techniques in ML [19]. Additionally, considering the close relationship between security and privacy [169], we must adopt a holistic approach to integrate the privacy and security aspects of DRL.

Research on various evaluation metrics to benchmark different work. Most existing methods in DRL predominantly use rewards as the evaluation metric. While this provides a straightforward measure of an algorithm's success, it may only partially encapsulate some performance facets, particularly concerning security, generalization [69], and portability [143, 188].

Research on certified robustness to end the attack-defense arms race. Numerous adversarial attacks developed later have been able to circumvent or bypass existing defenses. Nevertheless, certified robust learning methods [24, 125] demonstrate the possibility of ceasing the arms race between attack and defense under specific conditions. It is urgent to address the limitations of current certified robustness technologies, including its relatively high computational complexity and its limited functionality in particular function approximation architectures or perturbations.

7 CONCLUSION

In this survey, we presented a detailed classification of DRL security and privacy threats. We have showed various defense strategies and privacy-preserving approaches to counter these threats. Our survey reveals critical areas of vulnerability within DRL systems and identifies the most effective measures currently available to mitigate these risks. Furthermore, we have identified several open issues and potential future directions for research in this field. As the community continues to delve into DRL systems' security and privacy aspects, we hope this survey serves as a guiding resource for interested readers and researchers, stimulating further exploration and innovation in this rapidly evolving topic.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*. 308–318.

[2] Suleiman Abahussein, Zishuo Cheng, Tianqing Zhu, Dayong Ye, and Wanlei Zhou. 2022. Privacy-preserving in double deep-Q-network with differential privacy in continuous spaces. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 15–26.

[3] Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. 2017. Continuous adaptation via meta-learning in nonstationary and competitive environments. *Learning* (2017).

[4] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. 2020. Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* 39, 1 (2020), 3–20.

[5] Chace Ashcraft and Kiran Karra. 2021. Poisoning deep reinforcement learning agents with in-distribution triggers. *arXiv: Learning* (2021).

[6] Giuseppe Ateniese, Giovanni Felici, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, and Domenico Vitali. 2013. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *arXiv preprint arXiv:1306.4447* (2013).

[7] Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. 2020. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*. PMLR, 463–474.

[8] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862* (2022).

[9] Borja Balle, Maziar Gomrokchi, and Doina Precup. 2016. Differentially private policy evaluation. In *International Conference on Machine Learning*. PMLR, 2130–2138.

[10] Trapit Bansal, Jakub Pachocki, Szymon Sidor, Ilya Sutskever, and Igor Mordatch. 2018. Emergent complexity via multi-agent competition. In *International Conference on Learning Representations*.

[11] Vahid Behzadan and William Hsu. 2019. Adversarial exploitation of policy imitation. *arXiv preprint arXiv:1906.01121* (2019).

[12] Vahid Behzadan and Arslan Munir. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. *Mach. Learn. Data Min. Pattern Recog.* (2017).

[13] Vahid Behzadan and Arslan Munir. 2018. The faults in our pi stars: Security issues and open challenges in deep reinforcement learning. *arXiv: Learning* (2018).

[14] Richard Bellman. 1952. On the theory of dynamic programming. *ProcNat'l Acad. Sci.* 38, 8 (1952), 716–719.

[15] Jan Blumenkamp and Amanda Prorok. 2020. The emergence of adversarial communication in multi-agent reinforcement learning. In *Conference on Robot Learning*.

[16] Kanting Cai, Xiangbin Zhu, and Zhao-Long Hu. 2022. Black-box reward attacks against deep reinforcement learning based on successor representation.

[17] Lorenzo Canese, Gian Carlo Cardarilli, Luca Di Nunzio, Rocco Fazzolari, Daniele Giardino, Marco Re, and Sergio Spanò. 2021. Multi-agent reinforcement learning: A review of challenges and applications. *Appl. Sci.* 11, 11 (2021), 4948.

[18] Patrick P. K. Chan, Yaxuan Wang, and Daniel S. Yeung. 2020. Adversarial attack against deep reinforcement learning with static reward impact map. *Comput. Commun. Secur.* (2020).

[19] Hongyan Chang and Reza Shokri. 2021. On the privacy risks of algorithmic fairness. In *IEEE European Symposium on Security and Privacy (EuroS&P'21)*. IEEE, 292–303.

[20] Jianyu Chen, Shengbo Eben Li, and Masayoshi Tomizuka. 2020. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Trans. Intell. Transport. Syst.* (2020).

[21] Kangjie Chen, Shangwei Guo, Tianwei Zhang, Shuxin Li, and Yang Liu. 2021. Temporal watermarks for deep reinforcement learning models. *Auton. Agents. Multi-agent Syst.* (2021).

[22] Kangjie Chen, Shangwei Guo, Tianwei Zhang, Xiaofei Xie, and Yang Liu. 2021. Stealing deep reinforcement learning models for fun and profit. In *ACM Asia Conference on Computer and Communications Security*. 307–319.

[23] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 2017. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *10th ACM Workshop on Artificial Intelligence and Security*.

[24] Tianlong Chen, Huan Zhang, Zhenyu Zhang, Shiyu Chang, Sijia Liu, Pin-Yu Chen, and Zhangyang Wang. 2022. Linearity grafting: Relaxed neuron pruning helps certifiable robustness. In *International Conference on Machine Learning*. PMLR, 3760–3772.

[25] Xiaocong Chen, Lina Yao, Julian McAuley, Guanglin Zhou, and Xianzhi Wang. 2023. Deep reinforcement learning in recommender systems: A survey and new perspectives. *Knowl.-based Syst.* 264 (2023), 110335.
[26] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. 2020. SentiNet: Detecting localized universal attacks against deep learning systems. In *IEEE Security and Privacy Workshops (SPW'20)*. IEEE, 48–54.
[27] Sayak Ray Chowdhury and Xingyu Zhou. 2021. Differentially private regret minimization in episodic Markov decision processes. *arXiv preprint arXiv:2112.10599* (2021).
[28] Sayak Ray Chowdhury, Xingyu Zhou, and Ness Shroff. 2021. Adaptive control of differentially private linear quadratic systems. In *IEEE International Symposium on Information Theory (ISIT'21)*. IEEE, 485–490.
[29] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*. PMLR, 1310–1320.
[30] Christoph Dann, Tor Lattimore, and Emma Brunskill. 2017. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. *Adv. Neural Inf. Process. Syst.* 30 (2017).
[31] Thomas Degris, Martha White, and Richard S. Sutton. 2012. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839* (2012).
[32] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*. Springer, 265–284.
[33] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd USENIX Security Symposium (USENIX Security'14)*. 17–32.
[34] Ted Fujimoto, Timothy Doster, Adam Attarian, Jill Brandenberger, and Nathan Hodas. 2022. Reward-free attacks in multi-agent reinforcement learning.
[35] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *ACM SIGSAC Conference on Computer and Communications Security*. 619–633.
[36] Evrard Garcelon, Vianney Perchet, Ciara Pike-Burke, and Matteo Pirotta. 2021. Local differential privacy for regret minimization in reinforcement learning. *Adv. Neural Inf. Process. Syst.* 34 (2021).
[37] Hamid Gharagozlou, Javad Mohammadzadeh, Azam Bastanfard, and Saeed Shiry Ghidary. 2022. RLAS-BIABC: A reinforcement learning-based answer selection using the BERT model boosted by an improved ABC algorithm. *Computat. Intell. Neurosci.* 2022 (2022).
[38] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. 2019. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*.
[39] Parham Gohari, Bo Chen, Bo Wu, Matthew Hale, and Ufuk Topcu. 2021. Privacy-preserving kickstarting deep reinforcement learning with privacy-aware learners. *arXiv preprint arXiv:2102.09599* (2021).
[40] Parham Gohari, Matthew Hale, and Ufuk Topcu. 2020. Privacy-preserving policy synthesis in Markov decision processes. In *59th IEEE Conference on Decision and Control (CDC'20)*. IEEE, 6266–6271.
[41] Parham Gohari, Bo Wu, Matthew Hale, and Ufuk Topcu. 2020. The Dirichlet mechanism for differential privacy on the unit simplex. In *American Control Conference (ACC'20)*. IEEE, 1253–1258.
[42] Maziar Gomrokchi, Susan Amin, Hossein Aboutalebi, Alexander Wong, and Doina Precup. 2021. Where did you learn that from? Surprising effectiveness of membership inference attacks against temporally correlated data in deep reinforcement learning. *arXiv preprint arXiv:2109.03975* (2021).
[43] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv: Machine Learning* (2014).
[44] 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.
[45] Wenbo Guo, Xian Wu, Sui Huang, and Xinyu Xing. 2021. Adversarial policy learning in two-player competitive games. In *International Conference on Machine Learning*.
[46] Rob Hall, Alessandro Rinaldo, and Larry Wasserman. 2013. Differential privacy for functions and functional data. *J. Mach. Learn. Res.* 14, 1 (2013), 703–727.
[47] Ben Hambly, Renyuan Xu, and Huining Yang. 2023. Recent advances in reinforcement learning in finance. *Math. Finance* 33, 3 (2023), 437–503.
[48] Ali Hassan, Deepjyoti Deka, and Yury Dvorkin. 2021. Privacy-aware load ensemble control: A linearly-solvable MDP approach. *IEEE Trans. Smart Grid* 13, 1 (2021), 255–267.
[49] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. 2019. Towards privacy and security of deep learning systems: A survey. *arXiv preprint arXiv:1911.12562* (2019).
[50] Thomas Hickling, Nabil Aouf, and Phillippa Spencer. 2022. Robust adversarial attacks detection based on explainable deep reinforcement learning for UAV guidance and planning.

[51] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: Information leakage from collaborative deep learning. In *ACM SIGSAC Conference on Computer and Communications Security*. 603–618.
[52] Mengdi Huai, Jianhui Sun, Renqin Cai, Liuyi Yao, and Aidong Zhang. 2020. Malicious attacks against deep reinforcement learning interpretations. *Knowl. Discov. Data Min.* (2020).
[53] Sandy H. Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial attacks on neural network policies. *Learning* (2017).
[54] Yunhan Huang and Quanyan Zhu. 2019. Deceptive reinforcement learning under adversarial manipulations on cost signals. *Decis. Game Theor. Secur.* (2019).
[55] Léonard Hussenot, Matthieu Geist, and Olivier Pietquin. 2019. CopyCAT: Taking control of neural policies with constant attacks. *Adapt. Agents Multi-agents Syst.* (2019).
[56] Inaam Ilahi, Muhammad Usama, Junaid Qadir, Muhammad Umar Janjua, Ala Al-Fuqaha, Dinh Thai Hoang, and Dusit Niyato. 2020. Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *arXiv: Learning* (2020).
[57] Matthew Inkawhich, Yi Chen, and Hai Li. 2020. Snooping attacks on deep reinforcement learning. *Adapt. Agents Multi-agents Syst.* (2020).
[58] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A. Ortega, D. J. Strouse, Joel Z. Leibo, and Nando de Freitas. 2018. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. *arXiv: Learning* (2018).
[59] Alberto Jesu, Victor-Alexandru Darvariu, Alessandro Staffolani, Rebecca Montanari, and Mirco Musolesi. 2021. Reinforcement learning on encrypted data. *arXiv preprint arXiv:2109.08236* (2021).
[60] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against black-box membership inference attacks via adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security*. 259–274.
[61] Jiechuan Jiang and Zongqing Lu. 2018. Learning attentional communication for multi-agent cooperation. *Neural Inf. Process. Syst.* (2018).
[62] Chen Jin-Yin, Yan Zhang, Wang Xue-Ke, Hong-Bin Cai, Wang Jue, J. I. Shou-Ling, Zhang Yan, Cai Hong-Bin, and Ji Shou. 2022. A survey of attack, defense and related security analysis for deep reinforcement learning.
[63] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. 2019. PRADA: Protecting against DNN model stealing attacks. In *IEEE European Symposium on Security and Privacy (EuroS&P'19)*. IEEE, 512–527.
[64] Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. 2018. Model extraction warning in MLaaS paradigm. In *34th Annual Computer Security Applications Conference*. 371–380.
[65] Panagiota Kiourti, Kacper Wardega, Susmit Jha, and Wenchao Li. 2020. TrojDRL: Evaluation of backdoor attacks on deep reinforcement learning. In *57th ACM/IEEE Design Automation Conference (DAC'20)*. IEEE, 1–6.
[66] B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Yogamani, and Patrick Pérez. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transport. Syst.* 23, 6 (2021), 4909–4926.
[67] Jernej Kos and Dawn Song. 2017. Delving into adversarial attacks on deep policies. *Learning* (2017).
[68] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *Learning* (2016).
[69] Li-Cheng Lan, Huan Zhang, and Cho-Jui Hsieh. 2023. Can agents run relay race with strangers? Generalization of RL to out-of-distribution trajectories. *arXiv preprint arXiv:2304.13424* (2023).
[70] Jonathan Lebensold, William Hamilton, Borja Balle, and Doina Precup. 2019. Actor critic with differentially private critic. *arXiv preprint arXiv:1910.05876* (2019).
[71] 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 *National Conference on Artificial Intelligence*.
[72] Xian Yeow Lee, Aaron J. Havens, Girish Chowdhary, and Soumik Sarkar. 2019. Learning to cope with adversarial attacks. *arXiv: Learning* (2019).
[73] Chonghua Liao, Jiafan He, and Quanquan Gu. 2021. Locally differentially private reinforcement learning for linear mixture Markov decision processes. *arXiv preprint arXiv:2110.10133* (2021).
[74] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).
[75] 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 Symposium on Security and Privacy*.
[76] Yuanguo Lin, Yong Liu, Fan Lin, Lixin Zou, Pengcheng Wu, Wenhua Zeng, Huanhuan Chen, and Chunyan Miao. 2023. A survey on reinforcement learning for recommender systems. *IEEE Trans. Neural Netw. Learn. Syst.* (2023).
[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 Conference on Learning Representations*.

[78] Yen-Chen Lin, Ming-Yu Liu, Min Sun, and Jia-Bin Huang. 2017. Detecting adversarial attacks on neural network policies with visual foresight. *arXiv preprint arXiv:1710.00814* (2017).
[79] Michael L. Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*.
[80] Siqi Liu, Guy Lever, Josh Merel, Saran Tunyasuvunakool, Nicolas Heess, and Thore Graepel. 2018. Emergent coordination through competition. In *International Conference on Learning Representations*.
[81] Ximeng Liu, Robert H. Deng, Kim-Kwang Raymond Choo, and Yang Yang. 2019. Privacy-preserving reinforcement learning design for patient-centric dynamic treatment regimes. *IEEE Trans. Emerg. Topics Comput.* 9, 1 (2019), 456–470.
[82] Zhengshang Liu, Yue Yang, Tim Miller, and Peta Masters. 2021. Deceptive reinforcement learning for privacy-preserving planning. *arXiv preprint arXiv:2102.03022* (2021).
[83] Björn Lütjens, Michael Everett, and Jonathan P. How. 2020. Certified adversarial robustness for deep reinforcement learning. In *Conference on Robot Learning*. PMLR, 1328–1337.
[84] Paul Luyo, Evrard Garcelon, Alessandro Lazaric, and Matteo Pirotta. 2021. Differentially private exploration in reinforcement learning with linear representation. *arXiv preprint arXiv:2112.01585* (2021).
[85] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. 2017. Adversarially robust policy learning: Active construction of physically-plausible perturbations. *Intell. Robot. Syst.* (2017).
[86] Peta Masters and Sebastian Sardina. 2019. Goal recognition for rational and irrational agents. In *18th International Conference on Autonomous Agents and MultiAgent Systems*. 440–448.
[87] Seyyed Amir Hadi Minoofam, Azam Bastanfard, and Mohammad Reza Keyvanpour. 2022. RALF: An adaptive reinforcement learning framework for teaching dyslexic students. *Multim. Tools Applic.* 81, 5 (2022), 6389–6412.
[88] Hirofumi Miyajima, Noritaka Shigei, Hiromi Miyajima, and Norio Shiratori. 2018. Analog Q-learning methods for secure multiparty computation. *IAENG Int. J. Comput. Sci.* 45, 4 (2018), 623–629.
[89] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* (2015).
[90] Kanghua Mo, Weixuan Tang, Jin Li, and Xu Yuan. 2022. Attacking deep reinforcement learning with decoupled adversarial policy. *IEEE Trans. Depend. Sec. Comput.* (2022).
[91] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. 2017. Universal adversarial perturbations. *Comput. Vis. Pattern Recog.* (2017).
[92] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. DeepFool: A simple and accurate method to fool deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2574–2582.
[93] Zahra Movahedi and Azam Bastanfard. 2021. Toward competitive multi-agents in polo game based on reinforcement learning. *Multim. Tools Applic.* 80 (2021), 26773–26793.
[94] Dung Daniel T. Ngo, Giuseppe Vietri, and Steven Wu. 2022. Improved regret for differentially private exploration in linear MDP. In *International Conference on Machine Learning*. PMLR, 16529–16552.
[95] Thanh Thi Nguyen and Vijay Janapa Reddi. 2021. Deep reinforcement learning for cyber security. *IEEE Trans. Neural Netw. Learn. Syst.* (2021).
[96] Rui Nian, Jinfeng Liu, and Biao Huang. 2020. A review on reinforcement learning: Introduction and applications in industrial process control. *Comput. Chem. Eng.* 139 (2020), 106886.
[97] Olalekan Ogunmolu, Nicholas Gans, and Tyler H. Summers. 2017. Minimax iterative dynamic game: Application to nonlinear robot control tasks. *Intell. Robot. Syst.* (2017).
[98] Tuomas P. Oikarinen, Tsui-Wei Weng, and Luca Daniel. 2020. Robust deep reinforcement learning through adversarial loss. *Neural Inf. Process. Syst.* (2020).
[99] Hajime Ono and Tsubasa Takahashi. 2020. Locally private distributed reinforcement learning. *arXiv preprint arXiv:2001.11718* (2020).
[100] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* 35 (2022), 27730–27744.
[101] Xinlei Pan, Weiyao Wang, Xiaoshuai Zhang, Bo Li, Jinfeng Yi, and Dawn Song. 2019. How you act tells a lot: Privacy-leaking attack on deep reinforcement learning. In *18th International Conference on Autonomous Agents and MultiAgent Systems*. 368–376.
[102] Jaehyoung Park, Dong Seong Kim, and Hyuk Lim. 2020. Privacy-preserving reinforcement learning using homomorphic encryption in cloud computing infrastructures. *IEEE Access* 8 (2020), 203564–203579.
[103] Arpita Patra and Ajith Suresh. 2020. BLAZE: Blazing fast privacy-preserving machine learning. *arXiv preprint arXiv:2005.09042* (2020).

[104] Nhan H. Pham, Lam M. Nguyen, Jie Chen, Hoang Thanh Lam, Subhro Das, and Tsui-Wei Weng. 2022. Evaluating robustness of cooperative MARL: A model-based approach. *arXiv preprint arXiv:2202.03558* (2022).
[105] Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cédric Gouy-Pailler, and Jamal Atif. 2019. Theoretical evidence for adversarial robustness through randomization. *Adv. Neural Inf. Process. Syst.* 32 (2019).
[106] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning.
[107] Geong Sen Poh and Kok-Lim Alvin Yau. 2016. Preserving privacy of agents in reinforcement learning for distributed cognitive radio networks. In *23rd International Conference on Neural Information Processing (ICONIP'16)*. Springer, 555–562.
[108] Kritika Prakash, Fiza Husain, Praveen Paruchuri, and Sujit P. Gujar. 2021. How private is your RL policy? An inverse RL based analysis framework. *arXiv preprint arXiv:2112.05495* (2021).
[109] Dan Qiao and Yu-Xiang Wang. 2022. Offline reinforcement learning with differential privacy. *arXiv preprint arXiv:2206.00810* (2022).
[110] Erwin Quiring and Konrad Rieck. 2020. Backdooring and poisoning neural networks with image-scaling attacks. In *IEEE Symposium on Security and Privacy*.
[111] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. 2018. Modeling others using oneself in multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4257–4266.
[112] Amin Rakhsha, Goran Radanovic, Rati Devidze, Xiaojin Zhu, and Adish Singla. 2020. Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*.
[113] Amin Rakhsha, Xuezhou Zhang, Xiaojin Zhu, and Adish Singla. 2021. Reward poisoning in reinforcement learning: Attacks against unknown learners in unknown environments. *arXiv: Learning* (2021).
[114] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. 2020. Adversarial attacks and defenses in deep learning. *Engineering* (2020).
[115] Alessio Russo and Alexandre Proutiere. 2019. Optimal attacks on reinforcement learning policies. *arXiv: Learning* (2019).
[116] Jun Sakuma, Shigenobu Kobayashi, and Rebecca N. Wright. 2008. Privacy-preserving reinforcement learning. In *25th International Conference on Machine Learning*. 864–871.
[117] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).
[118] Vladimir Samsonov, Karim Ben Hicham, and Tobias Meisen. 2022. Reinforcement learning in manufacturing control: Baselines, challenges and ways forward. *Eng. Applic. Artif. Intell.* 112 (2022), 104868.
[119] Soumik Sarkar, Zhanhong Jiang, and Aaron J. Havens. 2018. Online robust policy learning in the presence of unknown adversaries. *Neural Inf. Process. Syst.* (2018).
[120] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.
[121] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International Conference on Machine Learning*. PMLR, 1889–1897.
[122] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
[123] Kanghyeon Seo and Jihoon Yang. 2020. Differentially private actor and its eligibility trace. *Electronics* 9, 9 (2020), 1486.
[124] Qianli Shen, Yan Li, Haoming Jiang, Zhaoran Wang, and Tuo Zhao. 2020. Deep reinforcement learning with robust and smooth policy. In *International Conference on Machine Learning*. PMLR, 8707–8718.
[125] Zhouxing Shi, Yihan Wang, Huan Zhang, J. Zico Kolter, and Cho-Jui Hsieh. 2022. Efficiently computing local Lipschitz constants of neural networks via bound propagation. *Adv. Neural Inf. Process. Syst.* 35 (2022), 2350–2364.
[126] Hocheol Shin, Yunmok Son, Youngseok Park, Yujin Kwon, and Yongdae Kim. 2016. Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems. In *10th USENIX Conference on Offensive Technologies (WOOT'16)*.
[127] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP'17)*. IEEE, 3–18.
[128] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* (2017).

[129] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. Reward is enough. *Artif. Intell.* 299 (2021), 103535.
[130] Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. 2019. Distributionally robust reinforcement learning. *arXiv: Machine Learning* (2019).
[131] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jung-Woo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX Security Symposium*.
[132] 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 *National Conference on Artificial Intelligence*.
[133] Yanchao Sun, Ruijie Zheng, Parisa Hassanzadeh, Yongyuan Liang, Soheil Feizi, Sumitra Ganesh, and Furong Huang. 2022. Certifiably robust policy learning against adversarial communication in multi-agent systems. *arXiv preprint arXiv:2206.10158* (2022).
[134] Richard S. Sutton. 1991. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bull.* 2, 4 (1991), 160–163.
[135] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.
[136] Weixuan Tang, Bin Li, Mauro Barni, Jin Li, and Jiwu Huang. 2020. An automatic cost learning framework for image steganography using deep reinforcement learning. *IEEE Trans. Inf. Forens. Secur.* 16 (2020), 952–967.
[137] Weixuan Tang, Bin Li, Mauro Barni, Jin Li, and Jiwu Huang. 2021. Improving cost learning for JPEG steganography by exploiting JPEG domain knowledge. *IEEE Trans. Circ. Syst. Vid. Technol.* (2021).
[138] Chen Tessler, Yonathan Efroni, and Shie Mannor. 2019. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*.
[139] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *25th USENIX Security Symposium (USENIX Security'16)*. 601–618.
[140] Edgar Tretschk, Seong Joon Oh, and Mario Fritz. 2018. Sequential attacks on agents for long-term adversarial goals. *arXiv: Learning* (2018).
[141] James Tu, Tsun-Hsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. 2021. Adversarial attacks on multi-agent communication. In *International Conference on Computer Vision*.
[142] Giuseppe Vietri, Borja Balle, Akshay Krishnamurthy, and Steven Wu. 2020. Private reinforcement learning with PAC and regret guarantees. In *International Conference on Machine Learning*. PMLR, 9754–9764.
[143] Nelson Vithayathil Varghese and Qusay H. Mahmoud. 2020. A survey of multi-task deep reinforcement learning. *Electronics* 9, 9 (2020), 1363.
[144] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *IEEE Symposium on Security and Privacy (SP'18)*. IEEE, 36–52.
[145] Baoxiang Wang and Nidhi Hegde. 2019. Privacy-preserving Q-learning with functional noise in continuous spaces. *Adv. Neural Inf. Process. Syst.* 32 (2019).
[146] Haoyu Wang, Guozheng Ma, Cong Yu, Ning Gui, Linrui Zhang, Zhiqi Huang, Suwei Ma, Yongzhe Chang, Sen Zhang, Li Shen, et al. 2023. Are large language models really robust to word-level perturbations? *arXiv preprint arXiv:2309.11166* (2023).
[147] Jingkang Wang, Yang Liu, and Bo Li. 2020. Reinforcement learning with perturbed rewards. In *National Conference on Artificial Intelligence*.
[148] Lun Wang, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. 2021. BACKDOORL: Backdoor attack against competitive reinforcement learning. In *International Joint Conference on Artificial Intelligence*.
[149] Ling Wang, Cheng Zhang, and Jie Liu. 2020. Deep learning defense method against adversarial attacks. *Syst., Man Cybern.* (2020).
[150] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. 2021. Beta-CROWN: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *Adv. Neural Inf. Process. Syst.* 34 (2021).
[151] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE Conference on Computer Communications (INFOCOM'19)*. IEEE, 2512–2520.
[152] Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. 2018. Towards fast computation of certified robustness for ReLU networks. In *International Conference on Machine Learning*. PMLR, 5276–5285.
[153] Tsui-Wei Weng, Krishnamurthy Dvijotham, Jonathan Uesato, Kai Xiao, Sven Gowal, Robert Stanforth, and Pushmeet Kohli. 2020. Toward evaluating robustness of deep reinforcement learning with continuous control. *Learning* (2020).
[154] Eric Wong and Zico Kolter. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*. PMLR, 5286–5295.

[155] Fan Wu, Linyi Li, Huan Zhang, Bhavya Kailkhura, Krishnaram Kenthapadi, Ding Zhao, and Bo Li. 2021. COPA: Certifying robust policies for offline reinforcement learning against poisoning attacks. In *International Conference on Learning Representations*.

[156] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. 2020. The value of collaboration in convex machine learning with differential privacy. In *IEEE Symposium on Security and Privacy (SP'20)*. IEEE, 304–317.

[157] Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. 2021. Adversarial policy training against deep reinforcement learning. In *30th USENIX Security Symposium (USENIX Security'21)*. 1883–1900.

[158] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864* (2023).

[159] Yingxiao Xiang, Wenjia Niu, Jiqiang Liu, Tong Chen, and Zhen Han. 2018. A PCA-based model to predict adversarial examples on Q-learning of path finding. In *IEEE International Conference on Data Science in Cyberspace*.

[160] Chaowei Xiao, Xinlei Pan, Warren He, Bo Li, Jian Peng, Mingjie Sun, Jinfeng Yi, Mingyan Liu, and Dawn Song. 2018. Characterizing attacks on deep reinforcement learning. *arXiv: Learning* (2018).

[161] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. 2019. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *USENIX Security Symposium*.

[162] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. 2018. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations*.

[163] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. 2019. Feature denoising for improving adversarial robustness. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 501–509.

[164] Tengyang Xie, Philip S. Thomas, and Gerome Miklau. 2019. Privacy preserving off-policy evaluation. *arXiv preprint arXiv:1902.00174* (2019).

[165] Zikang Xiong, Joe Eappen, He Zhu, and Suresh Jagannathan. 2022. Defending observation attacks in deep reinforcement learning via detection and denoising. *arXiv preprint arXiv:2206.07188* (2022).

[166] Hang Xu. 2022. Transferable environment poisoning: Training-time attack on reinforcement learner with limited prior knowledge. In *21st International Conference on Autonomous Agents and Multiagent Systems*. 1884–1886.

[167] Hang Xu, Rundong Wang, Lev Raizman, and Zinovi Rabinovich. 2021. Transferable environment poisoning: Training-time attack on reinforcement learning. In *20th International Conference on Autonomous Agents and Multiagent Systems*. 1398–1406.

[168] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. 2020. Automatic perturbation analysis for scalable certified robustness and beyond. *Adv. Neural Inf. Process. Syst.* 33 (2020).

[169] Mengdi Xu, Zuxin Liu, Peide Huang, Wenhao Ding, Zhepeng Cen, Bo Li, and Ding Zhao. 2022. Trustworthy reinforcement learning against intrinsic vulnerabilities: Robustness, safety, and generalizability. *arXiv preprint arXiv:2209.08025* (2022).

[170] Wanqi Xue, Wei Qiu, Bo An, Zinovi Rabinovich, Svetlana Obraztsova, and Chai Kiat Yeo. 2021. Mis-spoke or mis-lead: Achieving robustness in multi-agent communicative reinforcement learning. *arXiv: Learning* (2021).

[171] Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Yi Ouyang, I-Te Danny Hung, Chin-Hui Lee, and Xiaoli Ma. 2020. Enhanced adversarial strategically-timed attacks against deep reinforcement learning. In *International Conference on Acoustics, Speech, and Signal Processing*.

[172] Guoyu Yang, Yilei Wang, Zhaojie Wang, Youliang Tian, Xiaomei Yu, and Shouzhe Li. 2020. IPBSM: An optimal bribery selfish mining in the presence of intelligent and pure attackers. *Int. J. Intell. Syst.* 35, 11 (2020), 1735–1748.

[173] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *1st ACM International Conference on AI in Finance*. 1–8.

[174] Honggang Yu, Kaichen Yang, Teng Zhang, Yun-Yun Tsai, Tsung-Yi Ho, and Yier Jin. 2020. CloudLeak: Large-scale deep learning models stealing through adversarial examples. In *Network and Distributed System Security Symposium (NDSS'20)*.

[175] Jiahao Yu, Xingwei Lin, and Xinyu Xing. 2023. GPTFUZZER: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253* (2023).

[176] Liang Yu, Shuqi Qin, Meng Zhang, Chao Shen, Tao Jiang, and Xiaohong Guan. 2021. A review of deep reinforcement learning for smart building energy management. *IEEE Internet Things J.* 8, 15 (2021), 12046–12063.

[177] Yinbo Yu, Jiajia Liu, Shouqing Li, Kepu Huang, and Xudong Feng. 2022. A temporal-pattern backdoor attack to deep reinforcement learning. In *IEEE Global Communications Conference (GLOBECOM'22)*. IEEE, 2710–2715.

[178] Yinlong Yuan, Zhu Liang Yu, Zhenghui Gu, Xiaoyan Deng, and Yuanqing Li. 2019. A novel multi-step reinforcement learning method for solving reward hacking. *Appl. Intell.* (2019).

[179] Albert Zhan, Stas Tiomkin, and Pieter Abbeel. 2020. Preventing imitation learning with adversarial policy ensembles. *arXiv preprint arXiv:2002.01059* (2020).

[180] Huan Zhang, Hongge Chen, Duane S. Boning, and Cho-Jui Hsieh. 2021. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*.
[181] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane S. Boning, and Cho-Jui Hsieh. 2020. Robust deep reinforcement learning against adversarial perturbations on state observations. *Neural Inf. Process. Syst.* (2020).
[182] Haoqi Zhang and David C. Parkes. 2008. Value-based policy teaching with active indirect elicitation. In *National Conference on Artificial Intelligence*.
[183] Haoqi Zhang, David C. Parkes, and Yiling Chen. 2009. Policy teaching through reward function learning. *Electron. Commerce* (2009).
[184] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. 2018. Efficient neural network robustness certification with general activation functions. *Adv. Neural Inf. Process. Syst.* 31 (2018).
[185] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*. Springer, 321–384.
[186] Sai Qian Zhang, Qi Zhang, and Jieyu Lin. 2020. Succinct and robust multi-agent communication with temporal message control. *Neural Inf. Process. Syst.* (2020).
[187] Xueru Zhang and Mingyan Liu. 2021. Fairness in learning-based sequential decision algorithms: A survey. In *Handbook of Reinforcement Learning and Control*. Springer, 525–555.
[188] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. 2020. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *IEEE Symposium Series on Computational Intelligence (SSCI'20)*. IEEE, 737–744.
[189] Huaicheng Zhou, Kanghua Mo, Teng Huang, and Yongjin Li. 2023. Empirical study of privacy inference attack against deep reinforcement learning models. *Connect. Sci.* 35, 1 (2023), 2211240.
[190] Xingyu Zhou. 2022. Differentially private reinforcement learning with linear function approximation. *Proceedings of the ACM Measur. Anal. Comput. Syst.* 6, 1 (2022), 1–27.
[191] Ziyuan Zhou and Guanjun Liu. 2022. RoMFAC: A robust mean-field actor-critic reinforcement learning against adversarial perturbations on states. *arXiv preprint arXiv:2205.07229* (2022).
[192] Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528* (2023).
[193] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Adv. Neural Inf. Process. Syst.* 32 (2019).

Received 13 September 2022; revised 19 October 2023; accepted 1 December 2023