


SURVEY

Open Access

Adversarial attack and defense in reinforcement learning—from AI security view



Tong Chen , Jiqiang Liu, Yingxiao Xiang, Wenjia Niu^{*}, Endong Tong and Zhen Han

Abstract

Reinforcement learning is a core technology for modern artificial intelligence, and it has become a workhorse for AI applications ranging from *Atrai Game* to *Connected and Automated Vehicle System* (CAV). Therefore, a reliable RL system is the foundation for the security critical applications in AI, which has attracted a concern that is more critical than ever. However, recent studies discover that the interesting attack mode *adversarial attack* also be effective when targeting neural network policies in the context of reinforcement learning, which has inspired innovative researches in this direction. Hence, in this paper, we give the very first attempt to conduct a comprehensive survey on adversarial attacks in reinforcement learning under AI security. Moreover, we give briefly introduction on the most representative defense technologies against existing adversarial attacks.

Keywords: Reinforcement learning, Artificial intelligence, Security, Adversarial attack, Adversarial example, Defense

Introduction

Artificial intelligence (AI) is providing major breakthroughs in solving the problems that have withstood many attempts of natural language understanding, speech recognition, image understanding and so on. The latest studies (He et al. 2016) show that the correct rate of image understanding can reach 95% under certain conditions, meanwhile the success rate of speech recognition can reach 97% (Xiong et al. 2016).

Reinforcement learning (RL) is one of the main techniques that can realize artificial intelligence (AI), which is currently being used to decipher hard scientific problems at an unprecedented scale.

To summarized, the researches of reinforcement learning under artificial intelligence are mainly focused on the following fields. *In terms of autonomous driving* (Shalev-Shwartz et al. 2016; Ohn-Bar and Trivedi 2016), Shai et al. applied deep reinforcement learning to the problem of forming long term driving strategies (Shalev-Shwartz et al. 2016), and solved two major challenges in self driving. *In the aspect of game play* (Liang et al. 2016), Silver et al. (2016) introduced a new approach to computer Go which can evaluate board positions, and select the

best moves with reinforcement learning from games of self-play. Meanwhile, for Atari game, Mnih et al. (2013) presented the first deep learning model to learn control policies directly from high-dimensional sensory input using reinforcement learning. Moreover, Liang et al. (Guo et al. 2014) also built a better real-time Atrai game playing agent with DQN. *In the field of control system*, Zhang et al. (2018) proposed a novel load shedding scheme against voltage instability with deep reinforcement learning (DRL). Bougiouklis et al. (2018) presented a system for calculating the optimum velocities and the trajectories of an electric vehicle for a specific route. In addition, *in the domain of robot application* (Goodall and El-Sheimy 2017; Martínez-Tenor et al. 2018), Zhu et al. (2017) applied their model to the task of target-driven visual navigation. Yang et al. (Yang et al. 2018) presented a soft artificial muscle driven robot mimicking cuttlefish with a fully integrated on-board system.

In addition, reinforcement learning is also an important technique for *Connected and Automated Vehicle System* (CAV), which is a hotspot issue in recent years. Meanwhile, the security research for this direction has attracted numerous concerns (Chen et al. 2018a; Jia et al. 2017). Chen et al. performed the first security analysis on the next-generation Connected Vehicle (CV) based transportation systems, and pointed out the current signal control algorithm design and implementation choices

*Correspondence: niuwj@bjtu.edu.cn
Beijing Key Laboratory of Security and Privacy in Intelligent Transportation,
Beijing Jiaotong University, Beijing, China

are highly vulnerable to data spoofing attacks from even a single attack vehicle. Therefore, how to build a reliable and security reinforcement learning system to support the security critical applications in AI, has become a concern which is more critical than ever.

However, the weaknesses of reinforcement learning are gradually exposed which can be exploited by attackers. Huang et al. (2017) firstly discovered that neural network policies in the context of reinforcement learning are vulnerable to “Adversarial Attacks” in the form of adding tiny perturbations to inputs which can lead a model to give wrong results. Regardless of the learned task or training algorithm, they observed a significant drop in performance, even with very small adversarial perturbations which are invisible to human. Even worse, they found that the cross-dataset transferability property (Szegedy et al. 2013 proposed in 2013) also holds in reinforcement learning applications, so long as both policies have been trained to solve the same task. Such discoveries have attracted public interests in the research of adversarial attacks and their corresponding defense technologies in the context of reinforcement learning.

After Huang et al. (2017), a lot of works have focused on the issue of adversarial attack in the field of reinforcement learning (e.g., Fig. 1). For instance, *in the field of Atari game*, Lin et al. (2017) proposed a “strategically-timed attack” whose adversarial example at each time step is computed independently of the adversarial examples at other time steps, instead of attacking a deep RL agent at every time step (see “Black-box attack” section). Moreover, *in the terms of automatic path planning*, Liu et al. (2017), Xiang et al. (2018), Bai et al. (2018) and Chen et al. (2018b) all proposed methods which can take adversarial attack on reinforcement learning algorithms

(VIN (Tamar et al. 2016), Q-Learning (Watkins and Dayan 1992), DQN (Mnih et al. 2013), A3C (Mnih et al. 2016)) under automatic path planning tasks (see “Defense technology against adversarial attack” section).

In view of the extensive and valuable applications of the reinforcement learning in modern artificial intelligence (AI), and the critical role for reinforcement learning in AI security, inspiring innovative researches in the field of adversarial research.

The main contributions of this paper can be concluded as follows:

- 1 We give the very first attempt to conduct a comprehensive and in-depth survey on the literatures of adversarial research in the context of reinforcement learning from AI security view.
- 2 We make a comparative analysis for the characteristics of adversarial attack mechanisms and defense technologies respectively, to compare the specific scenarios and advantages/disadvantages of the existing methods, in addition, give a prospect for the future work direction.

The structure of this paper is organized as follow. In “Preliminaries” section, we first give a description for the common term related to adversarial attack under reinforcement learning, and briefly introduce the most representative RL algorithms. “Adversarial attack in reinforcement learning” section reviews the related research of adversarial attack in the context of reinforcement learning. For the defense technologies against adversarial attack in the context of reinforcement learning are discussed in “Defense technology against adversarial attack” section. Finally, we draw conclusion and discussion in “Conclusion and discussion” section.

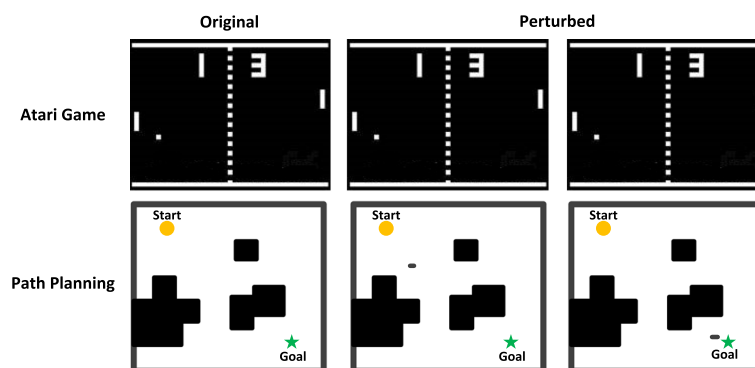


Fig. 1 Examples for adversarial attacks on reinforcement learning. As shown in the first line are the examples for adversarial attack in the field of Atari game. The first image denotes the original clean game background, while the others show the perturbed game background which can be called as “adversarial example”. Huang et al. (2017) found that the adversarial examples which are invisible to human have a significant impact on the game result. Moreover, the second line shows the examples for adversarial attack in the domain of automatic path planning. Same as the first row, the first image represents the original pathfinding map, and the remaining two images denote the adversarial examples generated by noise added. Chen et al. (2018b) found that the trained agent could not find its way correctly under such adversarial examples

Preliminaries

In this section, we give explanation for the common terms related to adversarial attack in the field of reinforcement learning. In addition, we also briefly introduce the most representative reinforcement learning algorithms, and take comparison of these algorithms from *approach type*, *learning type*, and *application scenarios*. So as to facilitate readers' understanding of the content for the following sections.

Common terms definitions

- **Reinforcement Learning**: is an important branch of machine learning, which contains two basic elements *state* and *action*. Performing a certain action under the certain state, what the agent need to do is to continuously explore and learn, so as to obtain a good strategy.
- **Adversarial Example**: Deceiving AI system which can lead them make mistakes. The general form of adversarial examples is the information carrier (such as image, voice or txt) with small perturbations added, which can remain imperceptible to human vision system.
 - 1 **Implicit Adversarial Example**: is a modified version of clean information carrier, which generated by adding human invisible perturbations to the global information on pixel level to confuse/fool a machine learning technique.
 - 2 **Dominant Adversarial Example**: is a modified version of clean map, which generated by adding physical-level obstacles to change the local information to confuse/fool A3C path finding.
- **Adversarial Attack**: Attacking on artificial intelligence (AI) system by utilizing adversarial examples. Adversarial attacks are generally can be classified into two categories:
 - 1 **Misclassification attacks**: aiming for generating adversarial examples which can be misclassified by target network.
 - 2 **Targeted attacks**: aiming for generating adversarial examples which can target misclassifies into an arbitrary label designated by adversary specially.
- **Perturbation**: The noise added on the original clean information carriers (such as image, voice or txt), which can make them to be adversarial examples.
- **Adversary**: The agent who attack AI system with adversarial examples. However, in some cases, it also refer to adversarial example itself (Akhtar and Mian 2018).
- **Black-Box Attack**: The attacker has no idea of the details related to training algorithm and corresponding parameters of the model. However, the attacker can still interact with the model system, for instance, by passing in arbitrary input to observe changes in output, so as to achieve the purpose of attack. In some work (Huang et al. 2017), for black-box attack, authors assume that the adversary has access to the training environment (e.g., the simulator) but not the random initialization of the target policy, and additionally may not know what the learning algorithm is.
- **White-Box Attack**: The attacker has access to the details related to training algorithm and corresponding parameters of the model. Attacker can interact with the target model in the process of generating adversarial attack data.
- **Threat Model**: Finding system potential threat to establish an adversarial policy, so as to achieve the establishment of a secure system (Swiderski and Snyder 2004). In the context of adversarial research, threat model considers adversaries capable of introducing small perturbations to the raw input of the policy.
- **Transferability**: an adversarial example designed to be misclassified by one model is often misclassified by other models trained to solve the same task (Szegedy et al. 2013).
- **Target Agent**: The target subject attacked by adversarial examples, usually can be a network model trained by reinforcement learning policy, which can detect whether adversarial examples can attack successfully.

Representative reinforcement learning algorithms

In this section, we list the most representative reinforcement learning algorithms, and make comparison among them which can be shown in Table 1, where "value-based" denotes that the reinforcement learning algorithm calculates the expected reward of actions under potential rewards, and takes it as the basis for selecting actions. Meanwhile, the learning strategy for "value-based" reinforcement learning is constant, in other words, under the certain state the action will be fixed.

While the "policy-based" represented that the reinforcement learning algorithm trains a probability distribution by strategy sampling, and enhances the probability of selecting actions with high reward value. This kind of reinforcement learning algorithm will learn different strategies, in other words, the probability of taking one action under the certain state is constantly adjusted.

- **Q-Learning**

Q-Learning is a classical algorithm for reinforcement learning, was proposed earlier and has been used widely.

Table 1 The comparison of the most representation reinforcement learning algorithms

RL algorithm	Approach type	Learning type	Application scenarios
Q-Learning (Watkins and Dayan 1992)	Value-based	Shallow Learning	Motion Control, Control System, and Robot Application et al.
DQN (Mnih et al. 2013)	Value-based	Deep Learning	Motion Control, Neutralization Reaction Control, and Robot Path Planning et al.
VIN (Tamar et al. 2016)	Value-based	Deep Learning	Path Planning, and Motion Control et al.
A3C (Mnih et al. 2016)	Combined	Deep Learning	motion Control, Game Playing, self-driving, and Path Planning et al.
TRPO (Schulman et al. 2015)	Policy-based	Deep Learning	Motion Control, and Game Playing et al.
UNREAL (Jaderberg et al. 2016)	Combined	Deep Learning	Motion Control, and Game Playing et al.

Approach Type contains two categories, namely *Policy-based*, and *Value-based*. Meanwhile, *learning Type* also contains two categories, namely *Shallow Learning*, and *Deep Learning*

Q-Learning was firstly proposed by C.Watkins (Watkins and Dayan 1992) in his Doctoral Dissertation Learning from delayed rewards in 1989. It is actually a variant of *Markov Decision Process (MDP)*(Markov 1907). The idea of Q-Learning is based on the value iteration, which can be concluded as, the agent perceives surrounding information from the environment and selects appropriate methods to change the state of environment according to its own method, and obtains corresponding incentives and penalties to correct the strategy. Q-Learning proposes a method to update the Q-value, which can be concluded as $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \lambda \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$. Throughout the continuous iteration and learning process, the agent tries to maximize the rewards it receives and finds the best path to the goal, and the Q matrix can be obtained. Q is an action utility function that evaluates the strengths and weakness of actions in a particular state and can be interpreted as the brain of an intelligent agent.

- **Deep Q-Network (DQN)**

DQN is the first deep enhancement learning algorithm proposed by Google DeepMing in 2013 (Mnih et al. 2013) and further improved in 2015 (Mnih et al. 2015). DeepMind applies DQN to Atari games, which is different from the previous practice, utilizing the video information as input and playing games against humans. In this paper, authors gave the very first attempt to introduce the concept of *Deep Reinforcement Learning*, and has attracted public attentions in this direction. For DQN, as the output for the value network is the Q-value, then if the target Q-value can be constructed, the loss function can be obtained by *Mean-Square Error (MSE)*. However, the input for value network are state S , action A , and feedback reward R . Therefore, how to calculate the target Q-value correctly is the key problem in the context of DQN.

- **Value Iterative Network (VIN)**

Tamar et al. (2016) proposed the value iteration network, a fully differentiable CNN planning module for approximate value iterative algorithms that can be used for learning to plan, such as the strategies in reinforcement learning. This paper mainly solved the problem of weak generalization ability of deep reinforcement learning. There is a special value iterative network structure in VIN (Touretzky et al. 1996). For this novel method proposed in this work, it not only needs to use neural network to learn a direct mapping from state to decision, but also can embeds the traditional planning algorithm into the neural network so that the neural network can learn how to act under current environment, and use long-term planning-assisted neural networks to give a better decision.

- **Asynchronous Advantage Actor-Critic Algorithm (A3C)**

The A3C algorithm is a deep enhancement learning algorithm proposed by DeepMind in 2016 (Mnih et al. 2016). A3C completely utilizes the Actor-Critic framework and introduces the idea of asynchronous training, which can improve the performance and speeds up the whole training process. If the action is considered to be bad, the possibility for this action will be reduced. Through iterative training, A3C constantly adjusts the neural network to find the best action selected policy.

- **Trust Region Policy Optimization (TRPO)**

TRPO is proposed by J.Schulman in 2015 (Schulman et al. 2015), it is a kind of random strategy search method in strategy search method. TRPO can solve the problem of step selection of gradient update, and gives a monotonous strategy improvement method. For each training iterative, whole-trajectory rollouts of a stochastic policy are used to calculate the update to the policy parameters θ , while

controlling the change in policy as measured by the KL divergence between the old and the new policies.

- **UNREAL**

The UNREAL algorithm is the latest depth-enhancement learning algorithm proposed by DeepMind in 2016 (Jaderberg et al. 2016). Based on the A3C algorithm, the performance and training process for this algorithm are further improved. The experimental results show that the performance for UNREAL at Atari is 8.8 times against human performance and 3D at the first perspective, moreover, UNREAL has reached 87% of human level in the first-view 3D maze environment Labyrinth. For UNREAL, there are two types of auxiliary tasks, the first one is the control task, including pixel control and hidden layer activation control. The other one is back prediction tasks, as in many scenarios feedback r is not always available, allowing the neural network to predict the feedback value will give it a better ability to express. UNREAL algorithm uses historical continuous multi-frame image input to predict the next-step feedback value as a training target and uses history information to additionally increase the value iteration task.

Adversarial attack in reinforcement learning

In this section, we discuss the related research of adversarial attack in the field of reinforcement learning. The reviewed literatures mainly conduct the adversarial research on specific application scenarios, and generate adversarial examples by adding perturbations to the information carrier, so as to realize the adversarial attack on reinforcement learning system.

We organize the review mainly according to chronological order. Meanwhile, in order to make readers can understand

the core technical concepts of the surveyed works, we go into technical details of important methods and representative technologies by referring to the original papers. In part 3.1, we discuss the related works of adversarial attack against the reinforcement learning system in the domain of White-box attacking. In terms of Black-box attacking, the design of adversarial attack against the target model is shown in part 3.2. Meanwhile, we analyze the availability and contribution of adversarial attack researches in the above two fields. Additionally, we also give summary on the attributions of adversarial attacking methods discussed in this section in part 3.3.

White-box attack

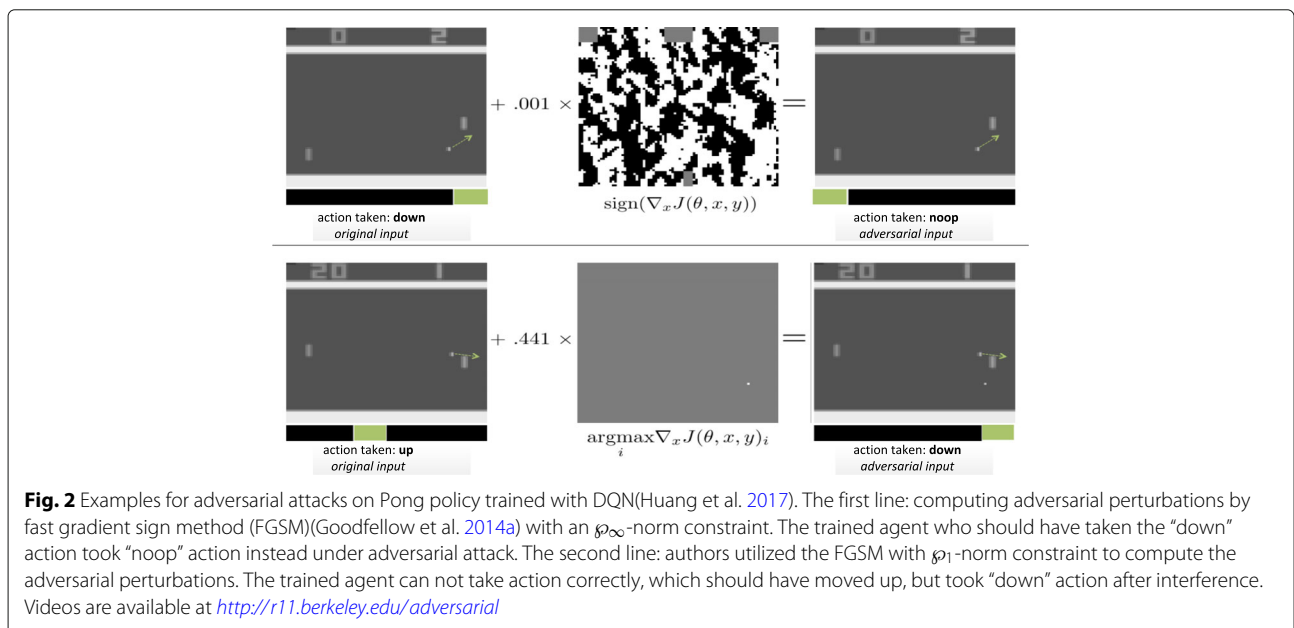
Fast gradient sign method (FGSM)

Huang et al. (2017) first showed that adversarial attacks are also effective when targeting neural network policies in reinforcement learning system. Meanwhile, for this work, the adversary attacks a deep RL agent at every time step, by perturbing each image the agent observes.

The main contributions for Huang et al. (2017) can be concluded as the following two aspects:

- (I) They gave the very first attempt to prove that reinforcement learning systems are vulnerable to adversarial attack, and the traditional generation algorithms designed for adversarial examples still can be utilized to attack under such scenario.
- (II) Authors creatively verified how effectiveness of adversarial examples are impacted by the deep RL algorithm used to learn the policy.

Figure 2 shows the adversarial attack on Pong game trained with DQN, we can see that after adding small



perturbation to the original clean game background, the trained agent cannot make a correct judgment according to the motion direction of ball. Noting that the adversarial examples are calculated by fast gradient sign method (FGSM) (Goodfellow et al. 2014a).

FGSM expects the classifier can assign the same class to the real example x and the adversarial example \tilde{x} with a small enough perturbation η which can be concluded as

$$\eta = \epsilon \text{sign}(\omega) , \|\eta\|_{\infty} < \epsilon \quad (1)$$

where ω denotes a weight vector, since this perturbation maximizes the change in output for the adversarial example \tilde{x} , $\omega^T \tilde{x} = \omega^T x + \omega^T \eta$.

Moreover, under image classification network with parameters θ , model input x , targets related to input y , and cost function $J(\theta, x, y)$. Linearizing the cost function to obtain an optimal max-norm constrained perturbation which can be concluded as

$$\eta = \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \quad (2)$$

In addition, authors also proved that policies trained with reinforcement learning are vulnerable to the adversarial attack. However, among the RL algorithms tested in this paper (DQN, TRPO (Schulman et al. 2015), and A3C), TRPO and A3C seem to be more resistant to adversarial attack.

Under the domain of Atari game, authors showed that by adding human invisible noises to the original clean game background can make the game unable to work properly, and realize adversarial attack successfully. Huang et al. (2017) gave a new attempt to take adversarial research under the scenario of reinforcement learning, and this work proved that the adversarial attack still exists in the domain of reinforcement learning. Moreover, FGSM motivates a series of related research work, Miyato et al. (2018) proposed a closely related mechanism to compute the perturbation for a given image, and Kurakin et al. (2016) named this algorithm as ‘‘Fast Gradient L_2 ’’ and also proposed an alternative of using ℓ_{∞} for normalization which named as ‘‘Fast Gradient L_{∞} ’’.

Start point-based adversarial attack on Q-learning (SPA)

Xiang et al. (2018) focused on the adversarial example-based attack on a representative reinforcement learning named Q-learning in automatic path finding. They proposed a probabilistic output model based on the influence factors and the corresponding weights to predict the adversarial examples under such scenario.

Calculating on four factors including the energy point gravitation, the key point gravitation, the path gravitation, and the included angle, a natural linear model is constructed to fit these factors with the weight parameters computation based on the principal component analysis (PCA) (Wold et al. 1987).

The main contribution for Xiang et al. is that they built a model, which can generate the corresponding probabilistic outputs for certain input points, and the probabilistic output of our model refers to the possibility of interference caused by interference point on the path of agent pathfinding.

Xiang et al. proposed 4 factors to determine whether the perturbation can impact the final result for the agent path planning, which can be concluded as:

Factor	Formula expression
Factor 1: The energy point gravitation	$\begin{cases} e_{ic} = k_c + i * d' * \frac{k'_c - k_c}{\sqrt{(k'_c - k_c)^2 + (k'_r - k_r)^2}} \\ e_{ir} = k_r + i * d' * \sqrt{1 - \left(\frac{k'_c - k_c}{\sqrt{(k'_c - k_c)^2 + (k'_r - k_r)^2}}\right)^2} \end{cases}$
Factor 2: The key point gravitation	$d_{1i} = a_{ic} - k_c + a_{ir} - k_r $ $(k_c, k_r) = k, (a_{ic}, a_{ir}) = a_i \in A$
Factor 3: The path gravitation	$d_{2i} = \min\{d_2 d_2 = a_{ic} - z_{jc} + a_{ir} - z_{jr} , z_j \in Z_1\}, (z_{jc}, z_{jr}) = z_j, (a_{ic}, a_{ir}) = a_i \in A$
Factor 4: The included angle	$v_{ka} = (a_{ic} - k_c, a_{ir} - k_r), v_{kt} = (t_c - k_c, t_r - k_r)$ $\cos \theta_i = v_{ka} \cdot v_{kt} / v_{ka} v_{kt} , \theta_i = \arccos \theta_i$

For Factor 1 can be named as *the energy point gravitation*, which denotes that it is more successful if the adversarial point k is the point on the key vector v . Factor 2 is *the key point gravitation*, which represents that the closer adversarial point is to the key point k , the more likely it is to cause interference. Factor 3 can be called as *the path gravitation*, which denotes that the closer adversarial point is to the initial path Z_1 , the more possible it is to bring about obstruct. Meanwhile, factor 4 can be concluded as *the included angle*, which represents that the angle θ between the vector from the point k to the adversarial point a_i and the vector from the key point to the goal t .

Therefore, the probability for each adversarial point a_i can be concluded as

$$p_{a_i} = \sum_{j=1}^4 p_{j a_i} = \omega_1 \cdot a_{ie} + \omega_2 \cdot d'_{1i} + \omega_3 \cdot d'_{2i} + \omega_4 \cdot \theta'_i \quad (3)$$

where ω_i denotes the weight for each factor respectively. Storing the p_{a_i} for each point, and select the top 10 as the adversarial point.

For this work, the adversarial examples can be found successfully for the first time on Q-learning in path finding and their model can make a satisfactory prediction (e.g., Fig. 3). Under a guaranteed recall, the precision of the proposed model can reach to 70% with the proper parameter setting. By adding small obstacle points to the original clean map, can interfere the agent’s path finding. However, the experimental map size for this work is 28×28 , and there is no additional verification for a larger maze map, which can be considered to research in future works. However, Xiang et al. paid attention to the adversarial attack problem in automatic path finding under the scenario of reinforcement learning. Meanwhile this work own practical significance, as the objective for this study is Q-learning which is the most widely used and representative reinforcement learning algorithm.

White-box based adversarial attack on DQN (WBA)

Based on the SPA algorithm introduced above, Bai et al. (2018) proposed that they first use DQN to find the optimal path, and analyzed the rules of DQN pathfinding. They proposed a method that can effectively find vulnerable points towards White-Box Q-table variation in DQN pathfinding training. Meanwhile, they built a simulation environment as a basic experiment platform to test their method.

Moreover, they classified two types of vulnerable points.

- (I) The vulnerable point is most likely on the boundary line. Moreover, the smaller ΔQ (the Q-value difference between the right and downward direction) is the more likely be a vulnerable point is.

For this characteristic of vulnerable pints, they proposed a method to detect adversarial examples. Let P denotes the set of points on the map $P = \{P_1, P_2, \dots, P_n\}$, and each point P_i obtains four Q-values $D_{ij} = (Q_{i1}, Q_{i2}, Q_{i3}, Q_{i4})$ respectively, which indicate up, down, right, and left. Meanwhile, selecting the direction with the max Q-value $f(P_i) = \{j | \max_j Q_{ij}\}$, and determining wether point P_i is on the boundary line

$$\begin{aligned} \varphi(P_i) &= OR(f(P_i) \neq f(P_{i1}), f(P_i) \neq f(P_{i2}), \\ &f(P_i) \neq f(P_{i3}), f(P_i) \neq f(P_{i4})) \end{aligned} \tag{4}$$

where $P_{ij} = \{P_{i1}, P_{i2}, P_{i3}, P_{i4}\}$ is the set of the adjoining points for four directions of P_i , $A = \{a_1, a_2, \dots, a_n\}$ represents the points on boundary line. Calculating the Q-value difference $\Delta Q = |Q_{i2} - Q_{i3}|$, and sorting ΔQ ascending to construct $B = \{b_1, b_2, \dots, b_n\}$. They took the first 3% of the list as the smallest ΔQ -value points. Finally got the set of suspected adversarial examples, which can be concluded as $X = \{x_1, x - 2, \dots, x_n\}, X = A \cap B$.

For the other type of vulnerable points can be concluded as:

- (II) Adversarial examples are related to the gradient of maximum Q-value for each point on the path.

Bai et al. found that when the Q-values of consecutive two points fluctuate greatly, their gradient is greater and they are more vulnerable to be attacked.

Meanwhile, they found that the larger angle between two adjacent lines is, the greater slope of the straight line is. Set angle between the direction vectors of two straight lines to be θ ($0 < \theta < \frac{\pi}{2}$), the function can be concluded as

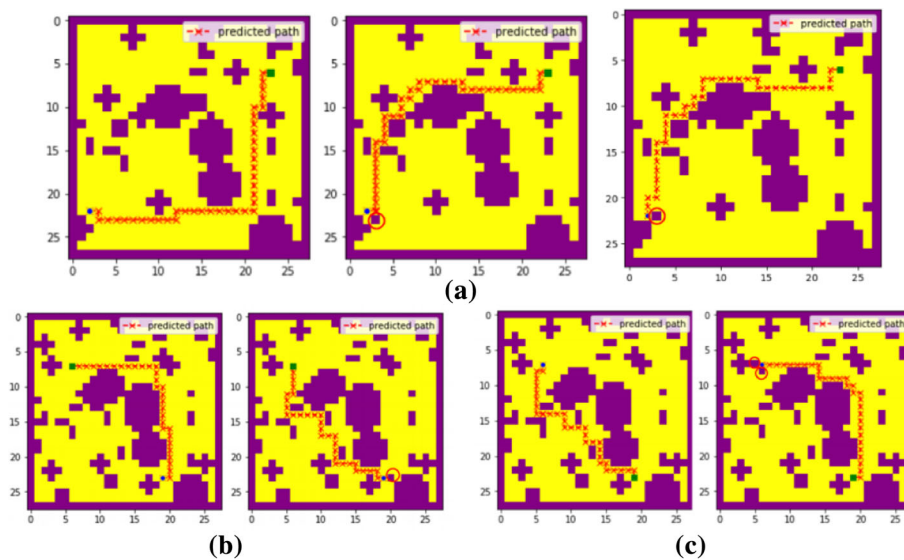


Fig. 3 An illustration of the interference effect before and after adding adversarial points when the path size is 2. We show two types of maps here, where (a) denotes the first type, and (b), (c) all belong to the second category

$$\cos \theta = \frac{|s_1 \cdot s_2|}{|s_1||s_2|} = \frac{|m_1 m_2 + n_1 n_2 + p_1 p_2|}{\sqrt{m_1^2 + n_1^2 + p_1^2} \sqrt{m_2^2 + n_2^2 + p_2^2}} \quad (5)$$

where $s_1 = (m_1, n_1, p_1), s_2 = (m_2, n_2, p_2)$ are the direction vectors for Line L_1, L_2 . Finally, can find the first large 1% of the angle between the two lines on the path as the suspected interference point.

For WBA, authors successfully found the adversarial examples and the supervised method they proposed is effective, which can be shown in Table 2 for details. However, in this work, with the increase of training times, the accuracy rate decreases. In other words, when training times are large enough, the interference point can make the path converge, although the training efficiency is reduced.

Similar to the work of Xiang et al., the maps used for experiment are 16×16 and 17×17 is size, and there is no way to verify the proposed adversarial attack method more accurately with such map size. It is recommended that the attack method can be verified on different categories of map-size, which can better illustrate the effectiveness of the proposed method in this paper.

Common dominant adversarial examples generation method (CDG)

Chen et al. (2018b) showed that dominant adversarial examples are effective when targeting A3C path finding, and designed a *Common Dominant Adversarial Examples Generation Method (CDG)* to generate dominant adversarial examples against any given map.

As shown in Fig. 4, are the dominant adversarial examples for the original map which can attack successfully.

Chen et al. found that on the dominant adversarial example perturbation band, the value gradient rises the fastest. Therefore, they call this perturbation band as “gradient band”. By adding obstacles on the cross section of gradient band can perturb the agent’s path finding successfully. The generation rule for dominant adversarial example can be defined as:

- **Generation Rule:** Adding “baffle-like” obstacles to the cross section of gradient band in which the value gradient rises the fastest, can impact A3C path finding.

Moreover, in order to calculate the Gradient Band more accurately, authors considered two kinds of situations according to the difference for original map and gradient function, one situation is that obstacles exist on both sides of the gradient function, and the other is that obstacles exist on one side if the gradient function.

A. Case 1: Obstacles exist on both sides of the gradient function.

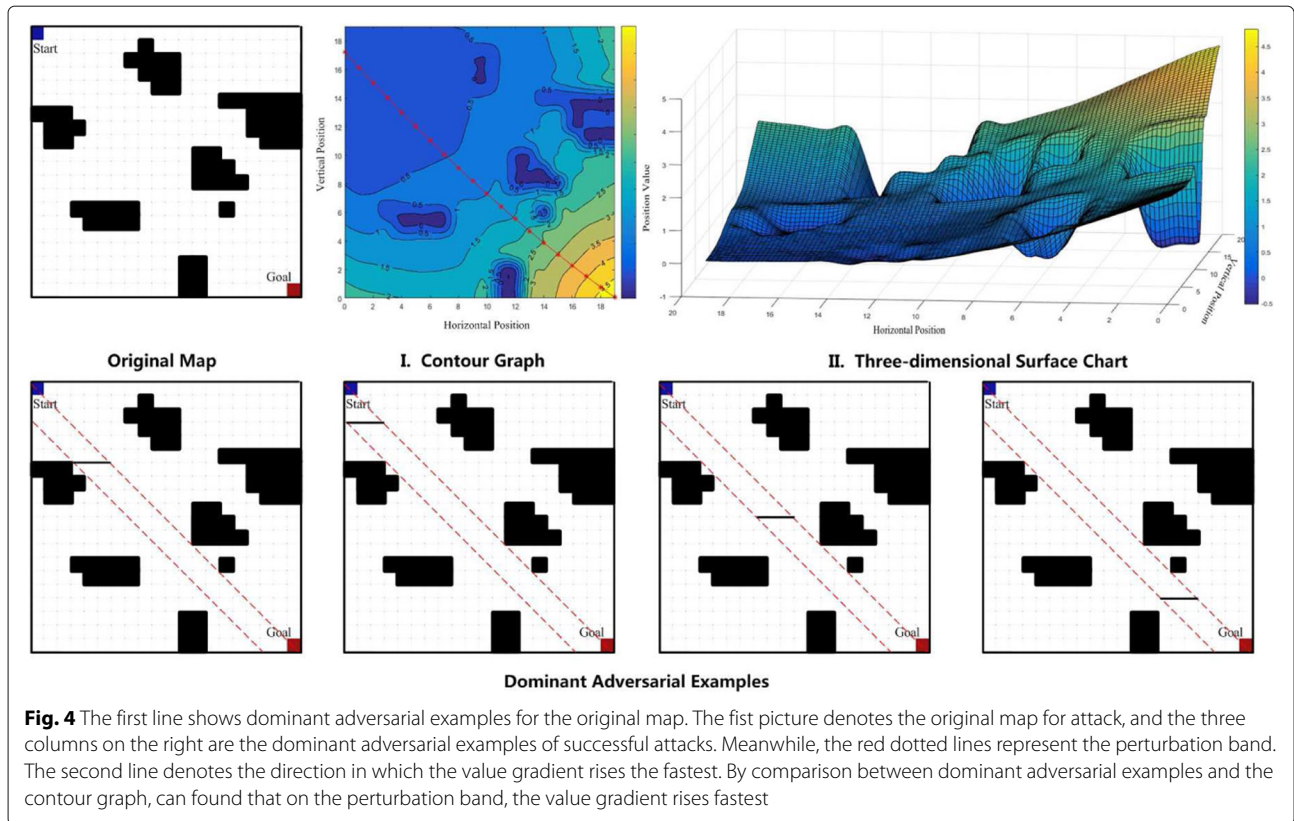
As in this case, obstacles exist on the both sides of the gradient curve, then need to traverse all the coordinate points in *Obstacle* = $\{(O_{x_1}, O_{y_1}), (O_{x_2}, O_{y_2}), \dots, (O_{x_n}, O_{y_n})\}$, and to find the nearest two points from this gradient curve in the upper and lower part respectively. Therefore, the Gradient Band function $F_{GB}(x, y)$ under such case can be concluded as:

$$\begin{cases} f(x, y)_{upper} = y - (U + a_0 + a_1 x + \dots + a_k x^k) \\ f(x, y)_{lower} = y - (L + a_0 + a_1 x + \dots + a_k x^k) \end{cases} \quad (6)$$

$$X_L < x < X_{max}, Y_L < y < Y_{max}$$

Table 2 Features for adversarial perturbations against single same original clean map, which can show how the different characteristics affect the interference of adversarial example

Number	Point coordinates	Max Q-value	Top ΔQ	On the boundary
<i>Point</i> ₁	(4,5)	90.2229	0.0198	True
<i>Point</i> ₂	(4,10)	140.7650	0.1616	True
<i>Point</i> ₃	(2,3)	60.9148	0.2214	True
<i>Point</i> ₄	(3,4)	71.4446	0.3199	True
<i>Point</i> ₅	(5,6)	109.0013	0.3595	True
<i>Point</i> ₆	(0,2)	48.4608	0.4645	True
<i>Point</i> ₇	(6,7)	126.3412	0.6992	True
Number	On the path	Top angle size	Angle size	Perturbation point
<i>Point</i> ₁	True	\emptyset	74°	True
<i>Point</i> ₂	False	\emptyset	\emptyset	True
<i>Point</i> ₃	True	\emptyset	75°	True
<i>Point</i> ₄	True	3	69°	True
<i>Point</i> ₅	True	1	84°	True
<i>Point</i> ₆	True	\emptyset	71°	True
<i>Point</i> ₇	True	2	77°	False



where $f(x, y)_{upper}$ and $f(x, y)_{lower}$ denote the upper/lower bound function respectively, X_{max} and Y_{max} denote the boundary value of the map, $(X_L, 0)$ and $(0, Y_L)$ are the intersection points of $f(x, y)_{lower}$ and the coordinate axis.

A. Case 2: Obstacles exist on one side of the gradient function.

In this case, the calculating for distance between obstacle edge points and gradient function is same with case 1. However, under such scenario, obstacles exist on one side of the gradient function curve, hence, under this case can only obtain the upper/lower bound function for the Gradient Band. Therefore, the Gradient Band function $F_{GB}(x, y)$ can be concluded as:

$$\begin{cases} f(x, y)_{upper} = \min\{f(X_{max}, 0), f(0, Y_{max})\} \\ f(x, y)_{lower} = y - (L + a_0 + a_1x + \dots + a_kx^k) \\ X_L < x < X_{max}, Y_L < y < Y_{max} \end{cases} \quad (7)$$

Finally setting $Y = [1, 2, \dots, Y_{max}]$ and $X = [1, 2, \dots, X_{max}]$ respectively, and generating the obstacle function set $O_{baffle} = \{F_{Y_1}, \dots, F_{X_1}, \dots\}$

For this paper, the lowest generation precision for CDG algorithm is 91.91% (e.g., Fig. 5), which can prove that the method proposed in this work can realize the common

dominant adversarial examples generated under A3C path finding with a high confidence.

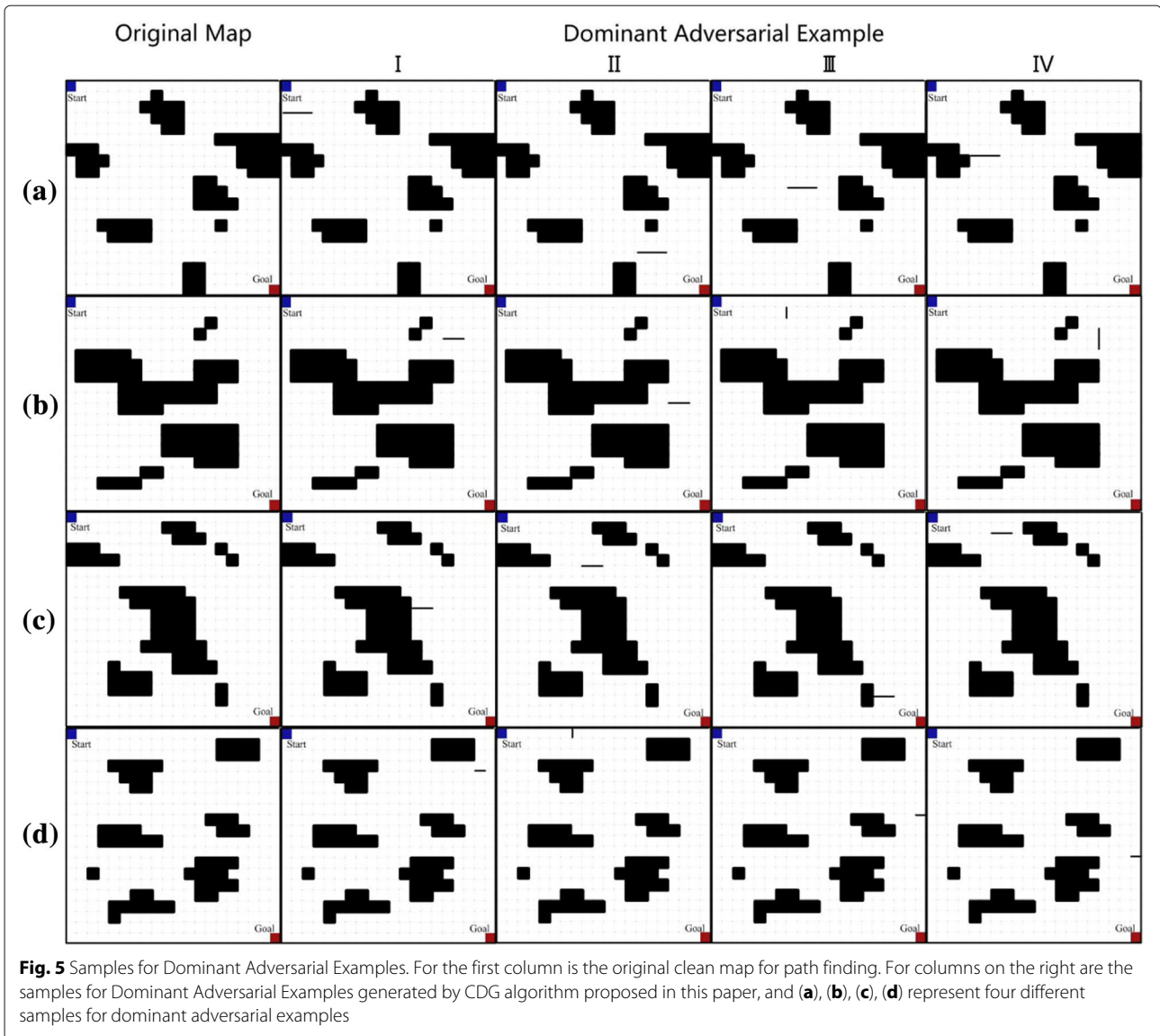
This paper showed that, the generation accuracy for adversarial examples of CDG algorithm is relatively high. By adding small obstacles at physical level on the original clean map, it will interfere with the path finding process of A3C agent. Comparing to other works in this field, the experimental map size for Chen’s work contains 10 categories, $10 \times 10, 20 \times 20, 30 \times 30, 40 \times 40, 50 \times 50, 60 \times 60, 70 \times 70, 80 \times 80, 90 \times 90, 100 \times 100$, which makes it possible to better verify the effectiveness of the proposed CDG algorithm proposed in this paper.

Black-box attack

Policy induction attack (PIA)

Behzadan and Munir (2017) also discover that Deep Q-network(DQN) based policy is vulnerability under adversarial perturbations, and verified that the transferability (Szegedy et al. (2013) proposed in 2013) of adversarial examples across different DQN model does exist.

Therefore, they proposed a new type of adversarial attack named *policy induction attack* based on this vulnerability of DQN. Their threat model considers that adversary can get limited priori information, reward function R and an estimate for the update frequency of the target network. In other words, adversary is not aware of target’s



network architecture and its parameters at every time step, adversarial examples must be generated by black-box techniques (Papernot et al. 2016c).

For every time step, adversary computes the perturbation vectors $\hat{\delta}_{t+1}$ for the next state s_{t+1} such that $\max_{a'} \hat{Q}(s_{t+1} + \hat{\delta}_{t+1}, a'; \theta_t^-)$ causes \hat{Q} to generate its maximum when $a' = \pi_{adv}^*(s_{t+1})$. The whole process for policy induction attack can be divided into two parts, namely initialization and exploitation.

The initialization phase must be done before target starts interacting with the environment. Specifically, this phase can be divided as follow:

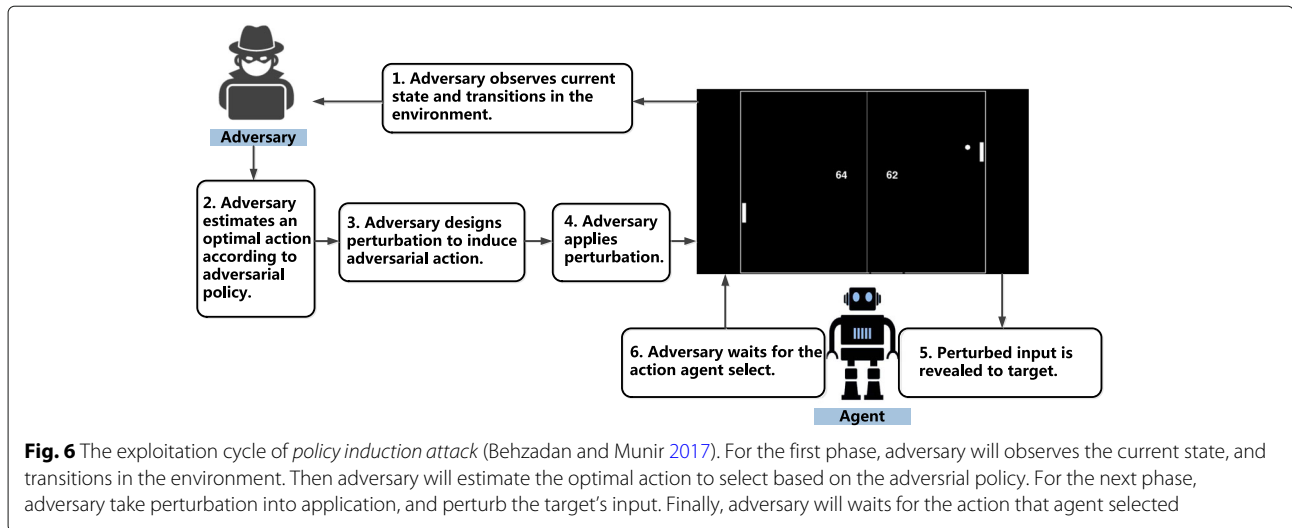
- 1) Training DQN policy based on the adversary's reward function r' to obtain an adversarial strategy π_{adv}^* .
- 2) Creating a replica of the target's DQN and initializing it with random parameters.

The exploitation phase takes adversarial attack operations (e.g., designing adversarial input), and constitutes the life cycle which can be shown in Fig. 6. The cycle is initialized by the first observation value of the environment, and to cooperate with the operation of the target agent.

In the context of policy induction attacks, this paper conjectured that the temporal features of the training process may be utilize to provide protection mechanisms. However, an analytical treatment of the problem to establish the relationship of model parameters will suggest a deeper insight and guidelines into design a more security deep reinforcement learning architecture.

Specific time-step attack

As the uniform attack strategies (e.g. Huang et al. (2017)) can be regarded as a direct extension of the adversarial attack in DNN-based classification system, since the



adversarial example at each time step is computed independently of the adversarial examples at other time step. However, such tactic has not consider the uniqueness of the RL problem.

Lin et al. (2017) proposed two tactics of adversarial attack in the specific scenario of reinforcement learning problem, which namely *strategically-time attack* and the *enchanted attack*.

- **Strategically-Timed Attack (STA)**

As the reward signal in many RL problems is sparse, an adversary need not attack the RL agent at every time step. Therefore, this adversarial attack tactic utilizes this unique characteristic to attack selected subset of time steps of RL agents. The core of strategically-timed attack is that the adversary can minimize the expected accumulated reward of target agent by strategically attacking less than $\Gamma \ll L$ time steps, to achieve the purpose of adversarial attack, which can be formulated intuitively as an optimization problem

$$\begin{aligned} & \min_{b_1, b_2, \dots, b_L, \delta_1, \delta_2, \dots, \delta_L} R_1(\bar{s}_1, \dots, \bar{s}_L) \\ & \bar{s}_t = s_t + b_t \delta_t \text{ for all } t = 1, \dots, L \\ & b_t \in \{0, 1\}, \text{ for all } t = 1, \dots, L \\ & \sum_t b_t \leq \Gamma \end{aligned} \tag{8}$$

where s_1, \dots, s_L denotes the sequence of observations or states, $\delta_1, \dots, \delta_L$ is the sequence of perturbations, R_1 represents the expected return at the first time step, b_1, \dots, b_L denotes when an adversarial example is applied, and the Γ is a constant to limit the total number of attacks.

However, the optimization problem in 8 is a mixed integer programming problem, which is difficult to solve.

Hence, authors proposed a *heuristic algorithm* to solve this task, with a relative action preference function c , which computes the preference of the agent in taking the most preferred action over the least preferred action at the current state (similar to Farahmand (2011)).

For policy gradient-based methods such as A3C algorithm, Lin et al. defined the function c as

$$c(s_t) = \max_{a_t} \pi(s_t, a_t) = \min_{a_t} \pi(s_t, a_t) \tag{9}$$

where s_t denotes the state at time step t , and a_t denotes the action at time step t , and π is the policy network which maps the state-action pair (s_t, a_t) to a probability.

Meanwhile, for value-based methods such as DQN, the function c can be defined as

$$c(s_t) = \max_{a_t} \frac{e^{\frac{Q(s_t, a_t)}{T}}}{\sum_{a_k} e^{\frac{Q(s_t, a_k)}{T}}} - \min_{a_t} \frac{e^{\frac{Q(s_t, a_t)}{T}}}{\sum_{a_k} e^{\frac{Q(s_t, a_k)}{T}}} \tag{10}$$

where Q denotes the Q-values of actions, and T denotes the temperature constant.

- **Enchanting Attack (EA)**

The purpose for enchanted attack is to push the RL agent to achieve the expected state s_g after H steps under the current state s_t at time step t . Under such attacking approach, the adversary needs to specially design a series of adversarial examples $s_{t+1} + \delta_{t+1}, \dots, s_{t+H} + \delta_{t+H}$, hence, this tactic of attack is more difficult than strategically-timed attack.

The first hypothesis assumed that we can take full control of the target agent, and enable to take any action in any time step. Therefore, under such condition, this problem can be simplified to planning an action sequence,

which can make agent to the target state s_g from state s_t . For the second hypothesis, Lin et al. specially designed an adversarial example $s_t + \delta_t$ to lure target agent to implement the first action in planned action sequence with method proposed by Carlini and Wagner (2017). After agent observes the adversarial examples and takes the first action designed by adversary, the environment will return a new state s_{t+1} and iterative build adversarial examples in this way. The attack flow for *enchanted attack* can be shown in Fig. 7.

For this work, strategically-time attack can achieve the same effect as the traditional method (Huang et al. 2017), while reduce the total time step for attacking. Moreover, enchanted attack can lures target agent to take planned action sequence, which suggests a new research idea for the follow-up studies. Videos are available at http://yclin.me/adversarial_attack_RL/.

Adversarial attack on VIN (AVI)

The main contribution for Liu et al. (2017) is that they proposed a method for detecting potential attack which can obstruct VIN effectiveness. They built a 2D navigation task demonstrate VIN and studied how to add obstacles to effectively affect VIN’s performance and propose a general method suitable for different kinds of environment.

Their threat model assumed that the entire environment (including obstacles, starting point and destination) is available, and they also know that the robot is trained by VIN, meanwhile, it is easy to get the VIN planning path and the theoretical path. Based on this threat model, they summarized three rules which can effectively obstructing VIN.

- **Rule 1:** The farther away from the VIN planning path, the less disturbance to the path.

Such rule can be formulated as:

$$v_{1y_k} = \omega_1 \min \{d_1 | d_1 = \sqrt{(x_r - y_{kr})^2 + (x_c - y_{kc})^2}, (x_r, x_c) = x \in X, (y_{kr}, y_{kc}) = y_k \in Y\} \tag{11}$$

where x_r, x_c is the coordinate of x , (y_{kr}, y_{kc}) is the coordinate of y_k , ω_1 is the weight of v_1 .

- **Rule 2:** It is most likely to be success when adding obstacles around the turning points on the path.

Such rule can be formulated as:

$$v_{2y_k} = \omega_2 \min \{d_2 | d_2 = \max(|t_r - y_{kr}|, |t_c - y_{kc}|), (t_r, t_c) = t \in T, (y_{kr}, y_{kc}) = y_k \in Y\} \tag{12}$$

where (t_r, t_c) denotes the coordinate of t , (y_{kr}, y_{kc}) represents the coordinate of y_k , ω_2 is the weight for v_2 . The formula considers the Chebyshev distance from y_k to the nearest turning point, and utilize the weight ω_2 to control the attenuation of v_2 .

- **Rule 3:** The closer the adding obstacle position is to the destination, the less likely it is to change the path.

The representative for (x_{nr}, x_{nc}) is the coordinate of x_n , (y_{kr}, y_{kc}) denotes the coordinate of y_k , ω_3 is the weight for v_3 . Hence, the formula can be concluded as:

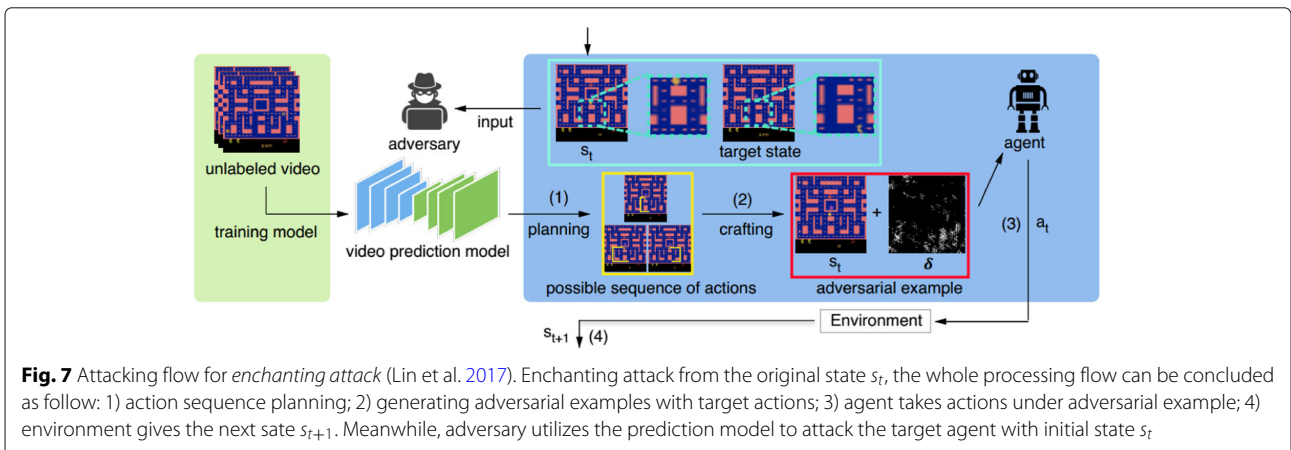
$$v_{3y_k} = \omega_3 \max(|x_{nr} - y_{kr}|, |x_{nc} - y_{kc}|), (x_{nr}, x_{nc}) = x_n, (y_{kr}, y_{kc}) = y_k \in Y \tag{13}$$

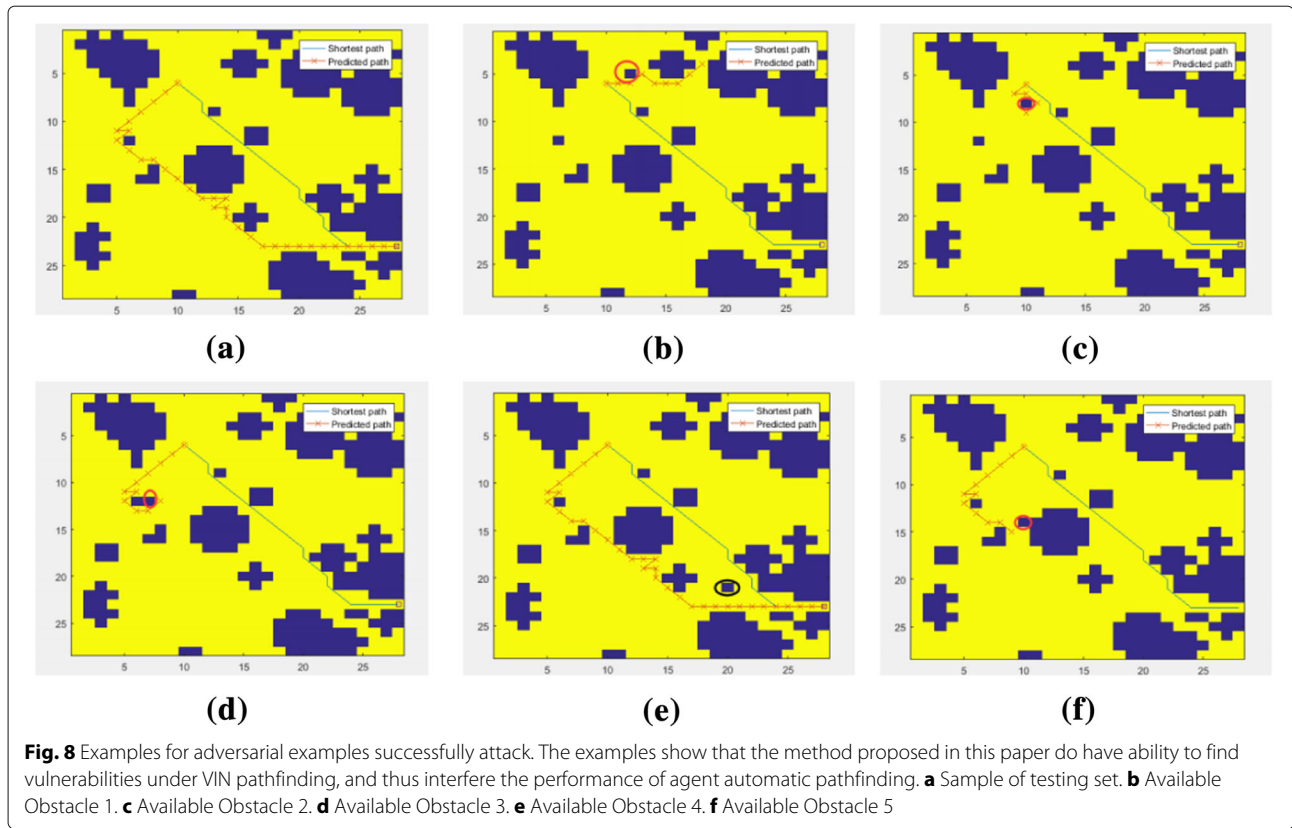
this formula considers the Chebyshev distance from y_k to the destination, and utilize the weight ω_3 to control the attenuation of v_3 .

Calculating the value v considering three rules for each available point, meanwhile, sorting the values to pick up most valuable points $S = y | v_{y_k} \in \max_i V, y \in Y, V = v_{y_1}, v_{y_2}, \dots, v_{y_k}$.

Liu’s method has great performance on automatically finding vulnerable points of VIN and thus obstructing navigation task, which can be shown in Fig. 8.

However, this work has not give an analysis of the successful adversarial attack from the algorithm level, but summarized the generation rules from the successful black-box adversarial examples. Meanwhile, similar to the





work of Xiang et al. and Bai et al., the map size has too many limitations. Only the size under 28×28 have been experimentally verified, and such size is not enough to prove the accuracy of the method proposed in this paper.

Summary for adversarial attack in reinforcement learning

We give summary on the attributions of adversarial attacking methods described above, which can be shown in Table 3.

FGSM (Goodfellow et al. 2014a), SPA (Xiang et al. 2018), WBA (Bai et al. 2018), and CDG (Chen et al. 2018b) belong to White-box attack, which have access to the details related to training algorithm and corresponding parameters of the target model. Meanwhile, the PIA (Behzadan and Munir 2017), STA (Lin et al. 2017), EA (Lin et al. 2017), and AVI (Liu et al. 2017) are Black-box attacks, in which adversary has no idea of the details related to training algorithm and corresponding parameters of the model, for the threat model discussed in these literatures, authors assumed that the adversary has access to the training environment but has no idea of the random initializations of the target policy, and additionally does not know what the learning algorithm is.

For White-box attack policies, we summarize the parameters utilized for such methods. SPA, WBA, CDG, PIA, and AVI all have the specific target algorithm,

however, the target for FGSM, STA, and EA is not single reinforcement learning algorithm, in this sense, such adversarial attack methods are more universal adaptability.

Moreover, the learning way for these adversarial attack methods are different, as FGSM, SPA, WBA, CDG, and AVI are all “One-shot” learning, and PIA, STA, and EA are “Iterative” learning. Additionally, for all attack methods introduced here can generate adversarial examples to achieve the purpose of attacking successfully under a relatively high confidence. The application scenario for FGSM, PIA, STA, and EA are Atari game, meanwhile, the scenario for SPA, WBA, CDG, and AVI are all path planning. We also take a statistical analysis of the attack results for the algorithms discussed above.

Defense technology against adversarial attack

Since the adversarial examples attack proposed by Szegedy et al. (2013) in 2013, meanwhile, there are many related researchers have investigated the approaches to defense against adversarial examples. In this section, we briefly discussed some representative attempts that have been done to resist adversarial examples. Mainly divided into three parts, which are modifying input, modifying the objective function, and modifying the network structure.

Table 3 Summary for the attributes of diverse attacking method

Method	Black/White box	Parameters	Target algorithm	Learning	Examples generated	Application scenario	Attack effect
FGSM (Goodfellow et al. 2014a)	White Box	θ, x, y	None	One shot	True	Atari Game	Taking wrong action
SPA (Xiang et al. 2018)	White Box	V_0, P_{start}	Q-Learning	One shot	True	Path Planning	No normal path planning
WBA (Bai et al. 2018)	White Box	V_0, x	DQN	One shot	True	Path Planning	No normal path planning
CDG (Chen et al. 2018b)	White Box	V, x	A3C	One shot	True	Path Planning	Unable to reach destination/Time increased
PIA (Behzadan and Munir 2017)	Black Box	None	DQN	Iterative	True	Atari Game	Taking wrong action
STA (Lin et al. 2017)	Black Box	None	None	Iterative	True	Atari Game	Taking wrong action
EA (Lin et al. 2017)	Black Box	None	None	Iterative	True	Atari Game	Taking wrong action
AVI (Liu et al. 2017)	Black Box	None	VIN	One shot	True	Path Planning	No normal path planning

The *Parameters* denotes the parameters utilized for calculating the perturbations, *Learning* denotes the learning style for each attack method

Modifying input

Adversarial training and its variants

- ***Adversarial training***

Adversarial training is the one of the most common strategies in the related literature to improve the robustness of neural networks. By continuously inputting new types of adversarial examples and conducting adversarial training, the network's robustness is continuously improved. In 2015, Goodfellow et al. (2014b) developed a method of generating adversarial examples (FGSM, see (Goodfellow et al. 2014a)), and they also proposed to conduct adversarial training to resist adversarial perturbation exploiting the adversarial examples generated by the attack method, the adversarial examples are constantly updated during the training process so that the classification model can resist the adversarial examples. However, Moosavi-Dezfooli et al. (2017) pointed out that no matter how many adversarial examples are added, there are new adversarial examples that can cheat the trained networks in 2017. After that, by combining adversarial examples with other methods, researchers have produced better approaches defending adversarial examples in some recent works.

- ***Ensemble adversarial training***

It trains networks by utilizing the several pre-trained vanilla networks to generate one-step adversarial examples. The model by adversarial training can defend weak perturbations attack but can't defend against strong ones. Based on this, Florian Tramer et al. (2017) introduced the ensemble adversarial training, which enhances training data with perturbations transferred from other static pre-trained models, this approach separates the generation of adversarial examples from the model being trained, simultaneously drawing an explicit connection with robustness to black-box adversaries. This model trained by ensemble adversarial training has strong robustness to black-box attacks on ImageNet.

- ***Cascade adversarial training***

For unknown iterative attacks, Na et al. (2018) proposed cascade adversarial training, they trained the network by inputting adversarial images generated from the iterative defended network and one-step adversarial images from the network being trained. At the same time, the authors regularized the training with a unified embedding so that the convolution filters can gradually learn how to ignore pixel-level perturbations. The cascade adversarial training is shown in Fig. 9.

- ***Principled adversarial training***

From the perspective of distributed robust optimization, Aman Sinha et al. (2018) provided a principled

adversarial training, which guaranteed the performance of neural networks under adversarial data perturbation. By utilizing the Lagrange penalty form of perturbation under the potential data distribution in the Wasserstein ball, the authors provide a training process that uses worst-case perturbations of training data to reinforce model parameter updates.

- ***Gradient Band-based Adversarial Training***

Chen et al. (2018b) proposed a generalized attack immune model based on gradient band, which can be shown in Fig. 10, mainly consists of *Generation Module*, *Validation Module*, and *Adversarial Training Module*.

For the original clean map, *Generation Module* can generate dominant adversarial examples based on the *Common Dominant Adversarial Examples Generation Method (CDG)* (see Section 3.2.4). *Validation Module* can utilize the well trained A3C agent against the original clean map, to calculate the F_{attack} for each example based on the success criteria for attack proposed in this paper. *Adversarial Training Module* utilize a single example which can attack successfully for adversarial training, and obtain a newly well trained A3C $agent_{new}$ which can finally realize "1:N" attack immunity.

Data randomization

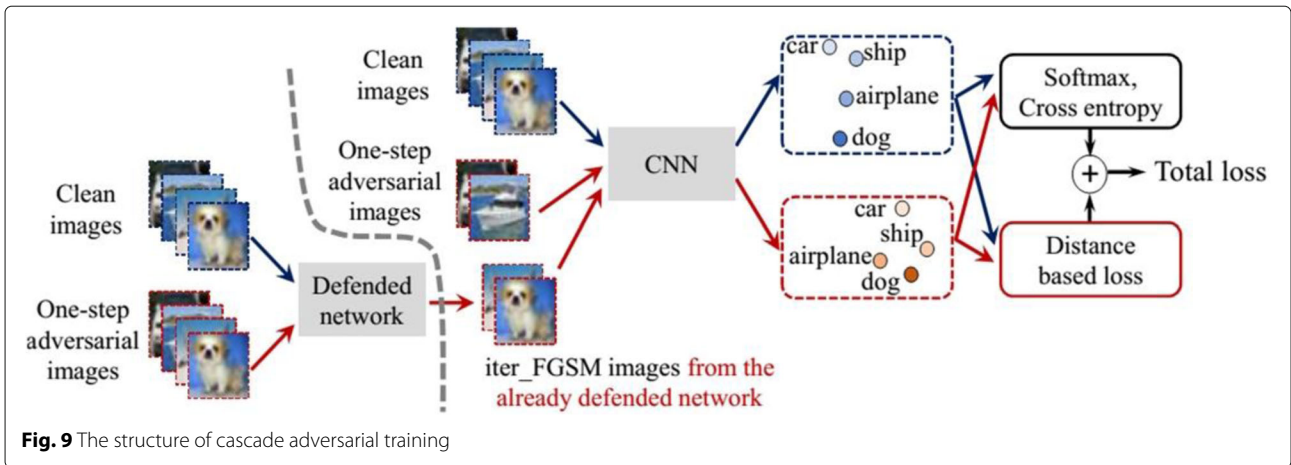
In 2017, Xie et al. (2017) found that introducing random resizing to the training images can reduce the strength of the attack. After that, they further proposed (Xie et al. 2018) to use randomization at inference time to mitigate the effects of adversarial attack. They add a random resize layer and a random padding layer before the network of classification, their experiments demonstrate that the proposed randomization method is very effective at resisting one-step and iterative attacks.

Input transformations

Guo et al. (2018) proposed strategies to defend against adversarial examples through transforming the inputs before feeding them to the image-classification system. The input transformations include bit-depth reduction, JPEG compression, total variance minimization, and image quilting before feeding the image. And the authors showed that total variance minimization and image quilting are very effective defenses on ImageNet.

Input gradient regularization

Ross and Doshi-Velez (2017) first exploited input gradient regularization (Drucker and Le Cun 1992) to improve the adversarial robustness. In this defense technology trains differentiable models that penalizes the degree to which small changes in inputs can alter model predictions. And the work shown that training with gradient regularization strengthened the robustness to adversarial perturbations, and it has a greater robustness combined the gradient



regularization with adversarial training, but the computational complexity is too high.

Modifying the objective function

Adding stability term

Zheng et al. (2016) conducted stability training through adding stability term to the objective function to encourage DNN to generate similar output for images of various perturbed versions. The perturbed copy I' of the input image I is generated by a Gaussian noise ϵ , the final loss L is consisted of the task objective L_o and the stability loss $L_{stability}$.

Adding regularization term

Yan et al. (2018) append the regularization term based on adversarial perturbations to the objective function, they proposed a training recipe called “deep defense”.

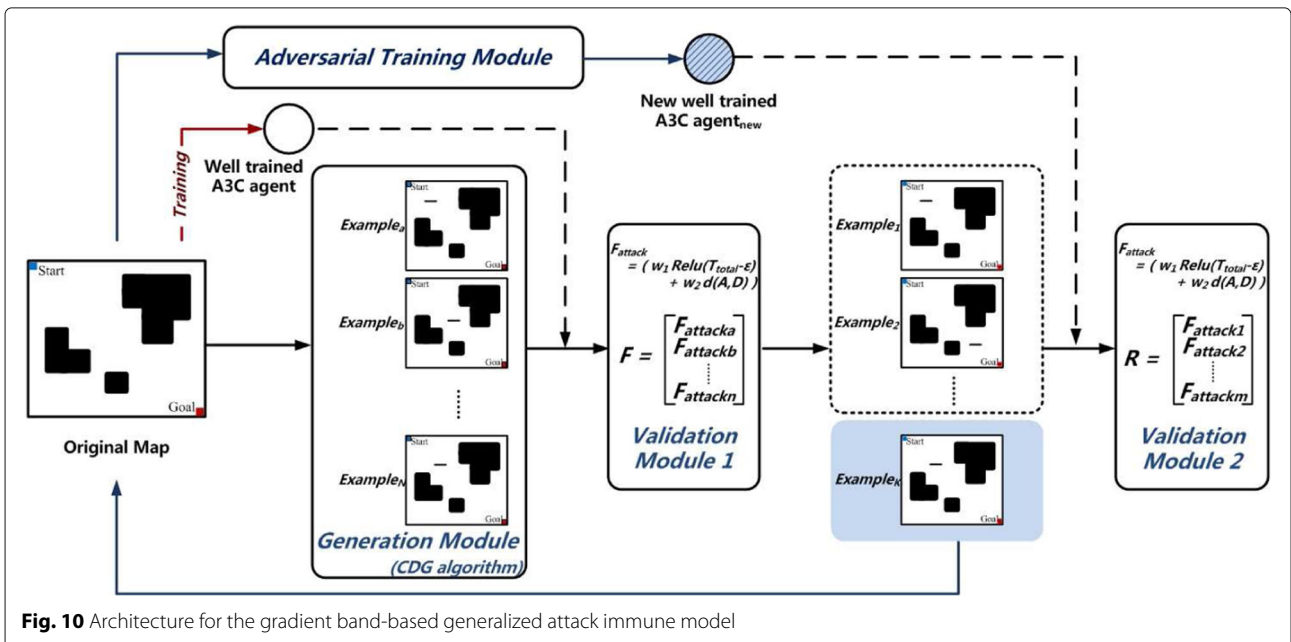
Specifically, the authors optimize the objective function jointed the original objective function term and a scaled $\|\Delta_x\|_p$ as a regularization term. Given a training set $\{(x_k, y_k)\}$ and the parameterized function f , and W collects learnable parameters of f , the new objective function can be optimized as below:

$$\min_W \sum_k L(y_k, f(x_k; W)) + \lambda \sum_k R\left(-\frac{\|\Delta_{xk}\|_p}{\|x_k\|_p}\right) \quad (14)$$

By combining an adversarial perturbation-based regularization with the classification objective function, the training model can learn to defend against adversarial attacks directly and accurately.

Dynamic quantized activation function

Rakin et al. (2018) first explored to use quantization of activation functions and proposed to exploit adaptive



quantization techniques for the activation functions so that training the network to defend against adversarial examples. They show the proposed Dynamic Quantized Activation (DQA) method greatly heightened the robustness of DNN under white-box attack, such as FGSM (Goodfellow et al. 2014a), PGD (Madry et al. 2017), and C&W (Carlini and Wagner 2017) attacks on MNIST and CIFAR-10 datasets. In this approach, the authors integrate the quantized activation functions into an adversarial training method, in which training model to learn parameters γ to minimize the risk $R_{(x,y) \sim L[U(\gamma, x, y)]}$, γ consists of parameters in DNN. Based on this, given the input image x and the adversary example $x + \epsilon$, this work aim to minimize the objective function to enhance the robustness

$$\min_{R_{(x,y) \sim L[\max J([\gamma, T], x + \epsilon, y)]}} \quad (15)$$

where adding a new set of learnable parameters $T := [t_1, t_2, \dots, t_{m-1}]$. For n -bit quantized activation function, the quantization will have $2^n - 1$ threshold values T , let $m = 2^n - 1$, sgn represents the sign function, then m level quantization function is as follows:

$$f(x) = 0.5 \times \left[\begin{aligned} &\text{sgn}(x - t_{m-1}) + \sum_{i=m-1}^{m/2+1} t_i (\text{sgn}(t_i - x)) \\ &+ \text{sgn}(x - t_{i-1}) + \sum_{i=m/2}^2 t_{i-1} (\text{sgn}(t_i - x)) \\ &+ \text{sgn}(x - t_{i-1}) - \text{sgn}(t_1 - x) \end{aligned} \right] \quad (16)$$

Stochastic activation pruning

Inspired by game theory, S. Dhillon et al. (2018) proposed a mixed strategy Stochastic Activation Pruning (SAP) for adversarial defense. The method prunes a random activation subset (preferentially pruning those with smaller magnitude) and expands survivors to compensate, using SAP to pretrained networks without any additional training provides robustness against adversarial examples. And the authors showed that combining SAP with adversarial examples has a greater benefits. In particularly, their experiments demonstrate that SAP can effectively defend against adversarial examples in reinforcement learning.

Modifying the network structure

Defensive distillation

Papernot et al. (2016a) proposed the defensive distillation mechanism for training network to resist adversarial attacks. Defensive distillation, a strategy that trains models to output the probability of different classes rather than the difficult decision of which class to output, the probability is provided by an early model that uses the labels of hard classification to train on the same task. Papernot

et al. showed that defensive distillation can be used to resist small-disturbed adversarial attacks through training network to defend L-BFGS (Szegedy et al. 2013) attack and FGSM (Goodfellow et al. 2014a) attack. Unfortunately, defensive distillation is only applicable to DNN models based on energy probability distributions. Nicholas Carlini and David Wagner proved that defensive distillation is ineffective in (Carlini and Wagner 2016), and they introduced a method of constructing adversarial examples (Carlini and Wagner 2017), this method is not affected by various anti-attack methods, including defensive distillation.

High-level representation guided denoiser

Liao et al. (2018) proposed high-level representation guided denoiser (HGD) to defend adversarial examples for image classification. The main idea is to train a denoiser based on neural network for removing the adversarial perturbation before sending them to the target model. FLiao et al. use denoising U-net (Ronneberger et al. 2015) (DUNET) as a denoising model. Compared to denoising autoencoder (DAE) (Vincent et al. 2008), DUNET is directly connected between encoder layers and decoder layers of the same resolution, so the network only needs to learn how to remove noise, instead of learning how to reconstruct the whole image. And without using a pixel-level reconstruction loss function, the authors use the difference between top-level outputs of the target model induced by original and adversarial examples as the loss function to guide the training of an image denoiser. The proposed HGD has a good generalization and the target model is more robust against both white-box and black-box attacks.

Add detector subnetwork

Metzen et al. (2017) proposed to add a detector subnetwork for augmenting deep neural networks, the subnetwork is trained on a binary classification task that distinguishes real data from data containing adversarial perturbations. Considering that detector is also adversarial, they proposed dynamic adversary training, which introduces a novel adversary that aims at fooling both the classifier and the detector, and trains the detector to counteracting this novel adversary. The experiment results show that dynamic detector has the robustness and its detectability is more than 70% on the CIFAR10 dataset (Krizhevsky and Hinton 2009).

Multi-model-based defense

Srisakaokul et al. (2018) explored a novel defense approach, MULDEF, based on the principle of diversity. The MULDEF approach firstly constructs a family of models by combining the seed model (the target model) with additional models (constructed from the seed model), the constructed family of models are complementary to

each other to obtain robustness diversity, specifically, the adversarial examples of a model usually doesn't be the adversarial examples of other models in model family. Then the method randomly selects one model in these models to be applied on a given input example. The randomness of selection reduces can reduce the success rate of the attack. The evaluation results demonstrate that MULDEF augmented the adversarial accuracy of the target model by about 35-50% and 2-10% in the white-box and black-box attack scenarios, respectively.

Generative models

- *PixelDefend*

Song et al. (2017) proposed a method named *PixelDefend* which can utilized generative models to defend against adversarial examples. In this paper, authors showed that the adversarial examples mainly lie on low probability regions of training distribution, regardless of the attack type and target model. Moreover, they found that neural density model outperform on detecting the human invisible adversarial perturbations, and based on this discovery, Song et al. proposed a new approach named *PixelDefend* which can purifies a perturbed image return to the distribution of training data. Meanwhile, they announced that *PixelDefend* can be utilized as a novel family of methods which can combined with other model-specific defenses. Experimental results (e.g., Fig. 11) showed that *PixelDefend* can greatly improves the recovery capability of varieties state-of-art defense methods.

- *Defense-GAN*

Samangouei et al. (2018) gave the first attempt to construct a defense model against adversarial attack based on GAN (Radford et al. 2015). They proposed a new defense policy named *Defense-GAN* which takes use of generation

model to improve the robustness against *Black/White-Box Attack*. Moreover, any classification model can utilize the *Defense-GAN* proposed in this paper, and will not change the structure of classifier or the process for training. *Defense-GAN* can be used as a defense technology that can against any adversarial attack as such method does not assume knowledge of the process for generating the adversarial examples. The experimental results showed that *Defense-GAN* proposed in this paper is effective when against different adversarial attacks, and can improve the performance on existing defense technologies.

Discriminative model

Since it is not guaranteed that the generated adversarial examples will obstruct the VIN path planning successfully generated in Liu et al. (2017), Wang et al. explored a fast approach to automatically identify VIN adversarial examples. In order to estimate whether an attack is successful, they compared the difference between the two paths on a pair of maps, the normal map and the adversarial map. By visualizing the pair of paths on a path image, they transformed the different attack results into different categories of path images. In this way, they analyzed the possible scenarios of the adversarial maps and define the categories of the predicted path pairs. They divided the results into four limited categories, which are the unreached path (UrP) class, the fork path (FP) class, the detour path (DP) class and the unchanged path (UcP) class. Based on the categories definition, they implemented a training-based identification method by combining the path feature comparison and path images classification.

In this method, the UrP and UcP can be identified through path feature comparison and the DP and FP can be identified through path image classification. The experimental results showed that this method can achieve

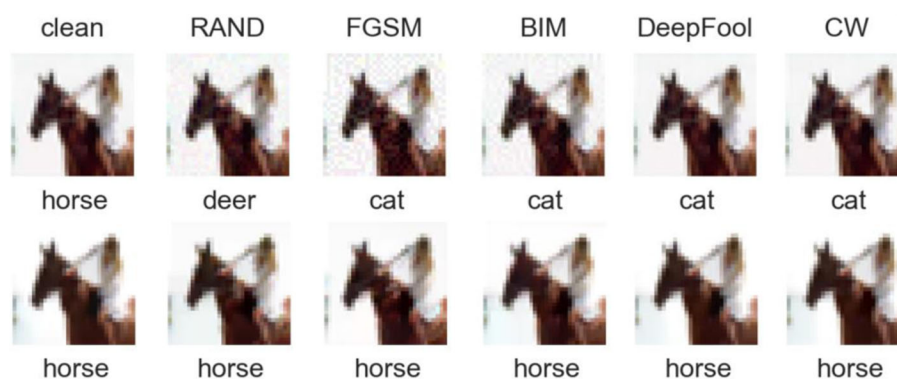


Fig. 11 The example for *PixelDefend* (Song et al. 2017). The first image denote the original clean image in CIFAR-10 (Krizhevsky et al. 2014), and the remaining pictures represent the adversarial examples based on varieties attack methods which have been shown above each example, and the predicted label has been shown on the bottom. Meanwhile, the second line denotes the corresponding purified images

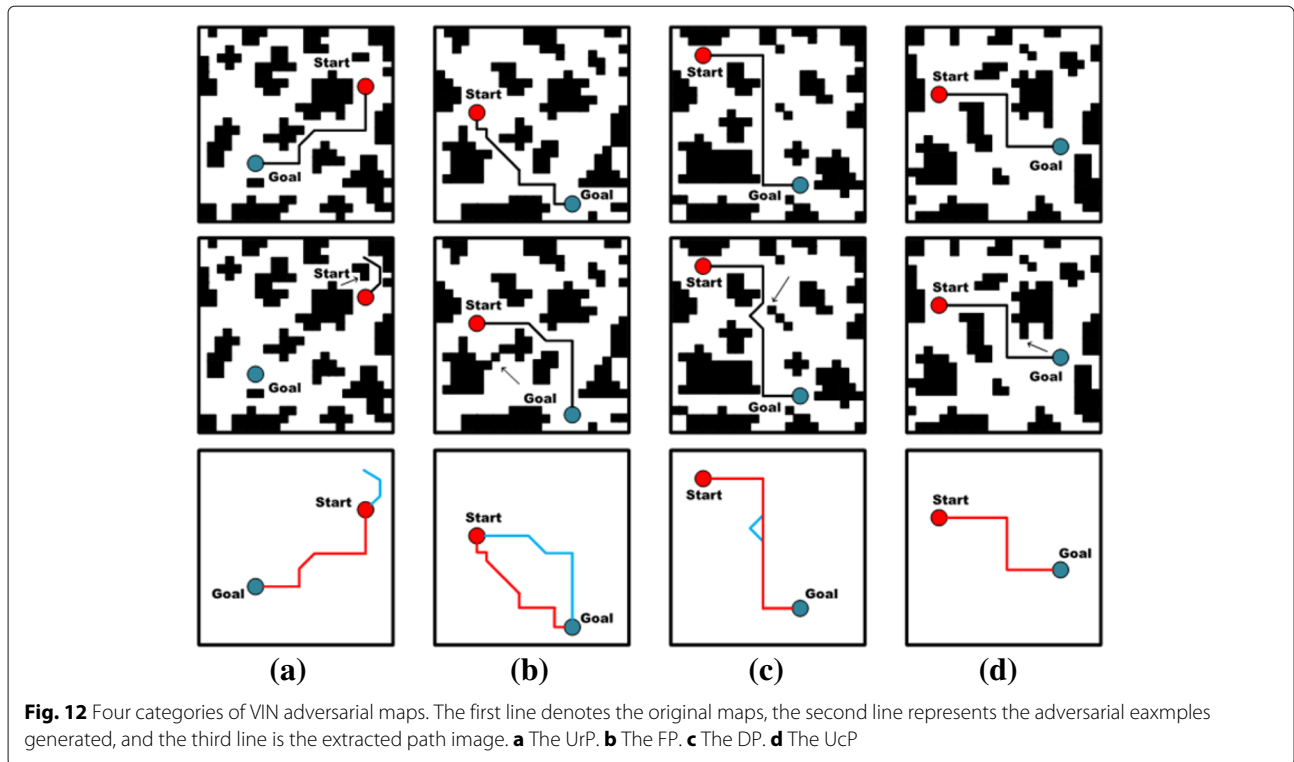


Table 4 Different attacks targeted by different defense technologies

	FGSM	IFGSM	PGM	CDG	DeepFool	C&W	JSMA	ITGSM
Modifying input								
Adversarial training	✓							
Ensemble adversarial training	✓							
Cascade adversarial training		✓						
Principled adversarial training			✓					
Gradient band-based adversarial training				✓				
Data randomization					✓	✓		
Input transformations					✓			
Input gradient regularization							✓	✓
Modifying the objective function	FGSM	DeepFool	C&W	Small perturbations	PGD			
Adding stability term				✓				
Adding regularization term		✓						
Dynamic quantized activation function	✓		✓		✓			
Stochastic activation pruning	✓							
Modifying the network structure	FGSM	IFGSM	DeepFool	C&W	JSMA	ITGSM	BIM	Opt
Defensive distillation	✓					✓		
High-level representation Guided Denoiser	✓	✓						
Add detector subnetwork	✓	✓	✓					
Multi-model-based defense	✓			✓				
Generative models	✓			✓				
Characterizing adversarial subspaces	✓			✓	✓		✓	✓

a high-accuracy and faster identification than manual observation method (e.g., Fig. 12).

Characterizing adversarial subspaces

Ma et al. (2018) gave the first attempt to explain the extent adversarial perturbation can effect the *Local Intrinsic Dimensionality (LID)* (Houle 2017) characteristic of adversarial regions. Moreover, they showed empirically that LID characteristics can facilitate the distinction of adversarial examples generated by several state-of-art attacks. Meanwhile, they proved that LID can be utilized to differentiate adversarial examples, and the experimental results show that among the five attack strategies (FGSM (Goodfellow et al. 2014a), BIM-a (Saad 2003), BIM-b (Saad 2003), JSMA (Papernot et al. 2016b), Opt) based on three benchmark data sets (MNIST (LeCun et al. 2010), CIFAR-10 (Krizhevsky et al. 2014), SVHN (Netzer et al. 2011)) considered for this paper, the method based on LID can outperform against most state-of-art methods.

Ma et al. announced that their analysis of LID characteristic for adversarial region, not only can motivates new direction for effective adversarial defense, but also provides more challenges for the development of new adversarial attacks, meanwhile, enable us to better understand the vulnerabilities of DNNs (LeCun et al. 1989).

Conclusion and discussion

In this paper, we give the very first attempt to conduct a comprehensive survey on adversarial attacks in the context of reinforcement learning under AI security. Reinforcement learning is a workhorse for AI applications ranging from *Atari Game* to *Connected and Automated Vehicle System (CAV)*, hence, how to build a reliable reinforcement learning system to support the security critical applications in AI, has become a concern which is more critical than ever. However, Huang et al. (2017) discovered that the interesting attack mode *adversarial attack* also be effective when targeting neural networks under reinforcement learning, which has inspired innovative researches in this direction. Therefore, our work reviews such contributions, and mainly focus on the most influential and interesting works in this field. We give a comprehensive introduction to the literatures on adversarial attack under various fields of reinforcement learning applications, and briefly analyze the most valuable defense technologies against existing adversarial attacks (Table 4).

Although, the RL system does exist the security vulnerability of “Adversarial attack”, by the survey on existing adversarial attack technologies, it is found that the exist of complete Black-box attacks are rare (complete Black-box attack means that the adversary has no idea of the target model, and can not interact with the target agent at all), which makes it very difficult for adversaries to attack the reinforcement learning system in practice. Moreover,

owing to the very high activity in this research direction, it can be expected that, in the future an largely reliable reinforcement learning system will be available to support critical security applications in AI.

Acknowledgements

The authors would like to thank the guidance of Professor Wenjia Niu and Professor Jiqiang Liu. Meanwhile this research is supported by the National Natural Science Foundation of China (No. 61672092), Science and Technology on Information Assurance Laboratory (No. 614200103011711), the Project (No. BMK2017B02-2), Beijing Excellent Talent Training Project, the Fundamental Research Funds for the Central Universities (No. 2017RC016), the Foundation of China Scholarship Council, the Fundamental Research Funds for the Central Universities of China under Grants 2018JBZ103.

Funding

This research is supported by the National Natural Science Foundation of China (No. 61672092), Science and Technology on Information Assurance Laboratory (No. 614200103011711), the Project (No. BMK2017B02-2), Beijing Excellent Talent Training Project, the Fundamental Research Funds for the Central Universities (No. 2017RC016), the Foundation of China Scholarship Council, the Fundamental Research Funds for the Central Universities of China under Grants 2018JBZ103.

Availability of data and materials

Not applicable.

Authors' contributions

TC conceived and designed the study. TC and YX wrote the paper. JL, WN, ET, and ZH reviewed and edited the manuscript. All authors read and approved the manuscript.

Authors' information

Wenjia Niu obtained his Bachelor degree from Beijing Jiaotong University in 2005, PhD degree from Chinese Academy of Science in 2010, all in Computer Science. Now he is currently a professor in Beijing Jiaotong University. His research interests are AI Security, Agent and Data Mining. He has published more than 50 research papers, including a number of regular papers in the famous international journals and Conferences, such as KAIS (Elsevier), ICDM, CIKM and ICSOC. He has published 2 edited books. He serves on the Steering Committee of ATIS (2013-2016) and the PC Chair of ASONAM C3/2015. He has been PhD Thesis Examiner of Deakin University, Title Page Click here to access/download;Title Page;cover letter.pdf the guest editor for the Chinese Journal of Computer, Enterprise Information System, Concurrency and Computing: Practise and Experience, and Future Generation Computer Systems, etc.. He is also members both of the IEEE and ACM.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 31 October 2018 Accepted: 12 February 2019

Published online: 29 March 2019

References

- Akhtar N, Mian A (2018) Threat of adversarial attacks on deep learning in computer vision: A survey. arXiv preprint arXiv:1801.00553
- 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
- Behzadan V, Munir A (2017) Vulnerability of deep reinforcement learning to policy induction attacks. In: International Conference on Machine Learning and Data Mining in Pattern Recognition. Springer, Cham, pp 262–275
- Bougiouklis A, Korkofigkas A, Stamou G (2018) Improving Fuel Economy with LSTM Networks and Reinforcement Learning. In: International Conference on Artificial Neural Networks. Springer, Cham, pp 230–239

- Carlini N, Wagner D (2016) Defensive distillation is not robust to adversarial examples. arXiv preprint arXiv:1607.04311
- 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
- Chen QA, Yin Y, Feng Y, Mao ZM, Liu HX (2018a) Exposing Congestion Attack on Emerging Connected Vehicle based Traffic Signal Control. In: Network and Distributed Systems Security (NDSS) Symposium
- Chen T, Niu W, Xiang Y, Bai X, Liu J, Han Z, Li G (2018b) Gradient band-based adversarial training for generalized attack immunity of a3c path finding. arXiv preprint arXiv:1807.06752
- Dhillon GS, Azizzadenesheli K, Bernstein JD, Kossaiji J, Khanna A, Lipton ZC, Anandkumar A (2018) Stochastic activation pruning for robust adversarial defense. In: International Conference on Learning Representations. <https://openreview.net/forum?id=H1uR4GZRZ>
- Drucker H, Le Cun Y (1992) Improving generalization performance using double backpropagation. *IEEE Trans Neural Netw* 3(6):991–997
- Farahmand AM (2011) Action-gap phenomenon in reinforcement learning. In: Advances in Neural Information Processing Systems. pp 172–180
- Goodall C, El-Sheimy N (2017) System and method for intelligent tuning of Kalman filters for INS/GPS navigation applications: U.S. Patent No. 9,593,952. Washington, DC: U.S. Patent and Trademark Office
- Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572
- Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. *CoRR abs/1412.6572*. [1412.6572](https://arxiv.org/abs/1412.6572)
- Guo C, Rana M, Cisse M, van der Maaten L (2018) Countering adversarial images using input transformations. In: International Conference on Learning Representations. <https://openreview.net/forum?id=Syl7CIWCb>
- Guo X, Singh S, Lee H, Lewis RL, Wang X (2014) Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. In: Advances in neural information processing systems. pp 3338–3346
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
- Houle ME (2017) Local intrinsic dimensionality I: an extreme-value-theoretic foundation for similarity applications. In: International Conference on Similarity Search and Applications. Springer, Cham. pp 64–79
- Huang S, Papernot N, Goodfellow I, Duan Y, Abbeel P (2017) Adversarial attacks on neural network policies. arXiv preprint arXiv:1702.02284
- Jaderberg M, Mnih V, Czarniecki WM, Schaul T, Leibo JZ, Silver D, Kavukcuoglu K (2016) Reinforcement learning with unsupervised auxiliary tasks. arXiv preprint arXiv:1611.05397
- Jia YJ, Zhao D, Chen QA, Mao ZM (2017) Towards secure and safe applied automated vehicles. In: 2017 IEEE Intelligent Vehicles Symposium (IV). IEEE. pp 705–711
- Krizhevsky A, Hinton G. (2009) Learning multiple layers of features from tiny images. Technical report, University of Toronto 1(4):7
- Krizhevsky A, Nair V, Hinton G (2014) The cifar-10 dataset. online: <http://www.cs.toronto.edu/kriz/cifar.html>
- Kurakin A, Goodfellow I, Bengio S (2016) Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236
- LeCun Y, Cortes C, Burges C (2010) Mnist handwritten digit database 2. AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>
- LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551
- Liang Y, Machado MC, Talvitie E, Bowling M (2016) State of the art control of atari games using shallow reinforcement learning. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems. pp 485–493
- Liao F, Liang M, Dong Y, Pang T, Hu X, Zhu J (2018) Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 1778–1787
- Lin Y-C, Hong Z-W, Liao Y-H, Shih M-L, Liu M-Y, Sun M (2017) Tactics of adversarial attack on deep reinforcement learning agents. arXiv preprint arXiv:1703.06748
- Liu J, Niu W, Liu J, Zhao J, Chen T, Yang Y, Xiang Y, Han L (2017) A Method to Effectively Detect Vulnerabilities on Path Planning of VIN. In: International Conference on Information and Communications Security. Springer, Cham. pp 374–384
- Ma X, Li B, Wang Y, Erfani SM, Wijewickrema S, Houle ME, Schoenebeck G, Song D, Bailey J (2018) Characterizing adversarial subspaces using local intrinsic dimensionality. arXiv preprint arXiv:1801.02613
- Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083
- Markov A (1907) Investigation of a remarkable case of dependent trials. *Izv Ros Akad Nauk* 1
- Martínez-Tenor Á, Cruz-Martín A, Fernández-Madrugal JA (2018) Teaching machine learning in robotics interactively: the case of reinforcement learning with Lego Mindstorms. *Interact Learn Environ*:1–14
- Metzen JH, Genewein T, Fischer V, Bischoff B (2017) On detecting adversarial perturbations. *CoRR abs/1702.04267*. [1702.04267](https://arxiv.org/abs/1702.04267)
- Miyato T, Maeda SI, Ishii S, Koyama M (2018) Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Trans Pattern Anal Mach Intell* PP(99):1
- Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602
- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, et al (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529
- Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp 1928–1937
- Moosavi-Dezfooli S-M, Fawzi O, Frossard P (2017) Universal adversarial perturbations. arXiv preprint
- Na T, Ko JH, Mukhopadhyay S (2018) Cascade adversarial machine learning regularized with a unified embedding. arXiv preprint arXiv:1708.02582
- Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning
- Ohn-Bar E, Trivedi MM (2016) Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Trans Intell Veh* 1(1):90–104
- Papernot N, McDaniel P, Wu X, Jha S, Swami A (2016a) Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP). IEEE. pp 582–597
- Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A (2016b) The limitations of deep learning in adversarial settings. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE. pp 372–387
- Papernot N, McDaniel P, Goodfellow I, Jha S, Celik ZB, Swami A (2016c) Practical black-box attacks against deep learning systems using adversarial examples. arXiv preprint arXiv:1602.02697 1(2):3
- Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434
- Rakin AS, Yi J, Gong B, Fan D (2018) Defend deep neural networks against adversarial examples via fixed and dynamic quantized activation functions. arXiv preprint arXiv:1807.06714
- Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, Cham. pp 234–241
- Ross AS, Doshi-Velez F (2017) Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. arXiv preprint arXiv:1711.09404
- Saad Y (2003) Iterative methods for sparse linear systems, vol. 82. *siam*
- Samangouei P, Kabkab M, Chellappa R (2018) Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605
- Schulman J, Levine S, Abbeel P, Jordan MI, Moritz P (2015) Trust Region Policy Optimization. In: *icml* Vol. 37. pp 1889–1897
- Shalev-Shwartz S, Shammah S, Shashua A (2016) Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295
- Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484
- Sinha A, Namkoong H, Duchi J (2018) Certifiable distributional robustness with principled adversarial training. In: International Conference on Learning Representations. <https://openreview.net/forum?id=Hk6kPgZA->

- Song Y, Kim T, Nowozin S, Ermon S, Kushman N (2017) Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766
- Srisakaokul S, Zhong Z, Zhang Y, Yang W, Xie T (2018) Muldef: Multi-model-based defense against adversarial examples for neural networks. arXiv preprint arXiv:1809.00065
- Swiderski F, Snyder W (2004) Threat modeling. Microsoft Press
- Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199
- Tamar A, Wu Y, Thomas G, Levine S, Abbeel P (2016) Value iteration networks. In: Advances in Neural Information Processing Systems. pp 2154–2162
- Touretzky DS, Mozer MC, Hasselmo ME (eds) (1996) Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference, vol. 8. MIT Press
- Tramèr F, Kurakin A, Papernot N, Boneh D, McDaniel PD (2017) Ensemble adversarial training: Attacks and defenses. CoRR abs/1705.07204. [1705.07204](https://arxiv.org/abs/1705.07204)
- Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. ACM. pp 1096–1103
- Watkins C, Dayan P (1992) Machine learning. Technical Note: Q-Learning 8:279–292
- Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemometrics and intelligent laboratory systems 2(1-3):37–52
- Xiang Y, Niu W, Liu J, Chen T, Han Z (2018) A PCA-Based Model to Predict Adversarial Examples on Q-Learning of Path Finding. In: 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE. pp 773–780
- Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille AL (2017) Adversarial examples for semantic segmentation and object detection. CoRR abs/1703.08603. [1703.08603](https://arxiv.org/abs/1703.08603)
- Xie C, Wang J, Zhang Z, Ren Z, Yuille A (2018) Mitigating adversarial effects through randomization. In: International Conference on Learning Representations. <https://openreview.net/forum?id=Sk9yuql0Z>
- Xiong W, Droppo J, Huang X, Seide F, Seltzer M, Stolcke A, Yu D, Zweig G (2016) Achieving human parity in conversational speech recognition. arXiv preprint arXiv:1610.05256
- Yan Z, Guo Y, Zhang C (2018) Deepdefense: Training deep neural networks with improved robustness. CoRR abs/1803.00404. [1803.00404](https://arxiv.org/abs/1803.00404)
- Yang T, Xiao Y, Zhang Z, Liang Y, Li G, Zhang M, Li S, Wong T-W, Wang Y, Li T, et al (2018) A soft artificial muscle driven robot with reinforcement learning. Sci Rep 8(1):14518
- Zhang J, Lu C, Fang C, Ling X, Zhang Y (2018) Load Shedding Scheme with Deep Reinforcement Learning to Improve Short-term Voltage Stability. In: 2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). IEEE. pp 13–18
- Zheng S, Song Y, Leung T, Goodfellow I (2016) Improving the robustness of deep neural networks via stability training. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4480–4488
- Zhu Y, Mottaghi R, Kolve E, Lim JJ, Gupta A, Fei-Fei L, Farhadi A (2017) Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE international conference on robotics and automation (ICRA). IEEE. pp 3357–3364

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
