Natural Black-Box Adversarial Examples against Deep Reinforcement Learning

Mengran Yu, Shiliang Sun *

School of Computer Science and Technology, East China Normal University,
3663 North Zhongshan Road, Shanghai 200062, P.R. China
mengranyu97@gmail.com, slsun@cs.ecnu.edu.cn

Abstract

Black-box attacks in deep reinforcement learning usually retrain substitute policies to mimic behaviors of target policies as well as craft adversarial examples, and attack the target policies with these transferable adversarial examples. However, the transferability of adversarial examples is not always guaranteed. Moreover, current methods of crafting adversarial examples only utilize simple pixel space metrics which neglect semantics in the whole images, and thus generate unnatural adversarial examples. To address these problems, we propose an advRL-GAN framework to directly generate semantically natural adversarial examples in the black-box setting, bypassing the transferability requirement of adversarial examples. It formalizes the black-box attack as a reinforcement learning (RL) agent, which explores natural and aggressive adversarial examples with generative adversarial networks and the feedback of target agents. To the best of our knowledge, it is the first RL-based adversarial attack on a deep RL agent. Experimental results on multiple environments demonstrate the effectiveness of advRL-GAN in terms of reward reductions and magnitudes of perturbations, and validate the sparse and targeted property of adversarial perturbations through visualization.

Introduction

Deep reinforcement learning (DRL) has achieved significant performance in widespread applications, e.g., video game playing (Mnih et al. 2015), robotics (Levine et al. 2016), and autonomous driving (Isele et al. 2018). Deep neural networks (DNNs) are usually adopted to approximate the value function and the policy in DRL, however, they are well known to be susceptive to adversarial examples with imperceptible perturbations (Szegedy et al. 2014; Kurakin, Goodfellow, and Bengio 2017; Carlini and Wagner 2017). This motivates researchers to investigate the vulnerability of DRL, especially in safety-critical tasks.

Although there have been several attempts in attacking DRL to evaluate the fragility, most approaches to construct adversarial examples against agents are conducted in the white-box setting. As the first attempt, the conventional gradient-based fast gradient sign method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) was adopted to construct adversarial states and attack DRL agents every time step (Huang et al. 2017). To decrease the risk of being detected, two techniques were proposed to reduce the number of attack steps. The one was that the attacker only injected perturbations when the value function estimated over the clean state was above a certain threshold, which meant that these states were critical (Kos and Song 2017). The other one identified pivotal states according to the difference between probabilities of the best action and the worst action (Lin et al. 2017; Yang et al. 2020; Sun et al. 2020). In this line of work, they leveraged the optimization-based C&W attack (Carlini and Wagner 2017) to generate adversarial perturbations.

In realistic applications, the black-box setting is more challenging and practical than the white-box setting. The common practice of black-box attacks against DRL agents retrains a substitute policy in the same environment, crafts adversarial examples with the surrogate policy by utilizing white-box methods, and attacks the target policy depending on the transferability of adversarial examples (Huang et al. 2017; Behzadan and Munir 2017). However, recent research indicates that adversarial examples for substitute networks are not always able to transfer to target networks and thus cannot probe the vulnerability of DRL (Narodytska and Kasiviswanathan 2016; Chen et al. 2017). Therefore, this paper designs an RL-based framework, which directly generates adversarial examples according to the feedback of target policies and circumvents the transferability requirement of adversarial examples crafted by substitute policies.

Moreover, current attack approaches in DRL are based on algorithms of crafting adversarial examples in image classification, e.g., FGSM and C&W. They only utilize simple pixel space metrics to encourage visual realism such as L_∞ and L_1 distance. However, adversarial examples generated by these methods look unnatural (Xiao et al. 2018; Zhao, Dua, and Singh 2018; Jandial et al. 2019). Recently, generative adversarial networks (GANs) are introduced to efficiently produce semantically more natural adversarial examples. As illustrated in Zhao, Dua, and Singh (2018), GANs-based crafting methods perturbed the benign digit image in a selective manner that slightly thickening the bottom stroke of the digit and thinning the one above it could make the target model misclassify the input image “3” into “2”, while FGSM added numerous gradient-based noises in the non-digital region of the input. Moreover, recent work has

[*] Corresponding author

demonstrated that latent features serve as better priors than input features for the generation of adversarial perturbations since latent features are more prone to alterations caused by noises (Zhao, Dua, and Singh 2018; Jandial et al. 2019).

However, current investigations of GANs-based crafting approaches mainly focus on the conventional image classification. As for sequential decision tasks, especially for the DRL setting, there is much space to explore. The biggest difference from image classification is that the generated perturbations in the current state will affect subsequent states.

To address these problems, we design an RL-based framework to model sequential relationships of perturbations and adjust perturbations only depending on the feedback of target policies in the black-box setting. Moreover, we adopt GANs with three regularization losses to generate semantically natural adversarial examples. We call this attack advRL-GAN. More specifically, the constructing process of adversarial examples is modeled as a Markov decision process (MDP) where the attack agent takes clean states as inputs and outputs perturbed latent features. To obtain rewards for instructing generations of adversarial perturbations, a generator decodes noisy latent features into adversarial examples which are then sent to the target agent for decisions. According to actions chosen by the target policy, the environment returns rewards whose opposites are regarded as rewards of the attack agent, and transfers next clean states to continue. During the process, there is no need to know model structures and internal parameters of target policies. To the best of our knowledge, this is the first RL-based attack in the DRL setting. We evaluate the attack strategy in multiple environments and illustrate the properties of generated adversarial examples through visualization.

Our contributions are listed as follows.

- We propose an RL-based adversarial attack on DRL agents in the black-box setting. It models the temporal impacts of perturbations and only receives the feedback of target agents to generate perturbations heuristically. Compared with previous black-box approaches, our attack does not need to retrain substitute policies and circumvents the transferability of adversarial examples.
- We adopt GANs as the backbone and employ three auxiliary losses to generate more aggressive and semantically more natural adversarial examples, compared to gradient-based and optimization-based attacks, e.g., FGSM and C&W.
- Experimental results demonstrate the effectiveness of the advRL-GAN attack in terms of reward reductions and magnitudes of perturbations, and validate the sparse and targeted property of adversarial perturbations.

Related Work

This section introduces adversarial attacks in DRL and techniques of crafting natural adversarial examples with GANs.

Adversarial Attacks in DRL

Since value functions and target policies in DRL are usually approximated by DNNs, the agent inherits the vulnerability of DNNs to adversarial examples. The current adversarial attacks in DRL can be divided into two categories: white-box attacks and black-box attacks. The attacker has full access to the target agent including model parameters and model architecture in the white-box setting, while black-box attacks only have access to outputs of the target agent. In realistic applications, the black-box setting is more challenging and practical than the white-box setting.

Most adversarial attacks in DRL concentrate on the white-box setting and depend on the gradient-based FGSM attack and the optimization-based C&W attack. Huang et al. (2017) made the first attempt to attack the DRL agent by employing FGSM to clean states every time step. To improve the attack efficiency and reduce the risk of being detected, Kos and Song (2017) only injected perturbations produced by FGSM into inputs either every N frames or when the trained value functions estimated over clean states were above a certain threshold. Lin et al. (2017) reduced the attack frequency from the perspective of an action preference function, which computed the probabilistic difference of the agent taking the most preferred action over the least preferred action at the current state. Following this work, Yang et al. (2020) proposed another action preference function by introducing d experts to weight revenues of taking d actions.They also adopted C&W to craft adversarial examples. Consider that previous work only focused on the current consequence, Sun et al. (2020) calculated damages of possible attack perturbations at the future states which were reached by a prediction model. Recently, Zhang et al. (2020, 2021) proposed critic-independent attacks and an optimal adversary to train robust policies. All the above attacks aim to minimize the expected accumulated rewards, while there exists a little work to lure the agent into designated target states (Lin et al. 2017; Tretschk, Oh, and Fritz 2018).

The black-box attacks in DRL are mainly based on the transferability of adversarial examples. These methods attack the target policy with adversarial examples which are crafted with white-box techniques for substitute models. One approach is to retrain surrogate policies with different algorithms, model structures or initial configurations in the same environment. Huang et al. (2017) investigated how vulnerable policies were to black-box attacks of having no knowledge of random initialization or the training algorithm. They also constructed these transferable adversarial examples with FGSM. Experimental results demonstrated that transferability across algorithms was less effective at decreasing the performance of the agent than transferability across policies. Behzadan and Munir (2017) verified the transferability of adversarial examples across different deep Q-learning network models (Mnih et al. 2015) with the Jacobian saliency map algorithm (Papernot et al. 2017). Another method is to train a sequence-to-sequence model, which imitates decisions of the target agent. Zhao et al. (2020) crafted adversarial examples with this type of prediction model by project gradient descent (PGD), an iterative version of FGSM. Gleave et al. (2019) proposed a black-box attacks in two-player games, which perturbed actions of the opponent to indirectly affect observations of the victim, rather than crafting adversarial states for the target agent. In contrast, our advRL-GAN method can directly generate black-box adversarial examples with no need to retrain substitute models or sequence-to-sequence prediction models in single-agent environments.

Crafting Natural Adversarial Examples

Although adversarial examples constructed by gradient-based and optimization-based strategies have effectively illustrated the vulnerability of DNNs, they look unnatural, i.e., the gradient-based method FGSM added multiple perturbations in the non-digital region of an image (Zhao, Dua, and Singh 2018) or synthetic images will not appear in realistic applications. To generate perceptually more realistic adversarial examples, Xiao et al. (2018) proposed the called AdvGAN method where the generator created perturbations according to original images and the discriminator distinguished the perturbed examples from the original examples. Jandial et al. (2019) explored latent features as priors to instruct productions of perturbations, since it was shown that latent features were more susceptible to adversarial perturbations than input features (Kumari et al. 2019). Meanwhile, Zhao, Dua, and Singh (2018) generated natural and legible adversarial examples from the dense and continuous representation space based on the Wasserstein GAN (WGAN) framework (Arjovsky, Chintala, and Bottou 2017).

However, these methods were all evaluated in supervised learning tasks, such as image classification. Their adaptations for DRL are not straightforward, which would require nontrivial work for model design. Similar to these methods, GAN is also adopted in this paper to generate semantically natural adversarial examples, yet for the DRL context.

Method

In this section, we first describe the context of DRL attacks and then introduce the proposed advRL-GAN attack.

Problem Statement

The DRL agent learns an optimal policy by interacting with an environment. This process is usually represented as an MDP, which can be formulated as a tuple of five elements: $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$. At time step t, assume that the agent is at the state $s_t \in \mathcal{S}$. It selects an action $a_t \in \mathcal{A}$ according to the current policy π which maps from a state s to an action a. The environment will return an immediate reward r_t and the next state s_{t+1} where $r_t = R(s_t, a_t, s_{t+1})$ and $s_{t+1} = \mathcal{P}(s_{t+1}|s = s_t, a = a_t)$. An episode starts from initial states and finishes until the agent reaches terminal states. The cumulative rewards from the time step t are defined as $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ where γ denotes the discount factor which ranges from 0 to 1 and controls the tradeoff between short-term and long-term rewards. The goal of an agent is to learn an optimal policy π^* by maximizing the expected cumulative rewards $\mathbb{E}_{\pi}[R_0]$.

The common practice to attack DRL policies is that crafting adversarial states makes the target agent perform wrong actions, which results in reduction of expected accumulated rewards in an episode. More specifically, the state at time step t is perturbed by the noise δ_t and is transformed into $s_t + \delta_t$, which leads the agent to perform a non-preferred action. The adversary tries to minimize the expected accumulated rewards $\mathbb{E}_{a_t \sim \pi(s_t+\delta_t)}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$.

In the previous work, the perturbation δ_t is computed by gradient-based or optimization-based methods. FGSM is the representative of gradient-based methods, which requires calculating the gradient of cost function $J(s_t, \pi(s_t))$ with respect to the input s_t. That is,

$$\delta_t = \epsilon \cdot \text{sign}(\nabla_{s_t} J(s_t, \pi(s_t))), \tag{1}$$

where $J(s_t, \pi(s_t))$ is the cross-entropy loss between $\pi(s_t)$ and the distribution that places all weights on the highest-weighted action in $\pi(s_t)$ and ϵ is the upper bound of L_∞-norm on the perturbations δ_t.

C&W attack is a famous optimization-based approach, which finds the optimal perturbation δ_t by solving the following optimization problem:

$$\begin{aligned} \min\nolimits_{\delta_t} \; & ||\delta_t||_p + c \cdot \pi(s_t + \delta_t) \\ & s.t. \; s_t + \delta_t \in [0,1]^n, \end{aligned} \tag{2}$$

where c controls the balance between the magnitude of perturbations and the strength of perturbations, and $s_t + \delta_t \in [0,1]^n$ confines the adversarial example $s_t + \delta_t$ to the valid image space since the original image s_t is transformed into $[0,1]^n$, where n is the dimensionality of s_t.

Although these crafting methods restrict differences between benign and adversary states to be as subtle as possible, they produce perturbations only in pixel space which make generated adversarial examples look unnatural. Moreover, latent features are known to be more vulnerable to perturbations than input features. Hence, we find adversarial perturbations in the latent space and adopt the GANs-based framework to generate semantically natural adversarial examples.

AdvRL-GAN Attack

There exist two difficulties in adversarial attacks on DRL agents. Firstly, the perturbation in the current state will affect subsequent states. Secondly, the environment only returns reward signals for perturbations in the DRL setting rather than classification errors in supervised tasks, which is not capable of backpropagation. To address these, we formulate adversarial attacks in DRL into an RL agent, which models the temporal relationships of perturbations and explores optimal perturbations depending on the rewards of target agents. We call it advRL-GAN. Figure 1 illustrates the overall framework of advRL-GAN attack. The purple box indicates the attack agent and other components make up the attack environment. The attack agent produces perturbed latent features from clean states. The attack environment returns corresponding rewards after the target agent is attacked by adversarial states, which are decoded by the generator and constrained by the discriminator.

The target agent $\mathcal{T}$ corresponds to an MDP $\mathcal{M}_{target} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$. The MDP of advRL-GAN attack is $\mathcal{M}_{attack} = \langle \mathcal{S}, \tilde{\mathcal{Z}}, \mathcal{P}, \tilde{R}, \gamma \rangle$ where $\mathcal{S}$ and $\mathcal{P}$ are the same as that of the target agent, the action space $\tilde{\mathcal{Z}}$ means the perturbed latent space, and the reward function $\tilde{R} = -R$. The

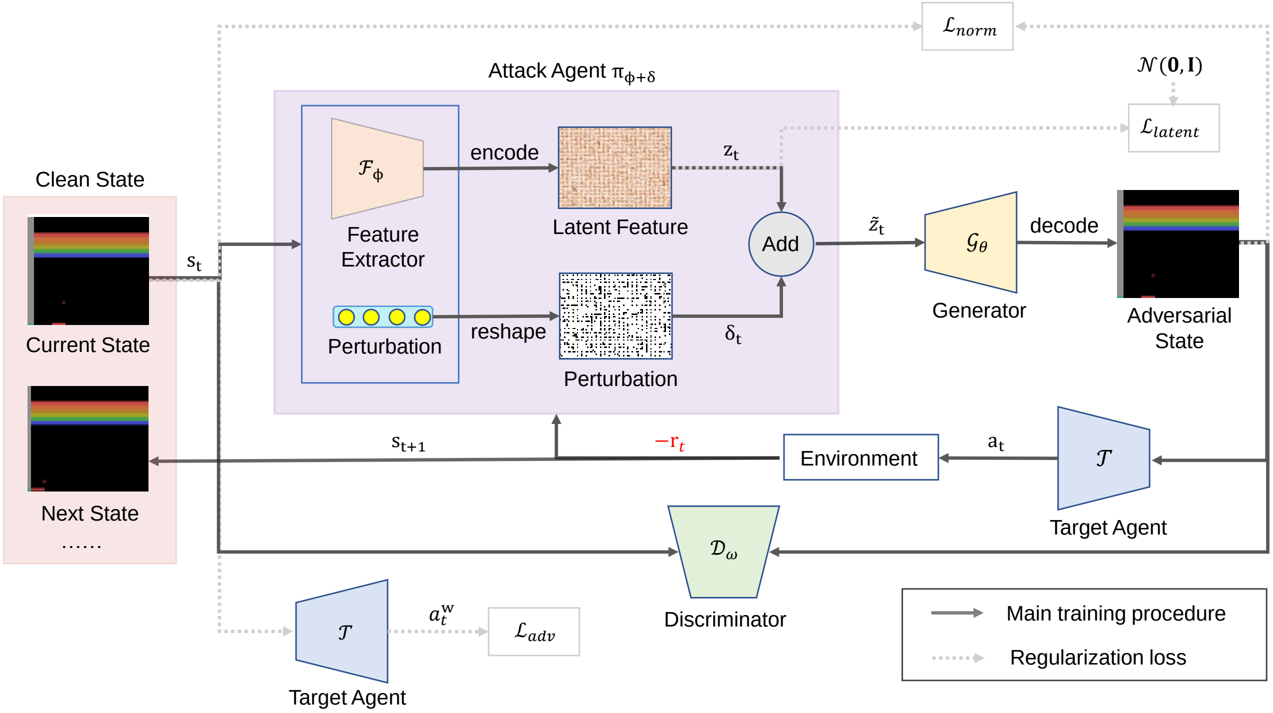


Figure 1: Overall architecture of the advRL-GAN attack. It mainly consists of three modules: the attack agent $\pi_{\phi+\delta}$ aims to produce the perturbed latent feature $\tilde{z}_t$ from the current clean state s_t where the quality of perturbation is evaluated by the target agent $\mathcal{T}$; the generator $\mathcal{G}_\theta$ struggles to generate indistinguishable and semantically natural adversarial state $\tilde{s}_t$; the discriminator $\mathcal{D}_\omega$ tries to distinguish the clean state from the adversarial state.

goal of the attack agent $\pi_{\phi+\delta}$ is to maximize expected cumulative rewards $\mathbb{E}_{\pi_{\phi+\delta}}\left[\sum_{t=0}^{\infty}\gamma^t \tilde{R}(s_t,\tilde{z}_t,s_{t+1})\right]$ where $\tilde{z}_t$ is the perturbed latent feature at the current state s_t. More specifically, the attack agent is implemented by a feature extractor $\mathcal{F}_\phi$ and a perturbation vector, which takes a clean state s_t as an input. $\mathcal{F}_\phi$ outputs the encoded latent representation z_t and meanwhile the vector reshapes itself to form a perturbation δ_t which has the same shape of z_t. By adding z_t and δ_t, the attack agent produces the noised latent representation $\tilde{z}_t$ as an action given the current state s_t.

The attack environment consists of three parts: a generator $\mathcal{G}_\theta$, a discriminator $\mathcal{D}_\omega$, and a target agent $\mathcal{T}$. The generator $\mathcal{G}_\theta$ aims to generate indistinguishable and semantically natural adversarial state $\tilde{s}_t$ with the perturbed latent feature $\tilde{z}_t$. Then $\tilde{s}_t$ and s_t are fed into the discriminator $\mathcal{D}_\omega$ for classification. To evaluate the quality of perturbation and instruct the attack agent to produce optimal perturbations, $\tilde{s}_t$ is also sent to the target agent $\mathcal{T}$ for decisions. $\mathcal{T}$ takes an action according to the target policy $\pi_\mathcal{T}$, which is trained in the target environment in advance. The target environment subsequently returns corresponding reward r_t and the next state s_{t+1}. We finally take the negative reward as the feedback for the attack agent and continue next step with s_{t+1}.

To generate semantically natural adversarial examples, we adopt WGAN as the backbone with auxiliary losses to realize the advRL-GAN attack. The training process consists of three stages: generation, discrimination, and attack.

Generation In this stage, the attack agent $\pi_{\phi+\delta}$ combines the generator $\mathcal{G}_\theta$ to generate aggressive, semantically natural, and indistinguishable adversarial states. To achieve this, we design three regularization losses except the following raw optimization objective of the generator in WGAN

$$\mathcal{L}_G(\theta,\phi,\delta) = -\mathbb{E}_{s_t}\left[\mathcal{D}_\omega(\mathcal{G}_\theta(\pi_{\phi+\delta}(s_t)))\right]. \quad (3)$$

Firstly, in conventional classification tasks, the attack goal is to decrease classification accuracy which can be achieved by classifying adversarial examples into wrong categories. In contrast to this, the goal of attacking DRL agents is to reduce expected accumulated rewards as much as possible. It means that the agent should take the worst action. However, if just decreasing the probability of taking the optimal action, the probability of taking the worst action does not necessarily increase. Therefore, we enforce the attack agent to produce more strong adversarial perturbations, which make the target agent take the worst action a_t^w rather than any non-optimal action. It leads to an optimal adversarial attack against DRL agents whose loss function is formulated as

$$\mathcal{L}_{adv} = -\mathbb{E}_{s_t}\left[\sum_{i=1}^{|\mathcal{A}|} p\left(a_t^i|s_t\right)\log \pi_{\phi+\delta}(s_t)\right], \quad (4)$$

where a_t is taken by the target policy $\pi_\mathcal{T}$ and the adversarial probability distribution $p(a_t|s_t)$ is denoted by

$$p(a_t^i|s_t) = \begin{cases} 1, & \text{if } a_t^i = \arg\min \pi_\mathcal{T}(s_t) \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

Note that there is no need to know internal parameters and model structures of the target agent in this process, we only require the worst action given in the current state according to the target policy.

Secondly, to generate semantically natural and legible adversarial examples which lie on the data manifold, we minimize the divergence between the probability distribution of perturbed latent features and the prior distribution. Here the prior is assumed to be an isotropic Gaussian distribution, i.e., $p(\tilde{z}_t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, to encourage the perturbed latent space to be normally distributed. The constraint on the latent space is

$$\mathcal{L}_{latent} = \mathbb{E}_{s_t}\left[\mathbb{KL}\left(\pi_{\phi+\delta}(s_t) \,||\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right)\right]. \tag{6}$$

Finally, in order to generate perceptually realistic adversarial examples, we restrict the L_1 norm of differences between original states and adversarial states to be as small as possible. The similarity constraint on the pixel space can be formulated as

$$\mathcal{L}_{norm} = \mathbb{E}_{s_t}\left[\left|\mathcal{G}_\theta(\pi_{\phi+\delta}(s_t)) - s_t\right|\right]. \tag{7}$$

To sum up, the optimization objective of the generation stage is given as

$$\begin{aligned}\mathcal{L}_G(\theta, \phi, \delta) = &-\mathbb{E}_{s_t}\left[\mathcal{D}_\omega(\mathcal{G}_\theta(\pi_{\phi+\delta}(s_t)))\right] \\ &+ \lambda_1 \cdot \mathcal{L}_{adv} + \lambda_2 \cdot \mathcal{L}_{latent} + \lambda_3 \cdot \mathcal{L}_{norm},\end{aligned} \tag{8}$$

where λ_1, λ_2, and λ_3 are regularization coefficients.

Discrimination The discrimination stage updates the discriminator $\mathcal{D}_\omega$, which aims to distinguish raw states from adversarial states. We adopt the original optimization objective of WGAN which is written as

$$\mathcal{L}_D(\omega) = \mathbb{E}_{s_t}\left[\mathcal{D}_\omega(\mathcal{G}_\theta(\pi_{\phi+\delta}(s_t)))\right] - \mathbb{E}_{s_t}\left[\mathcal{D}_\omega(s_t)\right]. \tag{9}$$

Attack The attack stage sends generated adversarial states to attack the target agent. Since there is no gradient information in the black-box setting, we design a reward signal mechanism to evaluate the performance of current perturbation. Specifically, the target environment will return a reward according to an action taken by the target agent, the negative of which will be regarded as the reward of the perturbation.

We train the attack policy with the policy gradient algorithm REINFORCE which maximizes the expectation of accumulated rewards in an episode via the gradient ascent mechanism. The objective function is written as

$$\begin{aligned}J(\phi, \delta) &= \text{arg max}_{\phi,\delta}\, \mathbb{E}_{\pi_{\phi+\delta}}\left[\sum\nolimits_{t=0}^{T} \gamma^t \tilde{R}(s_t, \tilde{z}_t, s_{t+1})\right] \\ &= \text{arg max}_{\phi,\delta}\, \mathbb{E}_{\pi_{\phi+\delta}}\left[-\sum\nolimits_{t=0}^{T} \gamma^t R(s_t, a_t, s_{t+1})\right],\end{aligned} \tag{10}$$

where T is the length of an episode. Employing the policy gradient theorem (Sutton and Barto 2018), we obtain the derivative $\nabla_{\phi,\delta} J(\phi, \delta)$ as follows

$$\begin{aligned}&\nabla_{\phi,\delta} J(\phi, \delta) = \\ &\mathbb{E}_{\pi_{\phi+\delta}}\left[-\sum_{t=0}^{T} \nabla_{\phi,\delta} \log \pi_{\phi+\delta}(s_t) \sum_{t=0}^{T} \gamma^t R(s_t, a_t, s_{t+1})\right].\end{aligned} \tag{11}$$

Algorithm 1: AdvRL-GAN Attack

```
Input: Regularization coefficients $\lambda_1$, $\lambda_2$, and $\lambda_3$;
Output: Learned attack policy $\pi_{\phi+\delta}$ and generator $\mathcal{G}_\theta$;
Randomly initialize attack policy $\pi_{\phi+\delta}$, generator $\mathcal{G}_\theta$,
  discriminator $\mathcal{D}_\omega$;
for each episode do
    Initialize the clean state $s_t = env.reset()$;
    for each step $t$ do
        \\ Generation Stage
        Record the worst action $a_t^w = arg\ min\ \pi_{\mathcal{T}}(s_t)$;
        Produce perturbed latent feature $\tilde{z}_t = \pi_{\phi+\delta}(s_t)$;
        Generate the adversarial state $\tilde{s}_t = \mathcal{G}_\theta(\tilde{z}_t)$;
        Compute losses $\mathcal{L}_{adv}$, $\mathcal{L}_{latent}$, and $\mathcal{L}_{norm}$;
        Send $\tilde{s}_t$ to $\mathcal{D}_\omega$ and update $\theta$, $\phi$, and $\delta$ by descending
          their stochastic gradients of $\mathcal{L}_G(\theta, \phi, \delta)$;
        \\ Discrimination Stage
        Send $s_t$ and $\tilde{s}_t$ to $\mathcal{D}_\omega$ and update $\omega$ by descending
          its stochastic gradient of $\mathcal{L}_D(\omega)$;
        \\ Collect Samples;
        $a_t = arg\ max\ \pi_{\mathcal{T}}(\tilde{s}_t)$;
        $r_t, s_{t+1} = env.step(a_t)$;
        Collect transitions $(s_t, \tilde{z}_t, -r_t, s_{t+1})$;
        $s_t = s_{t+1}$;
    end for
    \\ Attack Stage
    Update $\phi$ and $\delta$ with transitions by ascending their
      stochastic gradients of $J(\phi, \delta)$;
end for
```

It is intuitive that the attack policy will tend to increase the probabilities of strong perturbations, which makes the target agent achieve less rewards. Since reward signals are only related to outputs of the target agent, note again that there is no need to know model architecture and internal parameters.

Algorithm 1 outlines the training procedure of the proposed advRL-GAN attack.

Experiments

We evaluate the performance of the proposed advRL-GAN attack against DRL agents on 6 different Atari 2600 games, including Breakout, Pong, Chopper Command, Space Invaders, MsPacman, and Qbert, using OpenAI Gym (Brockman et al. 2016). For detailed descriptions about these environments, refer to supplementary material. Experiments are performed with the PyTorch library on GeForce RTX 2080Ti GPUs. We first introduce experimental configurations, then present attack methods to compare, and finally discuss experimental results in detail.

Configurations

Target policies are trained with the asynchronous advantage actor-critic (A3C) algorithm (Babaeizadeh et al. 2017) of 32 actor-learner threads, which is one of the state-of-the-art algorithms in DRL. We train the advRL-GAN model with 8×10^5 steps by adopting the REINFORCE algorithm. The hyper-parameters of the loss function $\mathcal{L}_G$ are set to $\lambda_1 = 10$, $\lambda_2 = 0.1$, and $\lambda_3 = 10$.

Comparing Methods

We compare advRL-GAN with other attacks in both white-box and black-box settings. For black-box attacks, substitute policies trained with different network architectures are employed to craft transferable adversarial examples. Brief introductions of the comparing methods are given below.

FGSM: To apply FGSM in the DRL setting, Huang et al. (2017) defined the cost function as the cross-entropy loss $J(s_t, \pi(s_t))$ between probabilities of actions and the distribution that places all weights on the optimal action, and calculated gradients of the cost function with respect to different states. Formally, $\delta_t = \epsilon \cdot \text{sign}(\nabla_{s_t} J(s_t, \pi(s_t)))$.

RAND+FGSM: Tramèr et al. (2018) enhanced the power of FGSM by applying a small perturbation before utilizing FGSM. In experiments, we add random noise at first for $\alpha < \epsilon$, that is, $s_{rand} = s + \alpha \cdot \text{ sign}(\mathcal{N}(\mathbf{0}, \mathbf{I}))$. Finally, RAND+FGSM creates adversarial examples by $s_{adv} = s_{rand} + (\epsilon - \alpha) \cdot \text{ sign}(\nabla_{s_{rand}} J(s_{rand}, \pi(s_{rand})))$.

Skip-Frame: This attack is also based on the FGSM strategy and determines the attack frequency with probability ρ in an episode (Kos and Song 2017). It is more random than the FGSM method and the strategically-timed attack.

Strategically-Timed: Lin et al. (2017) proposed the strategically-timed attack to reduce the frequency of perturbations. They only injected perturbations when the difference of value preference between the best action and the worst action was above a certain threshold β.

PGD: PGD is an iteration version of FGSM which adds a slight perturbation at each step (Madry et al. 2018). It can be formalized as: $s_0 = s_t, s_n = Proj\{s_{n-1} + \alpha \cdot \text{sign}(\nabla_{s_{n-1}} J(s_{n-1}, \pi(s_{n-1})))\}$ where $\alpha \cdot n = \epsilon$ and $Proj$ projects intermediate adversarial states into the ϵ-L_∞ neighbor of benign states. In experiments, n is set to be 10 .

C&W: To generate adversarial perturbations, Carlini and Wagner (2017) minimized the objective $||\delta||_p + c \cdot f(x + \delta)$ such that $x + \delta \in [0, 1]^n$. We craft adversarial states with transformation of this optimization objective, that is, $||\delta_t||_2 + c \cdot \pi(s_t + \delta_t)$ such that $s_t + \delta_t \in [0, 1]^n$.

Results

In this section, we present expected accumulated rewards after target policies are attacked by different approaches and visualize differences between clean states and adversarial states. Furthermore, we conduct an ablation study to illustrate the importance of each regularization term.

Attack performance Tables 1, 2, and 3 list comparisons among the advRL-GAN attack and other attacks in Breakout, Pong, and Chopper Command games, respectively. Attack results in other environments are presented in supplementary material. All results are averaged over 10 trials. The column Perturbation denotes the settings of hyper-parameters in various attacks. The columns White and Black record expected accumulated rewards of target policies after the target policies are attacked in the white-box and the black-box settings, respectively. The column Dist computes differences between clean states and adversarial states with the L_1 norm whose unit is 10^{-4}.

We can observe that these white-box attacks extremely degrade the expected accumulated rewards. It is reasonable since the target policies reveal more information to the adversary in the white-box setting, e.g., gradients. Our black-box advRL-GAN attack also significantly degrades expected accumulated rewards and performs competitively compared with some white-box attacks in simple games, e.g., Breakout and Pong, where the action space is relatively small.

Table 1: Comparisons among the advRL-GAN attack and other attacks in the Breakout game. The expected accumulated reward achieved by the target policy is 345.83. The bold presents the best attack performance.

Attack Methods	Perturbation	White	Black	Dist
FGSM	$\epsilon = .0003$	55.30	142.40	1.95
	$\epsilon = .0005$	18.90	116.60	3.27
RAND+FGSM	$\epsilon = .0004, \alpha = .0002$	41.90	73.00	1.30
	$\epsilon = .0006, \alpha = .0003$	34.40	64.89	1.96
Skip-Frame	$\epsilon = .0003, \rho = .5$	76.11	142.20	1.94
	$\epsilon = .0005, \rho = .5$	31.45	82.10	3.26
Strategically-Timed	$\epsilon = .0003, \beta = .8$	57.73	62.10	1.94
PGD	$\epsilon = .0003, n = 10$	58.30	80.10	1.90
C&W	-	**3.50**	44.40	3.55
advRL-GAN	-	-	**38.23**	**0.95**

Table 2: Comparisons among the advRL-GAN attack and other attacks in the Pong game. The expected accumulated reward achieved by the target policy is 20.62. The bold presents the best attack performance.

Attack Methods	Perturbation	White	Black	Dist
FGSM	$\epsilon = .001$	-20.70	20.58	9.50
	$\epsilon = .003$	-21.00	19.50	30.00
RAND+FGSM	$\epsilon = .001, \alpha = .0005$	-18.60	20.56	5.00
	$\epsilon = .003, \alpha = .0015$	**-22.00**	20.43	15.01
Skip-Frame	$\epsilon = .001, \rho = .5$	-11.42	20.80	10.00
	$\epsilon = .003, \rho = .5$	-16.33	20.20	30.00
Strategically-Timed	$\epsilon = .001, \beta = .8$	-13.40	20.90	10.00
PGD	$\epsilon = .001, n = 10$	-20.40	20.90	8.33
C&W	-	-19.50	20.53	4.67
advRL-GAN	-	-	**-12.07**	**1.55**

Generally, the advRL-GAN attack generates less adversarial perturbations and reduces more rewards compared with other black-box attacks in all the three environments. Furthermore, for the Pong game, adversarial examples crafted by these gradient-based and optimization-based attacks with the substitute policy can not transfer well to attack the target policy, while the advRL-GAN attack still drastically declines the performance of the target policy with the least perturbations. These demonstrate that the advRL-GAN attack achieves the state-of-the-art performance in attacking DRL policies within the black-box setting.

Visualization Since adversarial states seem to be the same as benign states from the perspective of human beings, we display their differences to illustrate the mechanisms of perturbing in different attacks. Figure 2 shows benign states, adversarial states, and their differences with various attacks in Breakout, Pong, and Chopper Command games. For visualization of other games, refer to supplementary material.

We can observe that advRL-GAN perturbs benign states

Table 3: Comparisons among the advRL-GAN attack and other attacks in the Chopper Command game. The expected accumulated reward achieved by the target policy is 3260.00. The bold presents the best attack performance.

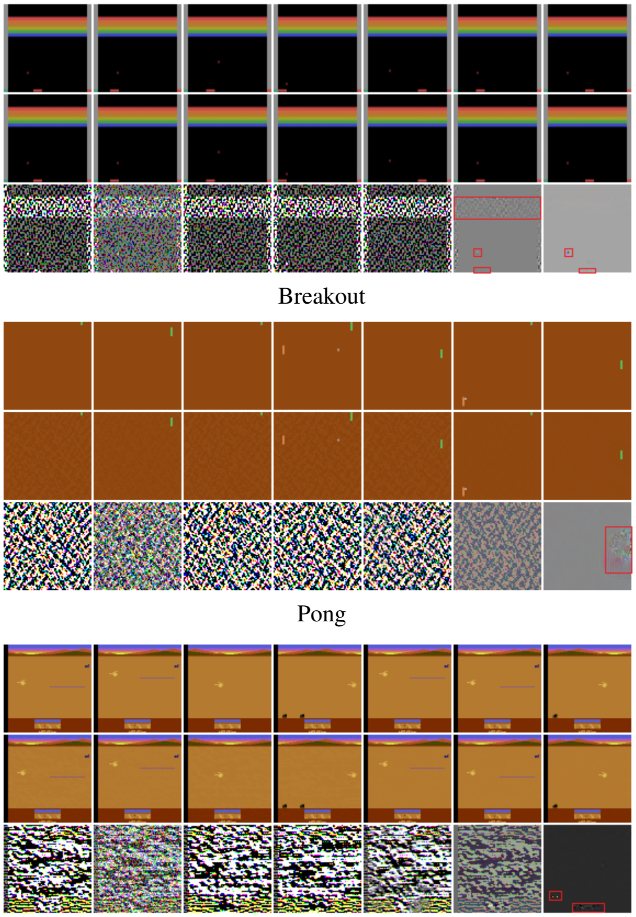
Attack Methods	Perturbation	White	Black	Dist
FGSM	$\epsilon = .005$	850.00	2390.00	47.44
	$\epsilon = .01$	440.00	1980.00	88.76
RAND+FGSM	$\epsilon = .005, \alpha = .0025$	1530.00	2350.00	23.77
	$\epsilon = .01, \alpha = .005$	730.00	2930.00	47.50
Skip-Frame	$\epsilon = .005, \rho = .5$	1530.00	2714.29	47.55
	$\epsilon = .01, \rho = .5$	1180.00	2337.50	94.90
Strategically-Timed	$\epsilon = .01, \beta = .8$	430.00	2190.00	88.45
PGD	$\epsilon = .005, n = 10$	540.00	2540.00	37.73
C&W	-	**250.00**	1890.00	35.91
advRL-GAN	-	-	**1766.67**	**18.15**

in a sparse and targeted manner, while other attacks inject noises in random manners. Specifically, FGSM-based attacks perturb all regions of an image in all the three environments. For the C&W attack, it injects perturbations in critical regions of blocks, the ball, and the paddle in Breakout, while it perturbs the whole image in Pong and Chopper Command. In contrast, advRL-GAN disturbs crucial regions of the paddle and the ball in Breakout, adds noises in the key area of the right paddle which will catch the ball in Pong, and perturbs the critical range of the protected truck in Chopper Command. These demonstrate that advRL-GAN can generate semantically more natural adversarial examples.

Ablation study To illustrate the effectiveness of each regularization term, we conduct an ablation study in the six environments. We eliminate the discriminator $\mathcal{D}_\omega$, the $\mathcal{L}_{adv}$ loss, the $\mathcal{L}_{latent}$ loss, or the $\mathcal{L}_{norm}$ loss, and retrain with the same training steps. Table 4 lists rewards achieved by attacked target policies and distances (10^{-4}) between benign states and adversarial states in different settings. In most cases, the full advRL-GAN attack achieves the best attack performance and the smallest perturbation distances. Even though deleting the discriminator could degrade more rewards in Pong, Space Invaders, and MsPacman games, it causes perturbations at least three times larger than that caused by the full advRL-GAN attack, which will increase the risk of being detected by defenders.

Table 4: Rewards and Dist (10^{-4}) under eliminating different components of the advRL-GAN framework in the six environments. The best performance is shown in bold.

Environment	metrics	-$\mathcal{D}_\omega$	-$\mathcal{L}_{adv}$	-$\mathcal{L}_{latent}$	-$\mathcal{L}_{norm}$	advRL-GAN
Breakout	Rewards	44.33	65.8	41.02	72.60	**38.23**
	Dist	3.48	6.10	4.78	6.55	**0.95**
Pong	Rewards	**-20.70**	20.50	20.90	20.10	-12.07
	Dist	29.65	5.19	7.64	17.29	**1.55**
Chopper Command	Rewards	2033.34	1850.00	2510.00	2116.67	**1766.67**
	Dist	74.87	51.39	30.68	67.88	**18.15**
Space Invaders	Rewards	**173.00**	812.00	901.50	875.50	724.00
	Dist	444.76	28.94	28.04	29.29	**21.15**
MsPacman	Rewards	**1619.00**	2691.00	2533.00	2518.00	2145.71
	Dist	96.67	43.61	35.97	48.69	**31.20**
Qbert	Rewards	9640.00	9920.00	9885.00	9820.50	**9250.00**
	Dist	49.89	31.63	23.64	26.52	**22.99**



Breakout

Pong

Chopper Command

Figure 2: Visualization of benign states (first row), adversarial states crafted by various attacks (second row), and their differences (third row). Seven columns correspond to FGSM, RAND+FGSM, Skip-Frame, Strategically-Timed, PGD, C&W, and advRL-GAN, respectively. Red boxes indicate regions perturbed by attacks.

Conclusion

In this paper, we have proposed an advRL-GAN framework to directly generate semantically natural adversarial examples for DRL policies in the black-box setting. The framework is formulated into an RL problem, which models sequential relationships of perturbations and only receives outputs of target policies as instructions for the generation of perturbations. It adopts WGAN as the backbone with three auxiliary losses to generate semantically natural adversarial examples. Experimental results have demonstrated that the advRL-GAN attack reduces the most rewards and produces the least magnitudes of perturbations in the black-box setting, and surprisingly achieves competitive performance with white-box attacks in some environments. Further visualization has illustrated that advRL-GAN injects perturbations in a sparse and targeted manner, and creates semantically more natural adversarial examples.

Acknowledgement This work was supported by the NSFC Projects 62076096 and 62006078, the Shanghai Municipal Project 20511100900, and the Open Research Fund of KLATASDS-MOE.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, 214–223.

Babaeizadeh, M.; Frosio, I.; Tyree, S.; Clemons, J.; and Kautz, J. 2017. Reinforcement learning through asynchronous advantage actor-critic on a GPU. In *International Conference on Learning Representations*.

Behzadan, V.; and Munir, A. 2017. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, 262–275.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540*.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 39–57.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM Workshop on Artificial Intelligence and Security*, 15–26.

Gleave, A.; Dennis, M.; Wild, C.; Kant, N.; Levine, S.; and Russell, S. 2019. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; and Abbeel, P. 2017. Adversarial attacks on neural network policies. In *International Conference on Learning Representations*.

Isele, D.; Rahimi, R.; Cosgun, A.; Subramanian, K.; and Fujimura, K. 2018. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning. In *International Conference on Robotics and Automation*, 2034–2039.

Jandial, S.; Mangla, P.; Varshney, S.; and Balasubramanian, V. 2019. AdvGAN++: Harnessing latent layers for adversary generation. In *International Conference on Computer Vision Workshops*, 2045–2048.

Kos, J.; and Song, D. 2017. Delving into adversarial attacks on deep policies. In *International Conference on Learning Representations*.

Kumari, N.; Singh, M.; Sinha, A.; Machiraju, H.; Krishnamurthy, B.; and Balasubramanian, V. N. 2019. Harnessing the vulnerability of latent layers in adversarially trained models. In *International Joint Conference Artifical Intelligence*, 2779–2785.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial examples in the physical world. In *International Conference on Learning Representations*.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1): 1334–1373.

Lin, Y.-C.; Hong, Z.-W.; Liao, Y.-H.; Shih, M.-L.; Liu, M.-Y.; and Sun, M. 2017. Tactics of adversarial attack on deep reinforcement learning agents. In *International Joint Conference on Artificial Intelligence*, 3756–3762.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Narodytska, N.; and Kasiviswanathan, S. P. 2016. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Asia Conference on Computer and Communications Security*, 506–519.

Sun, J.; Zhang, T.; Xie, X.; Ma, L.; Zheng, Y.; Chen, K.; and Liu, Y. 2020. Stealthy and efficient adversarial attacks against deep reinforcement learning. In *Association for the Advance of Artificial Intelligence*, 5883–5891.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *International Conference on Learning Represenations*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2018. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*.

Tretschk, E.; Oh, S. J.; and Fritz, M. 2018. Sequential attacks on agents for long-term adversarial goals. In *ACM Computer Science in Cars Symposium*.

Xiao, C.; Li, B.; Zhu, J.-Y.; He, W.; Liu, M.; and Song, D. 2018. Generating adversarial examples with adversarial networks. In *International Joint Conference on Artificial Intelligence*, 3905–3911.

Yang, C.-H. H.; Qi, J.; Chen, P.-Y.; Ouyang, Y.; Hung, I.-T. D.; Lee, C.-H.; and Ma, X. 2020. Enhanced adversarial strategically-timed attacks against deep reinforcement learning. In *International Conference on Acoustics, Speech and Signal Processing*, 3407–3411.

Zhang, H.; Chen, H.; Boning, D. S.; and Hsieh, C.-J. 2021. Robust reinforcement learning on state observations with learned optimal adversary. In *International Conference on Learning Representations*.

Zhang, H.; Chen, H.; Xiao, C.; Li, B.; Liu, M.; Boning, D.; and Hsieh, C.-J. 2020. Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33: 21024–21037.

Zhao, Y.; Shumailov, I.; Cui, H.; Gao, X.; Mullins, R.; and Anderson, R. 2020. Blackbox attacks on reinforcement learning agents using approximated temporal information. In *International Conference on Dependable Systems and Networks Workshops*, 16–24.

Zhao, Z.; Dua, D.; and Singh, S. 2018. Generating natural adversarial examples. In *International Conference on Learning Representations*.