Computers & Security 129 (2023) 103259

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

# ATS-O2A: A state-based adversarial attack strategy on deep reinforcement learning

Xiangjuan Li, Yang Li*, Zhaowen Feng, Zhaoxuan Wang, Quan Pan

*Northwestern Polytechnical University, Xi'an, 710129, Shaanxi, China*

ARTICLE INFO

*Article history:*
Received 12 December 2022
Revised 1 April 2023
Accepted 9 April 2023
Available online 11 April 2023

*Keywords:*
Deep reinforcement learning
Adversarial attack
Targeted attack
Deep learning security
Machine learning

ABSTRACT

In recent years, deep reinforcement learning has been widely applied in many decision-making tasks requiring high safety and security due to its excellent performance. However, if an adversary attacks when the agent making critical decisions, it is bound to bring disastrous consequences because humans cannot detect it. Therefore, it is necessary to study adversarial attacks against deep reinforcement learning to help researchers design highly robust and secure algorithms and systems. In this paper, we proposed an attack method based on Attack Time Selection (ATS) function and Optimal Attack Action (O2A) strategy, named ATS-O2A. We select the critical attack moment through the ATS function, and then combine the state-based strategy with the O2A strategy to select the optimal attack action which has profound influence as targeted action, finally we launch an attack by making targeted adversarial examples. In order to measure the stealthiness and effectiveness of the attack, we designed a new measurement index. Experiments show that our method can effectively reduce unnecessary attacks and improve the efficiency of attacks.



## 1. Introduction

Since Deep Reinforcement Learning (DRL) has surpassed the human level on the Atari game platform (Mnih et al., 2015), the research on the DRL algorithm has developed rapidly. It has been widely applied in digital games (Lample and Chaplot, 2017), robot control (Tai et al., 2017), and other fields in the past few years. Reinforcement Learning (RL) is defined as a learning process that attempts to find the best action based on the information that an individual observes when interacting with the surrounding environment. As a combination of deep learning and reinforcement learning, DRL is an end-to-end perceptual control system. The agent learns the optimal behavior by interacting with the environment, which can be expressed as Markov Decision Process (MDP). In general, it can be interpreted by a tuple $< S, A, P, R, \gamma >$, where $S$ is the state space, $A$ is the action space, $P{:}S \times A \times S \rightarrow [0, 1]$ is the transition probability, $R$ denotes the reward, and $\gamma \in [0, 1)$ is the discount factor. At each time step, the current state of the agent can be represented as $s_t$, and then it will select action $a_t$ to enter the next state $s_{t+1}$ according to the policy network $\pi$, where the policy $\pi(s_t, a_t) \in [0, 1]$ represents the possibility of choosing an action and can also directly output the optimal action in the state. $r(s_t, a_t)$ represents the immediate reward after executing the action $a_t$. The goal of the agent is to learn an optimal strategy to maximize the final reward $R$, which can be interpreted as:

$$R = \sum_{t}^{T-1} E_{\pi(s_t, a_t)}[\gamma^t r(s_t, a_t)] \tag{1}$$

Nowadays DRL is widely used in decision-making tasks with high security and stability. However, researchers have proved the vulnerability of Deep Neural Networks (DNN) (Su et al., 2019). Although DRL shows good performance, the combination of deep learning inevitably leads it to certain vulnerabilities. When brings convenience to other fields, it also causes security problems and there are many works around how to defend against attack, such as quantum encryption (Ni et al., 2022) in the communication field. Therefore, it's necessary to study the vulnerability to design more robust and secure algorithms and systems. In 2014, Goodfellow et al. (2015) proposed Fast Gradient Sign Method (FGSM) to generate perturbation on neural networks, which provided ideas for subsequent adversarial attacks against DRL. Huang et al. (2017) who was the first one to add perturbations generated by FGSM to the observation for the attack, but they did not consider the high correlation between states and actions in continuous time in DRL. What's more their method attacks at each time step and ignores the effectiveness and stealthiness of adversarial attacks. Therefore, Lin et al. (2017) proposed strategically-timed attack and enchanting attack. The former reduces the num-

* Corresponding author.
*E-mail address:* liyangnpu@nwpu.edu.cn (Y. Li).

https://doi.org/10.1016/j.cose.2023.103259
0167-4048/

ber of attacks to about 25% of the total time but without considering the global reward.

In order to verify the vulnerability of the DRL model by using a more effective and stealthy method, in this paper, we proposed an attack strategy based on the Attack Time Selection (ATS) function and Optimal Attack Action (O2A), named ATS-O2A. The attack algorithm mainly focuses on two problems: when to attack and how to attack. We introduced Attack Time Selection (ATS) function, State-based strategy, and Optimal Attack Action (O2A) strategy, the first two are used to detect whether the attack is critical and has an impact on the agent's state, while the last considers the impact of attack action on the total reward.

Using this method, the attacker can achieve an ideal attack effect with the least number of attacks, while ensuring stealthiness and effectiveness. We deployed the proposed attack algorithm on Atari game agent trained by Deep Q-Network (DQN) (Mnih et al., 2015) and Advantage Actor-critic (A2C) (Mnih et al., 2016) respectively, and proved the superiority of our algorithm compared with other algorithms through experiments. The key contributions of this paper are as follows:

- Proposed the attack time selection function which improves the stealthiness of the attack by measuring the importance of the current moment.
- Proposed optimal attack action strategy which considers immediate reward and total reward to improve the effectiveness of the attack.
- Introduced the state-based strategy by comparing the next state action of the normal agent and attacked agent, which determines whether the attack is effective.
- Proposed a new attack effect measurement index by comprehensively considering attack frequency, attack success rate and cumulative reward change.

The rest of the paper is organized as follows. Section 2 reviews the current adversarial attack on DRL. Section 3 describes the proposed method. Section 4 presents the details of experiment and analyses the results. Section 5 discusses the mechanism and performance of the algorithm. Section 6 concludes the paper and draws the future work.

## 2. Literature review

Adversarial attack refers to the deliberate manipulation of machine learning models by attackers through specially crafted input samples to deceive or mislead the models. With the rapid development of deep learning, attackers are constantly exploring new attack methods, including using poisoning attacks (Chen and Koushanfar, 2022), adversarial machine learning (Hernández-Castro et al., 2022; Zizzo et al., 2019), and other technologies (Song et al., 2019; Wenger et al., 2021), making the effects of adversarial attack increasingly difficult to detect and defend. Poisoning attacks mainly occur in the training stage which can reduce performance and reliability by using harmful data. Similarly, in DRL the attacker manipulates the input data during the training stage thus introducing a prediction bias to the model. The aim of adversarial machine learning is to improve robustness and security by studying potential attacks and threats. These methods are essentially the same as the attacks on DRL. Especially, in this paper, we will study the vulnerability of the DRL model which is mainly based on the adversarial machine learning method.

Adversarial attacks in the field of DRL can be divided into reward-based attack (Zhang et al., 2020), strategy-based attack (Behzadan and Hsu, 2019; Behzadan and Munir, 2017; Kos and Song, 2017; Mo et al., 2023), observation-based attack (Hussenot et al., 2020; Li et al., 2022; Lin et al., 2017; Sun et al., 2020), environment-based attack (Bai et al., 2018; Chen et al., 2018) and action-based attack (Lee et al., 2020) according to algorithm principle. Reward-based attack refers to modifying the reward signal of environment feedback, which can either directly modify the sign of reward value or replace the original reward function with an adversarial reward function. Strategy-based attack refers to the use of adversarial agents to generate states and behaviors beyond the victim agent's understanding ability, and then cause the victim agent to enter a chaotic state. Observation-based attack refers to the attacker adding perturbation to the observed image to make the victim agent execute the action expected by the attacker. This is usually achieved by adding perturbations to the image sensor of the agent. Environmental-based attack refers to modifying the training environment of the agent directly, mainly by modifying the dynamic model of the environment and adding obstacles to it. Action-based attack refers to directly modifying the action output, which can be implemented by modifying the action space in the training data. In this paper, we focus on observation-based attack.

In observation-based attack, the attacker adds appropriate perturbation to the observation image at a certain time to mislead the agent to make a wrong choice. While ensuring stealthiness, the cumulative reward is minimized, and the final state of the agent even be changed. At present, the research mainly focuses on reducing the frequency of attacks. In the study of Lin et al. (2017), strategically-timed attack reduces the frequency of attacks to some extent by calculating the action probability difference of each step. However, it only considers the attack effect at the current moment and ignores the impact of the current attack on the subsequent states and the final attack effect. And in the enchanting attack, although it considers the final target and global optimization of the attack, it is difficult to predict the states and actions in a long time range thus causing a low success rate. Sun et al. (2020) proposed critical point attack and antagonistic attack. But they both limit attack moments and ignore future critical moments. And the selection of attack action sequences in critical point attack needs to be exhaustive, which improves the time complexity.

Different from traditional adversarial attack methods of generating adversarial examples, Gu et al. (2017) proposed to launch the adversarial attack by implanting a backdoor and changing model parameters, resulting in the wrong output of the neural network. Based on the above background, Kiourti et al. (2020) proposed the backdoor attack and Rakhsha et al. (2020) proposed an environment poisoning attack against DRL. However, these methods need to poison a large number of states in the training process to achieve the gap in performance, and need to be further optimized. At the same time, all the above algorithms directly measure the attack effect by the variation of reward, without considering the internal relationship among attack frequency, attack success rate and the variation of cumulative reward.

Based on the above problems, we proposed an algorithm that uses ATS function and state-based strategy to control the number of attacks and uses O2A strategy to attack at the selected critical moment. In addition, in order to comprehensively evaluate the attack effect, we proposed a new measurement index $F$, and the experiment proved that the comprehensive performance of our proposed attack algorithm is better.

## 3. Methodology

In DRL, the agent interacts with the environment through a series of operations, each of which changes the corresponding state. Compared with other forms of adversarial attacks, adversarial attacks oriented to DRL need to consider the influence of actions on subsequent states and analyze whether to add perturbation to each state and the possible influence. At the same time, the goal is to reduce the total reward, and even mislead the agent into a dangerous state, which is different from the ordinary attack that

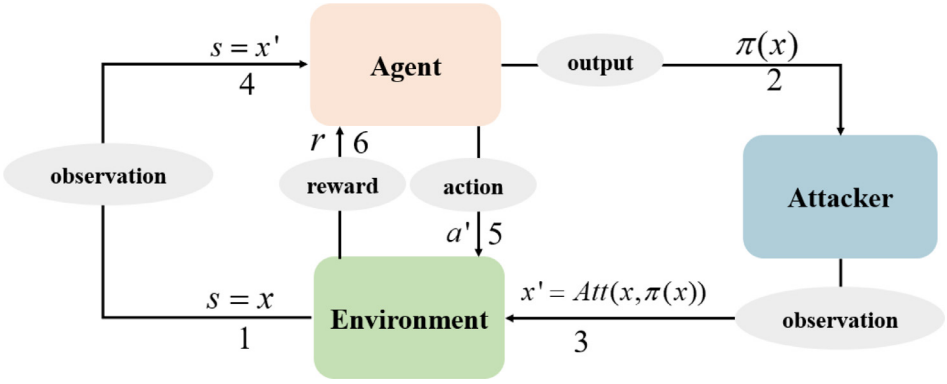


**Fig. 1.** Process of attack based on observation. Step 1: The victim agent receives an observation $x$ from the environment and sets it as the current state. Step 2: The victim agent outputs an action according to policy $\pi$ at the normal state. Step 3: The attacker gets the output and crafts an adversarial example $x'$ by $Att$ function to the environment. Step 4: The victim agent receives $x'$ and sets it as the current state. Step 5: The victim agent selects the wrong action $a'$ at attacked state. Step 6: The victim agent receives a reward after executing $a'$.

aims to reduce the classification accuracy. In contrast, tasks in DRL are more complex and specific, requiring a set of highly correlated action predictions. The adversary needs to change the final goal of the agent, rather than simply mislead the action of one step. What's more, an attack on DRL requires that, in addition to injecting a subtle perturbation into the input, it also causes damage to the entire process. Figure 1 shows the process in which an attacker adds perturbations to the agent's observations to make it select the wrong action.

For adversarial attacks oriented to DRL, there are two main problems: when to attack and how to attack (i.e. how to make adversarial examples). By analyzing the advantages and disadvantages of existing algorithms, we propose a method to select the critical attack moment using the ATS function, select the targeted action using $\pi_{adv}$ obtained by O2A strategy and then analyze whether to attack through the state-based strategy. The core of the algorithm is mainly composed of three parts: ATS function, state-based strategy and O2A strategy.

### 3.1. ATS function

The DRL algorithm based on policy gradient mainly parameterizes the policy $\pi$, calculates the policy gradient about the action, and then adjusts the action along the direction of the gradient to get the optimal policy gradually. Its output $\pi(a|s) = p[a|s,\theta]$ indicates that when the state is $s$, the action $a$ satisfies a certain probability distribution of the parameter $\theta$. Thus, we select the critical moment by measuring the action probability distribution of the agent output in a certain state, and the function can be expressed as:

$$C(t) = \alpha(\pi_{\max} - \pi_{\min}) + \beta(\pi_{\max} - \overline{\pi}) > \Delta \tag{2}$$

where $\pi_{\max}, \pi_{\min}$ *and* $\overline{\pi}$ respectively denote the maximum value, the minimum value and the average value of the output action probability in a given state, $\alpha$ and $\beta$ are two parameters which satisfy $\alpha + \beta = 1$, and $\Delta$ is the threshold value. For value function-based DRL algorithms such as DQN, DNN is mainly used to approximate the reward value function, and the Q value represents the value distribution of the action in this state. In this paper, the Q value needs to be processed by the softmax function first, and then the function can be expressed as:

$$\begin{aligned} C(t) = {} & \alpha(\max\Phi(Q(s_t,a_t)) - \min\Phi(Q(s_t,a_t))) \\ & + \beta(\max\Phi(Q(s_t,a_t)) - \overline{\Phi}(Q(s_t,a_t))) > \Delta \end{aligned} \tag{3}$$

where $\Phi(Q(s_t,a_t))$ means normalizing the Q value by softmax function (temperature coefficient $T$ is 1 in the experiment), $\overline{\Phi}$ is the average value of the function.

The function is composed of two parts, the first part represents the preference degree of the agent to a certain action in a given state, and the second part represents the difference among the probability of all actions. The larger value of $C(t)$, the greater preference of this moment for a certain action, and the more critical the moment is. On the contrary, it means that the difference in action probability at this moment is smaller, so it cannot be used as the critical attack moment. Through this function, we can select the most critical moment to ensure that as few attacks as possible and achieve efficient attack, so as to ensure the time stealthiness of the attack algorithm. Since $\alpha$ and $\beta$ are unknown, we designed a preliminary experiment to determine the values of two parameters.

### 3.2. State-based model

The model is used to judge whether an attack has a long-time impact on the subsequent states of the agent. We consider that if subsequent state and behavior after the attack are not changed, it means that the attack has little influence on the agent, and the cumulative reward may not have any influence on the final state although it decreases. In order to improve the effectiveness of each attack, we restrict the number of attacks based on state strategy, so as to ensure that a single attack can have a profound impact on the agent. The specific process is shown in Fig. 2: when an attacking moment is selected, the state is recorded as $s_t$. Normally, as the solid line shows, the agent will execute $a_t$ and update state to $s_{t+1}$ where its next normal action is $a_{t+1}$. Then the red dotted line represents the attack process. The attacker will generate a perturbation $\delta$ and craft an adversarial example that can change the input state to $s'_t$ and mislead the agent to state $s'_{t+1}$ by selecting targeted action $a'_t$. The next normal action of state $s'_{t+1}$ is $a'_{t+1}$. At last the attacker will compare $a_{t+1}$ and $a'_{t+1}$. If they are different, it means that both the state and behavior of the agent are affected and the attack is effective. Otherwise, the attack will be canceled.

### 3.3. O2A strategy

In traditional attack algorithms, targeted actions are usually selected by suboptimal actions given by the policy function or actions in the action sequence generated by exhaustive. The former has little effect on the cumulative reward, while the latter improves the complexity of the algorithm, so we introduce the O2A strategy. The model is used to select the action that minimizes the cumulative reward of the agent in a given state. The model is established based on a deep neural network and optimized by objec-

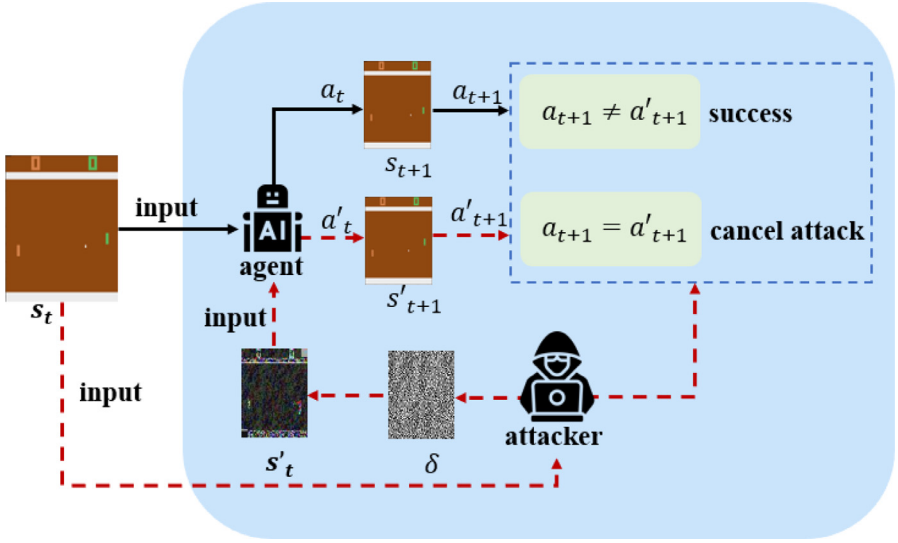


**Fig. 2.** Process of state-based strategy.

tive function:

$$\sum_{t}^{T-1} E_{(p_t, a'_t)\sim\pi_{adv}(s_t), a_t\sim\pi(s_t)}[\gamma^t r_{adv}(s_t, a'_t)] \tag{4}$$

where $r_{adv}(s_t, a'_t) = -r(s_t, a_t)$ and $\pi_{adv}$ means the O2A strategy. The model's input is the state, and the output is the optimal attack action in the state and the probability that the action is selected. The larger the $p_t$ is, the more important the attack action is. We train the model by using the original action and state space, and the parameters $(s_t, a'_t, p_t, r_{adv})$ are collected and updated in each training to obtain the attack action and state space.

### 3.4. Process of algorithm

Taking the DQN agent as an example, the attack process can be described as Algorithm 1: in line 1 we initialize the agent state.

**Algorithm 1:** ATS-O2A algorithm.

**Input**: Atari Environment, victim Agent $T$, $\pi_{adv}$,threshold value $\Delta$
**Output**: total reward $R$, attack rate, successful attack rate

1 Initialize state $s$;
2 **for** *each episode* **do**
3  **for** *each time step t and current episode is not done* **do**
4  **if** $c(t) > \Delta$ **then**
5  $a_t = AgentAct(s_t)$;
6  execute $a_t$, get state $s_{t+1}$;
7  $a_{t+1} = AgentAct(s_{t+1})$;
8  $a'_t = \pi_{adv}(s_t)$;
9  execute $a'_t$, get state $s'_{t+1}$;
10  $a'_{t+1} = AgentAct(s'_{t+1})$;
11  **if** $a_{t+1} = a'_{t+1}$ **then**
12  cancel attack;
13  set state $s_t = s_{t+1}$;
14  **end**
15  **end**
16  perform next step
17  **end**
18 **end**

Then, in line 4 we use Eq. (2) to obtain the attack moment. The state at this moment is denoted as $s_t$, and lines 5–6 indicate the state after performing an action $a_t$ under normal conditions is recorded as $s_{t+1}$, and in line 7 the action selection of the agent under state $s_{t+1}$ is denoted as $a_{t+1}$. In lines 8–9 the state after executing the attack is recorded as $s'_{t+1}$, and in line 10 the action selection of the agent in the state is recorded as $a'_{t+1}$. We compare $a_{t+1}$ and $a'_{t+1}$ in line 11, if they are the same, the attack is invalid, and we cancel the attack as shown in lines 12–13. Otherwise, we consider the attack successful and continue in line 16.

**Table 1**
Structure of parameters of neural networks .

| Layer | In-channels | Out-channels | Kernel-size | Stride | Activation |
|---|---|---|---|---|---|
| Conv1 | Any | 32 | 8 | 4 | ReLU |
| Conv2 | 32 | 64 | 4 | 2 | ReLU |
| Conv3 | 64 | 64 | 3 | 1 | ReLU |
| FC | — | — | — | — | Softmax |

## 4. Experiment and analysis

We evaluate the effectiveness of the proposed attack algorithm on Atari game datasets. At the same time, to further evaluate the effectiveness and stealthiness of the algorithm, a new measurement index is proposed in this section.

### 4.1. Experiment

#### 4.1.1. Agent training

The adversarial attack experiments based on DRL in this paper take the DQN algorithm and A2C algorithm as examples. The specific parameters of the convolutional neural network are shown in Table 1. Three convolutional layers and one fully connected layer are used, and the convolutional layer is activated by the ReLU function. 'Any' denotes the channel number which is 3 for RGB images and 1 for gray images. During training, based on Pytorch[1] and Tianshou[2] algorithm libraries, we used the DQNPolicy module and A2CPolicy module in Tianshou, the discount factor is set to 0.99,

[1] PyTorch is a Python-based scientific computing library and an open-source machine learning framework used for building neural networks.
[2] Tianshou is a PyTorch-based reinforcement learning framework designed to provide efficient implementation and easy-to-use API.

**Table 2**
Range of $C(t)$ value in the first 20%.

| Game | Range of threshold |
|---|---|
| Pong | (0.016,0.247) |
| Breakout | (0.13,0.85) |
| Qbert | (0.107,0.899) |
| MsPacman | (0.10,0.22) |
| SpaceInvaders | (0.046,0.261) |

the learning rate is 0.0001, and the targeted network is updated per 500 steps. The Normal Reward (NR) of the trained agent with DQN and A2C under different games is shown in Table 3.

#### 4.1.2. O2A strategy training

For attack strategy O2A, its principle is similar to the policy $\pi$ in DRL. The difference is that the original reward corresponding to each action is changed to a negative value, that is, the objective function is optimized in the direction of reward reduction. The original state-action space is used as the training set during model training. The goal of this strategy is to choose the action that reduces the overall reward the most in a given state. Take the game of Pong as an example, the rule of the game is that the first one to score 21 points wins. For the normal agent, the final reward should be close to 21, but for the agent using $\pi_{adv}$, the final result tends to -21, which is exactly opposite to the goal of the normal agent. In the training process, the original Atari game dataset in the gym library is used as the training set, and the episode is set to 100.

#### 4.1.3. Parameters of ATS function

For the ATS function, there are two problems to be solved: one is how to set $\alpha$ and $\beta$, and the other is how to select the function threshold to ensure the attack effect. In order to solve this problem, we restrict $\alpha + \beta = 1$, and define $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. In the Pong game, we recorded the function results under each $\alpha$ value and selected all data to draw the overall frequency distribution histogram. The results of the distribution frequency within 20% were taken as reference, and then the function results under each value were individually drawn frequency distribution histograms. By comparing the covariance between population and individual distribution, the final $\alpha$ value is 0.5. Then we got the threshold range of each game within the top 20% of $C(t)$ value. The specific results are shown in Table 2. According to this table, the $\Delta$ values of five games are 0.016, 0.13, 0.107, 0.10 and 0.046.

#### 4.1.4. Perturbation generation

In this paper, we selected three respective algorithms to generate perturbations:

**CW:** Carlini & Wagner (CW) attack algorithm (Carlini and Wagner, 2017) is an optimization-based attack, which takes into account the two aspects of high attack accuracy and low adversarial disturbance at the same time.

**FGSM:** Fast gradient sign method (FGSM) (Goodfellow et al., 2015) is a gradient-based attack, which mainly finds the derivative of the model with respect to the input to generate perturbations.

**PGD:** Project Gradient Descent (PGD) attack algorithm (Madry et al., 2018) is the strongest first-order attack algorithm at present. It performs several iterations, each iteration generates a new perturbation and trims it to the specified range.

Considering the time complexity and algorithm effect, we set the number of iterations of CW and PGD as 50 and set the epsilon in FGSM as 0.1.

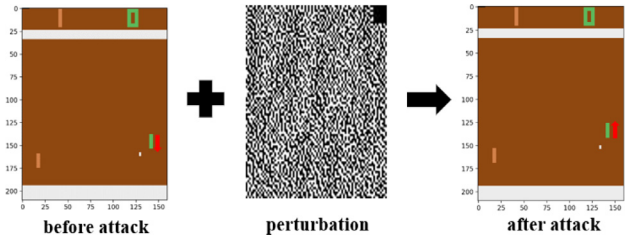


**Fig. 3.** Attack process (take Pong game as an example, the red arrow represents the direction of action): The right player is the agent. Normally, the agent should take downward action to ensure receiving the ball when the ball is close to it. After executing the attack (that is, using the FGSM algorithm to add perturbation to the observation to generate adversarial samples), the victim agent takes upward action, thus failing to receive the ball. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

### 4.2. Experimental results and analysis

In order to verify the performance of the proposed algorithm, we conduct experiments on five games: Pong, Breakout, Qbert, MsPacman, and SpaceInvaders. We implemented Strategically-Timed (ST) attack algorithm, Uniform (UN) attack algorithm (attacks in a uniform distribution) and our algorithm ATS-O2A, recorded the total number of steps, attack times, attack frequency, attack success rate, and cumulative reward. Figure 3 shows the attack process of ATS-O2A.

According to traditional measures that only focus on cumulative reward reduction, Table 3 records the minimum reward (average of 10 episodes) for each of the three attack algorithms in the experiment based on FGSM. However, the three algorithms have different attack frequencies and different attack success rates, it is not possible to directly measure the attack effect only by the change of cumulative reward. We need a measurement index that comprehensively considers the attack frequency, attack success rate and cumulative reward variation. In order to make a more intuitive comparison, we take the difference between the attack success rate and the attack frequency as the abscissa to represent the attack stealthiness, namely:

$$\Delta frequency = f_{suc} - f_{total} \tag{5}$$

where $f_{suc}$ represents attack success rate, and $f_{total}$ represents attack frequency.

We take the difference between the maximum cumulative reward of the normal agent and the minimum cumulative reward of the victim as the maximum reward variation, and then the ratio between the cumulative reward variation and the maximum reward variation after executing the attack algorithm is used as the ordinate to represent the attack effectiveness, namely:

$$\Delta R = \frac{\max(R_{normal}) - R_{attacked}}{\max(R_{normal}) - \min(R_{attacked})} \tag{6}$$

where $R_{normal}$ represents normal reward of agent and $R_{attacked}$ represents reward after being attacked. According to different rules of games, the $\min(R_{attacked})$ value of Pong is -21, and 0 in the remaining four games.

For an effective attack algorithm, the ideal attack effect is low attack frequency, high attack success rate and large accumulative reward variation. Therefore, we propose a measurement index $F$ composed of $\Delta frequency$ and $\Delta R$ to comprehensively evaluate the stealthiness and effectiveness of the attack. The larger the $F$ value, the better the attack effect. The mathematical formula is expressed as:

$$F = 0.5 \times (\Delta frequency + \Delta R) \tag{7}$$

We calculated the $F$ values of uniform attack, strategically-timed attack, and our attack algorithm under different attack frequencies

**Table 3**
Reward of DQN and A2C agents under different conditions.

| | Pong | | Breakout | | Qbert | | MsPacman | | SpaceInvaders | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DQN | A2C | DQN | A2C | DQN | A2C | DQN | A2C | DQN | A2C |
| NR | 21 | 21 | 349 | 400 | 5372 | 5445 | 1784 | 2236 | 719 | 906 |
| UN | -17 | -18 | 0 | 0 | 200 | 150 | 380 | 320 | 260 | 260 |
| ST | -21 | -21 | 1 | 2 | 0 | 75 | 350 | 270 | 145 | 210 |
| ATS-O2A | -21 | -21 | 0 | 0 | 100 | 50 | 10 | 380 | 310 | 60 |

**Table 4**
*F* value records after attack.

| | Pong | | Breakout | | Qbert | | MsPacman | | SpaceInvaders | |
|---|---|---|---|---|---|---|---|---|---|---|
| | DQN | A2C | DQN | A2C | DQN | A2C | DQN | A2C | DQN | A2C |
| UN+CW | 0.371 | 0.426 | 0.637 | 0.498 | 0.576 | 0.514 | 0.317 | 0.432 | 0.285 | 0.246 |
| UN+FGSM | 0.611 | 0.593 | 0.621 | 0.609 | 0.629 | 0.630 | 0.526 | 0.481 | 0.474 | 0.522 |
| UN+PGD | 0.653 | 0.677 | 0.712 | 0.705 | 0.693 | 0.683 | 0.550 | 0.560 | 0.555 | 0.576 |
| ST+CW | 0.822 | 0.364 | 0.573 | 0.320 | 0.785 | 0.212 | 0.644 | 0.030 | 0.308 | 0.117 |
| ST+FGSM | 0.587 | 0.359 | 0.779 | 0.372 | 0.481 | 0.377 | 0.550 | 0.027 | 0.360 | 0.224 |
| ST+PGD | 0.886 | 0.524 | 0.795 | 0.373 | 0.922 | 0.714 | 0.573 | 0.393 | 0.664 | 0.612 |
| ATS-O2A+CW | **0.951** | **0.765** | **0.977** | **0.653** | **0.950** | **0.704** | **0.679** | 0.384 | **0.687** | **0.411** |
| ATS-O2A+FGSM | **0.861** | 0.553 | **0.886** | **0.609** | **0.650** | **0.641** | **0.631** | 0.193 | **0.510** | 0.400 |
| ATS-O2A+PGD | **0.951** | **0.887** | **0.971** | **0.902** | **0.947** | **0.884** | **0.677** | 0.483 | **0.718** | **0.618** |

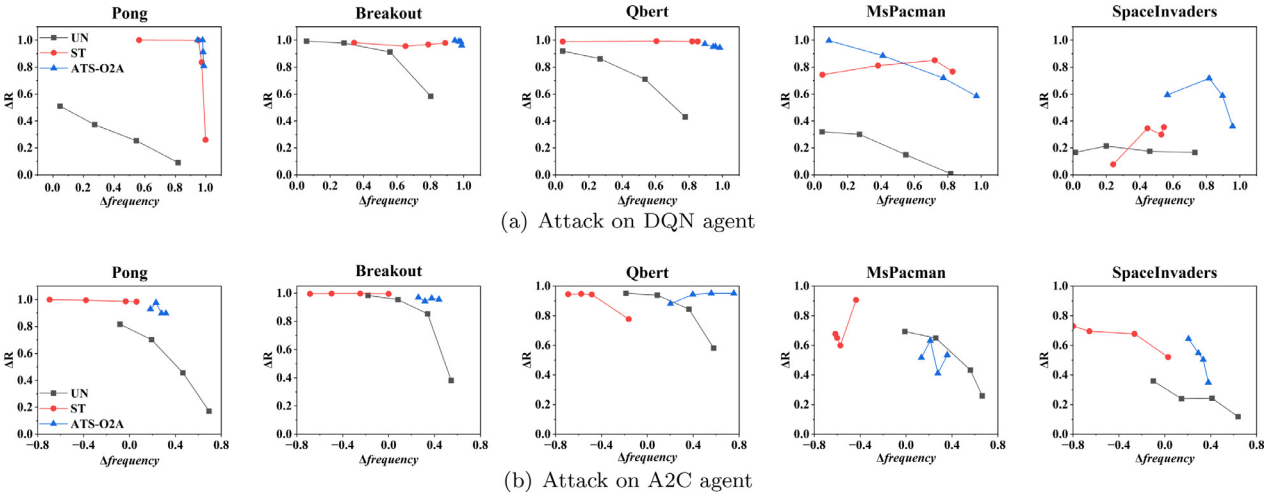


**Fig. 4.** Attack results using CW algorithm, (a) and (b) represent the attack results on the agents that trained with the five games under DQN and A2C, the abscissa represents $\Delta frequency$, and the ordinate represents $\Delta R$, UN represents a uniform attack, ST represents a strategically timed attack, Ours represents our algorithm.

by using CW, FGSM and PGD to generate adversarial samples respectively. All results are recorded in Table 4 after taking the mean value, the higher the better. ST is a strategically-timed attack, UN is a uniform attack, and Ours is our attack.

According to Table 4, we find that the performance of our algorithm is better than uniform attack and strategically-timed attack. Especially in Pong game and Breakout game, the *F* value of our algorithm combined with PGD algorithm on DQN is 0.951 and 0.971 respectively, and the *F* value combined with the CW algorithm is 0.951 and 0.977 respectively, which is much higher than other algorithms. Although the performance of our attack algorithm in the MsPacman game on A2C agent is slightly lower than that of uniform attack, the overall results show that the *F* value of our algorithm for the five games has been improved to varying degrees. In addition, Figure 4 shows the results of the uniform attack, strategically-timed attack, and our attack algorithm respectively combined with CW algorithm.

In Fig. 4, when the $\Delta frequency$ and $\Delta R$ are larger, it means that the attack frequency is lower, the attack success rate is higher, and the cumulative reward decreases more. We find that our algorithm curves are generally above the results of strategically-timed attacks and uniform attacks except on the A2C agent of MsPacman. In Fig. 4(a), uniform attack performs worse than the strategically-timed attack on the whole, while in Fig. 4(b), it is the opposite. At the same time, the $\Delta frequency$ and $\Delta R$ of our algorithm in Pong, Breakout and Qbert are close to 1 in Fig. 4(a). It means that the attack success rate of our algorithm is much higher than the attack frequency, and the attack times are greatly reduced while guaranteeing the attack effect, which further illustrates the effectiveness and stealthiness of the algorithm. In attacks against A2C agents, the $\Delta frequency$ of most games are in (0,0.6). Although there are cases in Qbert games that are greater than 0.6, the overall performance is still slightly lower than attacks against DQN agents. We analyze one of the reasons is that the robustness of A2C algorithm is higher than that of the DQN algorithm. At the same time, by comparing the above figures, we can see that our algorithm has higher performance under the same $\Delta R$. In the attack in Pong, Breakout, and Qbert, our algorithm $\Delta R$ is distributed around 1, which is greatly improved compared to the strategically-timed attack and uniform attack. Therefore, according to the above analysis, the performance of our algorithm is better than the other two algorithms.

## 5. Discussion

In this paper we focus on the vulnerability of the well-trained deep reinforcement learning model, which means that the model may exist some threats we don't know. The attacker explores and exploits the vulnerability before any information about the vulnerability has been released. So our algorithm ATS-O2A is essentially similar to zero-day attack (Sayed et al., 2023). Compared with other attack algorithms like UN and ST, ATS-O2A has shown good performance by selecting the critical attack moment and inducing the agent to perform actions that have the greatest impact on the long-term reward.

Moreover, there are still several limitations in our study regarding the use of adversarial examples in DRL. On the one hand, our algorithm ATS-O2A is a white-box attack, which requires the attacker to access the state space and action space. On the other hand, training the O2A strategy will be more difficult if the targeted agent has a complex state and action space.

## 6. Conclusions and future works

In this paper, we proposed an attack algorithm ATS-O2A aimed at DRL which ensures both effectiveness and stealthiness. According to experimental results, the attack success rate of the ATS-O2A is significantly improved and over 90% in most cases. Compared with the same effect of the strategically-timed attack, the time stealthiness of the attack algorithm is greatly improved. And we proposed a new measure index $F$ to show the effectiveness and stealthiness of attack. We found the $F$ value of the ATS-O2A is obviously larger than the strategically-timed attack algorithm and uniform attack algorithm in most cases. Especially in Pong and Breakout games, the $F$ value is almost near 1, which means that only in a few steps the agent will be confused and eventually fail. The results proved the effectiveness and stealthiness of ATS-O2A.

Based on the completed work in this paper, we have verified the vulnerability of the DRL. In order to improve the robustness and security of DRL algorithms, we plan to explore the following aspects in the future: (1) To defend the observation-based attack, we consider improving the upper bound of anti-interference and design a denoising model, which will enhance the robustness of DRL. (2) In this paper we mainly focus on theory while DRL has been applied in many tasks, such as autonomous navigation and robot control. Thus, we plan to deploy our algorithm in those tasks to explore more characteristics.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Xiangjuan Li:** Conceptualization, Investigation, Writing – original draft. **Yang Li:** Writing – review & editing, Supervision. **Zhaowen Feng:** Supervision. **Zhaoxuan Wang:** Supervision. **Quan Pan:** Funding acquisition.

## Data availability

The data this paper used is opensource.

## Acknowledgments

This research is supported by the National Natural Science Foundation of China (No.62103330, 62203358, 62233014), and the Fundamental Research Funds for the Central Universities (3102021ZDHQD09).

## References

Bai, X., Niu, W., Liu, J., Gao, X., Xiang, Y., Liu, J., 2018. Adversarial examples construction towards white-box Q table variation in DQN pathfinding training. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, pp. 781–787. doi:10.1109/dsc.2018.00126.

Behzadan, V., Hsu, W.H., 2019. Adversarial exploitation of policy imitation. In: Proceedings of the Workshop on Artificial Intelligence Safety 2019 Co-Located with the 28th International Joint Conference on Artificial Intelligence. CEUR-WS.org. https://ceur-ws.org/Vol-2419/paper_40.pdf

Behzadan, V., Munir, A., 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In: Proceedings of the 13th International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, pp. 262–275. doi:10.1007/978-3-319-62416-7_19.

Carlini, N., Wagner, D., 2017. Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 39–57. doi:10.1109/sp.2017.49.

Chen, H., Koushanfar, F., 2022. Tutorial: towards robust deep learning against poisoning attacks. ACM Trans. Embed. Comput. Syst. doi:10.1145/3574159.

Chen, T., Niu, W., Xiang, Y., Bai, X., Liu, J., Han, Z., Li, G., 2018. Gradient band-based adversarial training for generalized attack immunity of A3C path finding. CoRR. http://arxiv.org/abs/1807.06752

Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples. In: the 3rd International Conference on Learning Representations. http://arxiv.org/abs/1412.6572

Gu, T., Dolan-Gavitt, B., Garg, S., 2017. Badnets: identifying vulnerabilities in the machine learning model supply chain. CoRR. http://arxiv.org/abs/1708.06733

Hernández-Castro, C.J., Liu, Z., Serban, A., Tsingenopoulos, I., Joosen, W., 2022. Adversarial machine learning. In: Security and Artificial Intelligence, pp. 287–312. doi:10.1007/978-3-030-98795-4_12.

Huang, S.H., Papernot, N., Goodfellow, I.J., Duan, Y., Abbeel, P., 2017. Adversarial attacks on neural network policies. In: Proceedings of the 5th International Conference on Learning Representations. OpenReview.net. https://openreview.net/forum?id=ryvlRyBKl

Hussenot, L., Geist, M., Pietquin, O., 2020. CopyCAT: taking control of neural policies with constant attacks. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, pp. 548–556. doi:10.5555/3398761.3398828.

Kiourti, P., Wardega, K., Jha, S., Li, W., 2020. TrojDRL: evaluation of backdoor attacks on deep reinforcement learning. In: 2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE, pp. 1–6. doi:10.1109/dac18072.2020.9218663.

Kos, J., Song, D., 2017. Delving into adversarial attacks on deep policies. In: 5th International Conference on Learning Representations. OpenReview.net.

Lample, G., Chaplot, D.S., 2017. Playing FPS games with deep reinforcement learning. In: the 31st AAAI Conference on Artificial Intelligence. IEEE, pp. 600–603. doi:10.1609/aaai.v31i1.10827.

Lee, X.Y., Ghadai, S., Tan, K.L., Hegde, C., Sarkar, S., 2020. Spatiotemporally constrained action space attacks on deep reinforcement learning agents. In: The 34th AAAI Conference on Artificial Intelligence. AAAI Press, pp. 4577–4584. https://ojs.aaai.org/index.php/AAAI/article/view/5887

Li, Y., Pan, Q., Cambria, E., 2022. Deep-attack over the deep reinforcement learning. Knowl. Based Syst. 250, 108965. doi:10.1016/j.knosys.2022.108965.

Lin, Y., Hong, Z., Liao, Y., Shih, M., Liu, M., Sun, M., 2017. Tactics of adversarial attack on deep reinforcement learning agents. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. ijcai.org, pp. 3756–3762. doi:10.24963/ijcai.2017/525.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. In: 6th International Conference on Learning Representations. OpenReview.net. https://openreview.net/forum?id=rJzIBfZAb

Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning. PMLR, pp. 1928–1937. http://proceedings.mlr.press/v48/mniha16.html

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Human-level control through deep reinforcement learning. Nature 518, 529–533. doi:10.1038/nature14236.

Mo, K., Tang, W., Li, J., Yuan, X., 2023. Attacking deep reinforcement learning with decoupled adversarial policy. IEEE Trans. Dependable Secur. Comput. 20, 758–768. doi:10.1109/TDSC.2022.3143566.

Ni, Z., O'Neill, M., Liu, W., et al., 2022. A high-performance SIKE hardware accelerator. IEEE Trans. Very Large Scale Integr. Syst. 30, 803–815. doi:10.1109/TVLSI.2022.3152011.

Rakhsha, A., Radanovic, G., Devidze, R., Zhu, X., Singla, A., 2020. Policy teaching via environment poisoning: training-time adversarial attacks against reinforcement learning. In: Proceedings of the 37th International Conference on Machine Learning. PMLR, pp. 7974–7984. http://proceedings.mlr.press/v119/rakhsha20a.html

Sayed, M.A., Anwar, A.H., Kiekintveld, C., Bosansky, B., Kamhoua, C., 2023. Cyber deception against zero-day attacks: a game theoretic approach. In: Lecture Notes in Computer Science. Springer, pp. 44–63. doi:10.1007/978-3-031-26369-9_3.

Song, L., Shokri, R., Mittal, P., 2019. Membership inference attacks against adversarially robust deep learning models. In: 2019 IEEE Security and Privacy Workshops (SPW). IEEE, pp. 50–56. doi:10.1109/SPW.2019.00021.

Su, J., Vargas, D.V., Sakurai, K., 2019. One pixel attack for fooling deep neural networks. IEEE Trans. Evol. Comput. 23, 828–841. doi:10.1109/tevc.2019.2890858.

Sun, J., Zhang, T., Xie, X., Ma, L., Zheng, Y., Chen, K., Liu, Y., 2020. Stealthy and efficient adversarial attacks against deep reinforcement learning. In: The 34th AAAI Conference on Artificial Intelligence, pp. 5883–5891. doi:10.1609/aaai.v34i04.6047.

Tai, L., Paolo, G., Liu, M., 2017. Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 31–36. doi:10.1109/iros.2017.8202134.

Wenger, E., Passananti, J., Bhagoji, A.N., Yao, Y., Zheng, H., Zhao, B.Y., 2021. Backdoor attacks against deep learning systems in the physical world. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6206–6215. doi:10.1109/CVPR46437.2021.00614.

Zhang, X., Ma, Y., Singla, A., Zhu, X., 2020. Adaptive reward-poisoning attacks against reinforcement learning. In: Proceedings of the 37th International Conference on Machine Learning, pp. 11225–11234. http://proceedings.mlr.press/v119/zhang20u.html

Zizzo, G., Hankin, C., Maffeis, S., Jones, K., 2019. Adversarial machine learning beyond the image domain. In: Proceedings of the 56th Annual Design Automation Conference 2019, pp. 1–4. doi:10.1145/3316781.3323470.

**Xiangjuan Li** is a postgraduate with the school of Cyberspace Security at Northwestern Polytechnical University, Xi'an, China. After receiving her bachelor of engineering degree in 2021, she continued studying for a master's degree with the school of Cyberspace Security at Northwestern Polytechnical University. Her research interests include adversarial attack and defense on deep reinforcement learning, Unmanned Aerial Vehicle(UAV) information security.

**Yang Li** is an associate professor with the school of automation at Northwestern Polytechnical University, Xi'an, China. After receiving his bachelor's and doctoral degrees from Northwestern Polytechnical University in 2014 and 2018 respectively, he worked as a research fellow in Sentic Team under Professor Erik Cambria at Nanyang Technological University in Singapore and also was an adjunct research fellow at the A *STAR High-Performance Computing Institute (IHPC). His research goal is to build a trustworthy AI system in the real application, and his research interests are in Natural Language Processing, Machine Learning, Recommender systems, Explainable Artificial Intelligence, etc. He has published several papers on these topics at international conferences and peer-reviewed journals. He is an active reviewer of several journals, e.g., INFORM FUSION, IEEE TAFF, NEUCOM, KBS, KAIS, etc. He also is an guest editor of Future Generation Computer Systems.

**Zhaowen Feng** is a senior engineer and a tierone expert of the Aviation Industry Corporation of China. He works as the associate chief engineer in the Information Security Institute of the Aviation Industry Development Center. His research interest includes but not limited in network security attack and defense, systems penetration testing, unmanned aerial vehicle (UAV) information security evaluation and industrial control systems security analysis. He received the bachelor degree and M.S. degree in Electrical Engineer from Beihang University and Chinese Aeronautical Establishment of China in 2010 and 2013, respectively. He currently also works on his PhD program in Cyber Security with Northwestern Polytechnical University (NPU) in China.

**Zhaoxuan Wang** received the BS degree from the School of Cybersecurity, Northwestern Polytechnical University, China, in 2020. He is working toward the Ph.D. degree from the School of Cybersecurity, Northwestern Polytechnical University. His research interests include unmanned aerial system security, autonomous driving security and artificial intelligence security.

**Quan Pan** was born in China, in 1961. He received the BS degree in automatic control from the Huazhong University of Science and Technology, in1982, and the MS and PhD degrees in control theory and application from Northwestern Polytechnical University. From 1991 to 1993, he was an Associate Professor at Northwestern Polytechnical University, where he has been a Professor with the Automatic Control Department, since 1997. He has authored 11 books, more than 400 articles. His research interests include information fusion, target tracking and recognition, deep network and machine learning, UAV detection navigation and security control, polarization spectral imaging and image processing, industrial control system information security, commercial password applications, and modern security technologies. He is an Associate Editor of the journal Information Fusion and Modern Weapons Testing Technology.