Applied Intelligence (2025) 55:327
https://doi.org/10.1007/s10489-024-05974-1

Fooling machine learning models: a novel out-of-distribution attack through generative adversarial networks

Hailong Hu[1,2] · **Jun Pang**[2,3]

Accepted: 4 October 2024 / Published online: 16 January 2025

Abstract

Recent advancements in machine learning (ML) have facilitated the deployment of ML models across various real-world applications. However, these ML models might suffer from various potential security threats. In this paper, we propose a novel out-of-distribution attack: Leveraging pre-trained generative adversarial networks (GANs), an adversary aims to fool an ML model and make the model misclassify a sample from GANs as a pre-specified target class. Our attack is based on the insight that ML models do not know when they do not know, and ML models can unexpectedly recognize a completely different sample (e.g. cartoon face) as a certain class (e.g. airplane) with high confidence. Specifically, we introduce a targeted attack framework through GANs for white-box and black-box scenarios. Our framework casts this problem as an optimization problem and a family of attack methods are developed. Extensive experimental results show that our methods can achieve competitive performance, even compared with several state-of-the-art adversarial example attacks. Furthermore, our methods can evade several widely-used and the latest defenses. We also elaborately analyze various factors that affect the attack performance. Our work will provide a supplementary test to comprehensively evaluate the robustness of ML systems.

Keywords Out-of-distribution attacks · Out-of-distribution detection · Generative adversarial networks · Robustness in machine learning · Open world recognition

1 Introduction

Recent years have witnessed significant progress in machine learning (ML), ranging from computer vision [1–3] to natural language processing [4–6]. The success of ML has also enabled technology companies to deploy various ML-based applications in the real world, including safety-critical applications such as self-driving cars. Despite these advances, ML models face various security threats in the open world [7–12].

A substantial body of research has exposed the vulnerability of ML models to adversarial example attacks [13–17]. Given a correctly classified example, an adversary can make an adversarial example by adding imperceptible perturbations, which causes the ML model to change its prediction result [14]. From the perspective of human vision, these imperceptible perturbations totally do not affect the category of the example, but the ML model has made a different decision. In other words, ML models are easy to be fooled by adversarial perturbations and are not robust in knowing what they know. In addition, another line of research studies demonstrates that for samples that are far from the training data or are completely unrecognizable to human beings (e.g. noises), ML models also classify them as a particular class with high confidence. To put it another way, *ML models do not know when they do not know* [18–20].

Indeed, when ML models are deployed in the real world, any type of input sample could occur and these models face the risk fooled by adversarial samples. Considering the prevalence and availability of pre-trained generative adversarial networks (GANs), this motivates us to think about a new potential threat: generated samples from a well-trained GAN might be utilized by adversaries to fool ML models and evade detection, even if ML models are protected.

Our code is available at: https://github.com/HailongHuPri/OODGANAttack.

✉ Hailong Hu
huhl5861@gmail.com

Jun Pang
jun.pang@uni.lu

[1] National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China

[2] Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette 4365, Luxembourg

[3] Faculty of Science, Technology and Medicine, University of Luxembourg, Esch-sur-Alzette 4365, Luxembourg

In this paper, we introduce a novel *out-of-distribution* (OOD) attack in which an adversary aims to craft a completely different sample (e.g. cartoon face) to deceive an ML model into recognizing a certain class that the ML model has learned (e.g. airplane). Concretely, leveraging off-the-shelf pre-trained GANs, we propose a novel targeted attack framework, which leads ML models to make a particular prediction for inputs from a GAN. For consistency with prior literature [21, 22], we refer to these generated examples as OOD examples, characterized by a distribution that differs from the training set of the victim model.

In general, ML models make arbitrary predictions for OOD examples because they indeed do not know OOD examples. Taking advantage of a pre-trained GAN, our attack framework attempts to construct generated samples so that the victim ML model misclassifies them as a particular class that the adversary wishes. More broadly, our attack fools ML models into knowing what they actually do not know. In addition, due to the diverse and numerous samples generated by pre-trained GAN models, our attack framework provides a new angle to find adversarial examples that state-of-the-art defense mechanisms do not consider.

Technically, we formulate our attack as an optimization problem: finding a generated example from a pre-trained GAN model that minimizes the distance between the victim model's prediction and a target class specified by the adversary. Under the white-box and black-box attack scenarios, gradient-based and non-gradient based optimization methods are proposed to obtain the generated sample. Extensive experimental evaluations demonstrate that our attack methods are highly effective, achieving over 97% average attack success rate on the white-box scenario and at least 62% average attack success rate on the black-box scenario (see Section 5.1).

Moreover, we compare our attack methods with eight state-of-the-art adversarial example attacks, including C&W [23], PGD [15], AutoPGD [24] and CGA [25] in the white-box scenario, as well as ZOO [26], DBA [16], HSJA [17] and SignOPT [27] in the black-box scenario. Extensive experimental evaluations show that the performance of our attack methods is still competitive (see Section 5.3). Here, we underline that the mechanism of our attack methods is totally different from that of adversarial example attacks. That is, the OOD adversarial example is generated from a well-trained GAN rather than an image with perturbations. While both attacks have the same attack objective — fooling ML models, our methods provide a novel angle of attack against ML models. We further evaluate our attack methods by investigating whether OOD detection techniques can detect these OOD adversarial examples. Experimental results demonstrate that our attack methods can evade three state-of-the-art OOD detection techniques, i.e. ODIN [28], OE [29] and ATOM [30] (see Section 5.3). We also analyze our attack performance from various aspects, encompassing five types of loss functions, two optimization methods, and presenting the optimization processes in both white-box and black-box scenarios (see Section 5.4).

Contributions Our contributions in this paper are threefold.

(1) We introduce a novel and unified targeted attack framework for white-box and black-box scenarios, which is the first work to study out-of-distribution attacks through off-the-shelf pre-trained GANs.
(2) We thoroughly evaluate our attack methods on various victims trained on various datasets and expand our method to attack models with classical and state-of-the-art defense measures.
(3) We systematically analyze factors affecting attack performance, including various loss functions, optimization methods and optimization processes in both attack scenarios.

Ethics statement Our primary goal is to advance the development of robust defense measures for machine learning models, and our methods can serve as as pivotal tools for conducting both white-box and black-box testing, thereby facilitating the safe deployment of machine learning models in real-world applications. While we recognize the potential for malicious use of these methods, we firmly believe that the benefits that our work provides to researchers designing new defenses significantly mitigate the associated risks. In addition, developing strong attacks is essential for constructing robust machine learning models.

Organization The remainder of this paper is structured as follows. Section 2 provides an overview of related works. Section 3 outlines our unified attack framework. Section 4 describes experimental setups and Section 5 presents our experimental results. We discuss the paper in Section 6 and conclude the paper in Section 7 with future work.

2 Related work

In this section, we provide a comprehensive review of previous research in relation to our method, organized into three main areas: adversarial example attacks, out-of-distribution detection, and generative adversarial networks.

2.1 Adversarial example attacks

Depending on whether the adversary has access to the whole victim model, adversarial example attacks can be broadly categorized into two types: white-box and black-box attacks. White-box adversarial example attacks generally leverage gradient information of the victim model to perturb exam-

ples [13, 14, 31]. Goodfellow et al. introduce a fast gradient sign method to generate adversarial examples under the ℓ_∞ norm. This attack method only involves a single-step gradient update [31]. Furthermore, Madry et al. develop an iterative method — Projected Gradient Descent (PGD), to generate more powerful adversarial examples, and an adversarial training method is also proposed [15] to enhance robustness. Croce et al. improve the PGD attack method via addressing challenges related to suboptimal step size and the objective function's limitations [24]. Yamamura et al. propose to employ the conjugate gradient method, known for its effectiveness in overcoming the ill-conditioned problem, to further enhance the attack performance [25]. Unlike previous works, the C&W attack method leverages new objective functions and converts the box constrained optimization problem into an unconstrained problem to construct adversarial examples [23].

Because it is impossible to directly compute the gradients based on the victim model under the black-box scenario, various gradient estimation attack methods are proposed. The ZOO attack method, for instance, estimates the gradients through monitoring the changes of prediction confidence [26]. Chen et al. utilize the binary information at the decision boundary to approximate the gradient direction [17]. Furthermore, Cheng et al. propose an attack framework that estimates the sign of the gradient in any direction, aiming to reduce the number of queries for effective attacks [27]. Instead of making an estimate of gradient information, Brendel et al. introduce an iterative black-box attack method by rejection sampling [16]. *Unlike these methods which construct adversarial examples by applying perturbations directly, our proposed method generates examples from pre-trained GANs, offering a novel paradigm in adversarial example generation.*

There exist several works generating adversarial examples with GANs. Baluja et al. propose an adversarial transformation network that is trained to directly produce adversarial examples [32]. In a similar vein, Xiao et al. develop a conditional GAN where it first generates a perturbation and an adversarial example is constructed by adding the perturbation into the original sample [33]. Song et al. further extended this method by proposing an auxiliary classifier GAN to model the class-conditional distribution of data samples and utilize it to generate samples [34]. However, these approaches require a tailored GAN when constructing adversarial examples. Additionally, extra data is required not only for a substitute model training in the black-box scenario but also for a tailored GAN training, whereas these data have to have the same distribution of the training set of the victim model and may not be easily obtained by the adversary. *In contrast, our proposed work introduces a targeted attack framework applicable in both white-box and black-box scenarios, where off-the-shelf pre-trained GANs can be employed directly. Notably, our methods circumvent the need for training a substitute model even in the black-box scenario, presenting a more streamlined and efficient method to construct adversarial examples.*

2.2 Out-of-distribution detection

The enhancement of machine learning model robustness in open-world settings through out-of-distribution (OOD) detection has been extensively studied [12, 35, 36]. Generally, OOD detection seeks to improve the robustness of models by identifying samples that deviate from the training distribution. Hendrycks et al. establish an initial baseline of OOD detection by utilizing probabilities from softmax distributions. This method is based on the insight: compared to OOD examples, correctly classified examples tend to have a higher prediction probability [21]. Building on this, Liang et al. improve the performance of OOD detection by utilizing temperature scaling and adding small perturbations to the inputs, which results in a larger separation between in-distribution and out-of-distribution samples [28]. Hendrycks et al. further enhance detection performance by leveraging an auxiliary dataset of outliers to train anomaly detectors [29]. Chen et al. propose to carefully select informative outliers from an auxiliary OOD dataset and utilize adversarial training to further enhance detection performance [30]. We note that almost all works evaluate their detection performance on OOD datasets collected from the real world. Although Chen et al. present their detection performance on adversarial examples, these examples are constructed by adding adversarial perturbations [30]. *Our GAN-based attack methods have a different mechanism in generating OOD examples, which may provide supplementary testing means to thoroughly evaluate the performance of OOD detection.*

2.3 Generative adversarial networks

Generative Adversarial Networks (GANs) were first introduced by Goodfellow et al. in 2014, revolutionizing the generation of synthetic data [1]. Since then, numerous GAN models have been proposed to generate increasingly realistic and diverse images [37–42]. The Deep Convolutional GAN (DCGAN) improves image quality through convolutional layers [37]. Martin et al. utilize the Wasserstein distance to stabilize the training process to further improve the quality of images [38]. Karras et al. propose a growing strategy to stabilize the training process, which allows GAN models to produce more realistic and high-quality images [41]. The StyleGAN introduces a novel hierarchical latent style layer, which allows GAN models to produce diverse styles of images [43]. With the emergence of GANs, we are no longer limited to collecting data on our own or utilizing existing datasets when needing images. *Benefiting from this advan-*

tage, in this paper, we aim to directly find adversarial OOD examples from a pre-trained GAN, especially utilizing these state-of-the-art GANs.

3 Method

In this section, we start with necessary preliminaries and notations. Subsequently, we provide an overview of the attack. Finally, we present a detailed description of our unified attack framework.

3.1 Preliminaries and notations

An unconditional GAN consists of a generator G and a discriminator D. During the training process, the generator G learns to synthesize an image $x_g = G(z)$, where a latent code $z \in \mathbb{R}^n$ is drawn from a prior distribution p_z, such as Gaussian distribution. The discriminator D is responsible for distinguishing between fake samples x_g generated by the G and real samples x_r from the training set X. Once the GAN model finishes training, only the generator G is used to produce a novel image through a latent code z, i.e. $x = G(z)$.

We consider an m-class classifier $F(\cdot)$ as the victim model. The classifier takes an image x and returns a full prediction $y = F(x)$. Here, the full prediction $y = F(x)$ refers to logits or probabilities. y_{max} denotes the predicted label.

In this paper, we refer to victim models without any defenses as *raw models*. In addition, if the distribution of examples is different from that of the training set of the victim model, we refer to these examples as *OOD examples*. For instance, an image of a horse is considered as an OOD example for a victim model trained on the cat/dog dataset. *Adversarial OOD examples* refer to OOD examples that are manipulated by an adversary, and they are classified into two types: restricted adversarial OOD examples and unrestricted adversarial OOD examples. *Restricted adversarial OOD examples* refer to OOD adversarial examples that are constructed based on norm-bounded perturbations, while *unrestricted adversarial OOD examples* refer to OOD adversarial examples from a GAN model. When it is clear from the context, we interchangeably use OOD examples, adversarial OOD examples, restricted adversarial OOD examples, and unrestricted adversarial OOD examples.

3.2 Attack overview

We provide a high-level description of our attack, as depicted in Fig. 1. The victim model is trained on an in-distribution dataset and can accurately classify in-distribution samples, such as airplanes and cats, as shown in the left section of Fig. 1. Conversely, images that are completely different from the in-distribution samples, such as human faces and cartoon faces, can be misclassified as specific classes predetermined by the adversary, as depicted on the right side of Fig. 1. Our attack framework provides one method to craft the generated samples that are misclassified as a pre-specified class by the victim model.

3.3 A unified attack framework

Generally, attack goals can be categorized into two classes: untargeted and targeted attacks. An untargeted attack aims to lead to a victim model's misclassification while a targeted attack seeks to change a victim model's prediction to a pre-defined target class. This paper only focuses on targeted attacks against raw models, because any OOD example is regarded as a successful untargeted attack.

Specifically, the objective of our attack is to find a generated sample from a given GAN, which leads this sample to be classified as a pre-defined class by the victim model. Formally, this problem is framed as an optimization problem:

$$z' = \arg\min_{z} \ell(F(G(z)), F(x_{ref})), \tag{1}$$

where $\ell(\cdot)$ denotes the loss function, x_{ref} represents a reference sample from the target class t that the adversary wishes. By minimizing the distance of the victim model's outputs

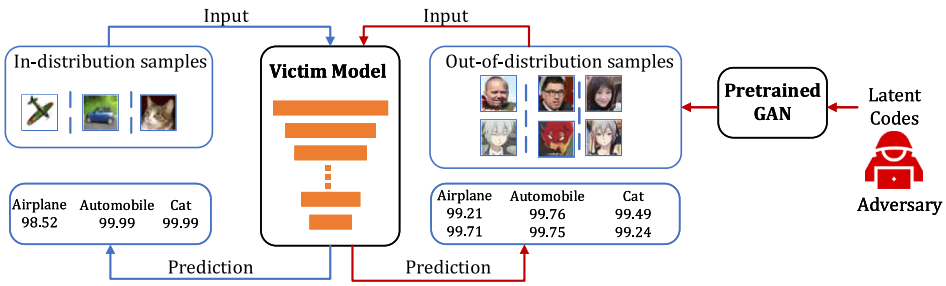


Fig. 1 Attack overview. Our proposed attack constructs out-of-distribution samples by a pre-trained GAN to make a victim model classify them as certain classes that an adversary wishes. For instance, OOD samples, such as human faces or cartoon faces generated from GANs, are recognized as certain classes, such as airplanes, by the victim classifier trained on CIFAR-10

between the generated sample $F(G(z))$ and the reference sample $F(x_{ref})$, (1) aims to find a latent code z' and causes the victim model to classify the generated image $x' = G(z')$ into the pre-specified class t.

Equation (1) is agnostic to the specific type of unconditional GAN. Thus, the generator G of any pre-trained GAN models can be utilized. For our experiments, we choose the cross-entropy loss and the StyleGAN [43] as our loss function and the pre-trained GAN model to attack raw models. Therefore, (1) can be reformulated as:

$$z' = \arg\min_{z} \ell_{CE}(F(G(z)), t) \tag{2}$$

Other types of loss functions can be utilized to further improve attack performance for models with defenses, which is discussed in Section 5.3. We leave the exploration of different GAN models as future work.

We consider two typical attack scenarios: white-box and black-box. For each attack scenario, different optimization methods are proposed to solve the (2).

3.3.1 White-box scenario

In this attack scenario, the adversary has full access to the whole victim model, including its weights and architecture. Therefore, the gradient information can be easily obtained for the adversary. Any gradient-based optimization methods can be used to solve the (1). In this work, we apply the stochastic gradient descent (SGD) [44] for its computational efficacy. We discuss different types of optimizations in Section 5.4.2.

Algorithm 1 Attack procedure,

Input: a victim model F, a target class t, a pretrained GAN G, the maximum number of iterations K, the learning rate α, *flag*
Output: an unrestricted adversarial OOD example x

Initialization: draw a latent code z from the standard normal distribution
for $k = 0, 1, \ldots, K-1$ **do**
  if $F(G(z))_{max} == t$ **then**
    break ▷ Early stopping
  end if
  $\mathcal{L} = computeLoss(F(G(z)), t)$ ▷ (2)
  if *flag is white-box* **then**
    $z \leftarrow z - \alpha \nabla_z \mathcal{L}$ ▷ Stochastic gradient descent
  else if *flag is black-box* **then**
    $z \leftarrow$ Powell($\mathcal{L}$) ▷ Powell optimization
  end if
end for
return $x = G(z)$

3.3.2 Black-box scenario

In this attack scenario, the adversary is only allowed to query the model and has access to the model's outputs. Gradient-based optimization methods cannot be applied due to the unavailability of the whole victim model. Here, we adopt Powell's conjugate direction method (Powell) [45] to optimize (1), because it does not require the gradient of the victim model. Specifically, Powell's method first creates a set of mutually conjugate directions for the latent code z and then finds the local minimum by line search along with these directions.

The attack procedures for both the white-box and black-box scenarios are described in Algorithm 1.

4 Experiments

In this section, we detail the experiments conducted in our work, including the datasets, victim models, evaluation metrics, and experimental settings used to assess the performance of our attack methods.

4.1 Datasets

Our experiments utilize five different datasets. These datasets can be further divided into three types: in-distribution, out-of-distribution, and auxiliary datasets. We refer to a dataset used to train a victim model as an in-distribution dataset, while an out-of-distribution dataset refers that its distribution is different from that of the training set of the victim model. Auxiliary datasets are also out-of-distribution datasets but they are commonly used in OOD detection to improve the robustness of machine learning models.

In-distribution datasets We use CIFAR-10 [46] and the German Traffic Sign Recognition Benchmark (GTSRB-43) [47] as in-distribution datasets for training the victim models. CIFAR-10 has 10 classes and contains 50,000 training images and 10,000 test images, while GTSRB-43 has 43 classes of traffic signs and includes 39,209 training images and 12,630 test images.

Out-of-distribution datasets We use FFHQ [43] and iCartoonFace [48] as out-of-distribution datasets. These datasets are used for GAN training. In addition, they are utilized to make adversarial examples when we compare our work with adversarial example attacks. FFHQ consists of 70,000 high-quality human face images and these images have a large amount of variation in the aspect of age, ethnicity, identity, and accessories. iCartoonFace contains 389,678 cartoon face

images and these images are collected from 1,302 cartoon albums.

Auxiliary datasets The ImageNet dataset [49] is used as the auxiliary dataset. It is widely used in various OOD detection methods to help improve the robustness of ML models. The ImageNet dataset consists of 1,281,167 diverse images and 1,000 classes.

4.2 Victim models

We evaluate our attack methods on two widely adopted model architectures: WideResNet [50] with depth 28 and widening factor of 4 and DenseNet [3] with depth 100 and growth rate 12. These models are chosen for their excellent performance and frequent use in the research community of machine learning security [51, 52].

4.3 Evaluation metrics

To assess the performance of our attack method, We use Attack Success Rate (ASR) as our primary metric. The ASR is the ratio of success samples in all test samples. In our attack, an example is considered a success if it is recognized as a pre-specified class. With the aim to thoroughly evaluate attack methods and reveal the vulnerability of victim models, we further report average ASR, best ASR, and worst ASR from the perspective of classes of victim models.

The average ASR reports the mean of ASRs of all classes. The best ASR refers to the best ASR in all classes while the worst ASR reports the worst ASR in all classes. The best ASR indicates the attack success rate of the most vulnerable category in a victim model, which can be regarded as the most vulnerable point of the model. The worst ASR represents the probability of success for the least vulnerable class in a victim model. The worst ASR of 0 means that there exist classes that are harder to attack in the model. For computing efficiency, in our work, 10 test samples are used to compute the ASR of each class. Thus, we totally use 100 samples and 430 samples for CIFAR-10 and GTSRB-43, respectively. Note that in practice the attack is successful even if only one sample can fool the model.

4.4 Experimental settings

In our work, we use the standard split of CIFAR-10 and GTSRB-43 to train victim models. As for OOD datasets, all samples in FFHQ are used for GAN training. Due to various image sizes in iCartoonFace, images whose size is equal to or larger than 128 are chosen. Consequently, we obtain 35,2459 images in total. Similarly, these images are used for GAN training. Samples that are utilized to make unrestricted adversarial OOD examples are randomly selected from the OOD datasets. All images in all datasets are resized to 32×32 pixels and rescaled to the range [0, 1].

All victim models are trained using SGD optimizer with an initial learning rate of 0.1. For CIFAR-10, the number of training epochs is set to 100 and the learning rate is decayed by a factor of 0.1 at the 50th, 75th, and 90th epochs. For GTSRB-43, we set the number of training epochs to 20 and the learning rate is decayed by a factor of 0.1 at the 7th, 12th, and 17th epochs. For GAN models, StyleGAN is chosen due to its excellent performance on image generation and the suggested hyperparameters in its original publication are used for training [43].

For our attack methods, we utilize SGD with a learning rate of 0.1 and a maximum of 5,000 iterations in the white-box scenario. Early stopping is allowed when OOD examples are found successfully. In the black-box scenario, the maximum query limit is set as 25,000.

5 Results

In this section, we first present our attack results on raw models (see Section 5.1). Next, we compare our methods with eight state-of-the-art adversarial example attacks, as detailed in Section 5.2. We further evaluate our attack performance on models with the protection of three representative defense measures, discussed in Section 5.3. Finally, we investigate diverse factors to better delineate the properties of our attack methods in Section 5.4.

5.1 Attack performance on raw models

5.1.1 Performance of raw models

Table 1 summarizes the performance of raw victim models trained on different datasets. All models achieve close to state-of-the-art performance. For instance, at least 94% accuracy and 96% accuracy can be seen on CIFAR-10 and GTSRB-43, respectively.

5.1.2 Attack results

Table 2 presents attack performance on raw models trained on various datasets across both white-box and black-box

Table 1 Performance of raw models on CIFAR-10 and GTSRB-43

Dataset	Model	Accuracy
CIFAR-10	WideResNet	95.16%
	DenseNet	94.43%
GTSRB-43	WideResNet	96.46%
	DenseNet	96.08%

Table 2 Attack performance on raw models trained on various datasets

In-distribution Dataset	Victim Model	OOD Dataset	White-box Average ASR (SD) %	White-box Best ASR %	White-box Worst ASR %	Black-box Average ASR (SD) %	Black-box Best ASR %	Black-box Worst ASR %
CIFAR-10	WideResNet	FFHQ	100.00 (0.00)	100.00	100.00	90.00 (10.00)	100.00	70.00
		iCartoonFace	100.00 (0.00)	100.00	100.00	97.00 (4.58)	100.00	90.00
	DenseNet	FFHQ	100.00 (0.00)	100.00	100.00	83.00 (17.35)	100.00	50.00
		iCartoonFace	100.00 (0.00)	100.00	100.00	95.00 (6.71)	100.00	80.00
GTSRB-43	WideResNet	FFHQ	100.00 (0.00)	100.00	100.00	91.86 (13.16)	100.00	50.00
		iCartoonFace	99.07 (2.90)	100.00	90.00	89.07 (13.61)	100.00	40.00
	DenseNet	FFHQ	99.77 (1.51)	100.00	90.00	68.60 (27.50)	100.00	10.00
		iCartoonFace	97.44 (5.32)	100.00	80.00	62.09 (27.75)	100.00	10.00

SD: standard deviation

scenarios. Overall, all raw models in both scenarios are vulnerable to our proposed attack methods.

In the white-box scenario, the adversary can achieve an extraordinarily high average ASR among all victim models, ranging from 97.44% to 100.00%. For example, the attack method can achieve an average ASR of 100% on CIFAR-10, no matter which victim models are attacked. Even in terms of the worst ASR, the attack method still remains an attack success rate of 100% for CIFAR-10 and no less than 80% for GTSRB-43. This indicates that a victim model without any protection is easily fooled and our proposed method can always find a sample from the pre-trained GAN model and make the victim model recognized as a pre-specified class.

In the black-box scenario, the attack method can still achieve above 83% average ASR on CIFAR-10 and 62% average ASR on GTSRB-43, while the performance is inferior to that of the white-box scenario. We observe that even in the black-box scenario, 100% best ASR can be seen on all raw models, which indicates that certain classes of victim models are extremely vulnerable. In spite of restrictions of the black-box scenario where the adversary can not have access to the model and only be allowed to obtain output information, the vulnerability of the victim model, i.e. these classes, could not be reinforced. At the same time, we also see that all worst ASR values in the black-box scenario show a decline, compared with that in the white-box scenario. For example, the attack method achieves 10% worst ASR on the victim model DenseNet trained on GTSRB-43. This might be because classes more resistant to attack in the white-box scenario become even harder to compromise when the adversary's knowledge is restricted.

An important aspect of our attack methods is the generation of OOD examples using a pre-trained GAN model. These OOD examples differ significantly from the training data of the victim models. The underlying principle of our attack leverages the fact that machine learning models often fail to recognize when they encounter unfamiliar inputs. The success of our attack methods in white-box and black-box scenarios reminds model providers that when deploying an ML model in the open world, it is necessary to guarantee the legitimacy of each input. Otherwise, the ML model could make unexpected predictions for illegal inputs and this characteristic can be abused by an adversary to mount a novel security attack.

5.2 Comparison with adversarial example attacks

5.2.1 Overview of adversarial example attacks

We compare the proposed methods against eight state-of-the-art adversarial example attacks in white-box and black-box scenarios. The selected attacks represent a diverse range of strategies and have been widely adopted in adversarial robustness research.

In the white-box scenario, we choose C&W [23], PGD [15], AutoPGD [24] and CGA [25], all of which necessitate access to the gradient information of a victim model.

C&W [23] This method minimizes perturbation by formulating the adversarial example generation as an unconstrained optimization problem, utilizing novel objective functions to deceive machine learning models.

PGD [15] This method starts by adding a small random noise to the input. Then, in each iteration the input is slightly adjusted toward the direction that maximizes the loss function to increase the likelihood of misclassification. After each change, the input is projected back onto the epsilon-ball of allowable perturbations, ensuring that the adversarial example remains close to the original input in Euclidean distance.

AutoPGD [24] This method enhances PGD by automatically adjusting parameters, such as step sizes and iteration counts,

which overcome the limitations of PGD's fixed parameters. Additionally, it introduces flexibility in selecting loss functions.

CGA [25] This method proposes to consider both the gradient direction and the conjugate gradient direction for generating adversarial examples. This also allows the CGA method to diversify the search directions during optimization, potentially exploring more effective perturbations to fool the models.

In the black-box scenario, we consider ZOO [26], DBA [16], HSJA [17] and SignOPT [27], which generate adversarial examples based solely on model outputs.

ZOO [26] This method generates adversarial examples through zeroth order optimization, which approximates the gradient of a victim model. It employs zeroth order stochastic coordinate descent, supplemented by dimension reduction, hierarchical attacks, and importance sampling techniques to enhance the attack's efficiency.

DBA [16] This method utilizes a decision-based method to generate adversarial examples. Specifically, it initiates with a significant adversarial perturbation and progressively minimizes this perturbation while maintaining its adversarial nature.

HSJA [17] Building on the decision-based method, HSJA employs a Monte Carlo method to estimate the gradient direction at the boundary between adversarial and non-adversarial examples, based on binary search results from model queries.

SignOPT [27] This method directly estimates the sign of the gradient of an attack's objective function, rather than the gradient itself, enabling the generation of adversarial examples under constrained information scenarios.

We implement all eight algorithms by the open-source library — Adversarial Robustness Toolbox [53] and the suggested hyperparameters are used. WideResNet is chosen as the architecture of victim models. All restricted adversarial OOD examples are constructed under the ℓ_∞ distance. Following the tradition of prior work [17], we set the maximum perturbation as 8/255. An example is considered a success if it is recognized as a pre-specified class and the magnitude of perturbation does not exceed the maximum perturbation. In addition, the maximum number of queries in the black-box scenario is set as 25,000 per image for all attack methods.

Note that while there is a work studying OOD attack [22], this work only applies the PGD method to OOD datasets. Therefore, we do not explicitly compare this work and the attack method of this work can be considered as equivalent to the PGD method. In addition, white-box methods, such as C&W and PGD, are typically unsuitable for the black-box scenario because these methods are required to train a substitute model which requires extra training data whose distribution has to be similar to that of the training set of the victim model. Instead, for the black-box scenario, we choose ZOO, DBA, HSJA and SignOPT, which all show promising attack performance without the need for training a substitute model. We highlight that our proposed framework is versatile and applicable across both white-box and black-box scenarios without requiring additional data.

5.2.2 Comparative results

Table 3 and Fig. 2 show a comprehensive comparison of our proposed methods against the eight state-of-the-art adversarial example attacks in both white-box and black-box scenarios. Overall, our methods achieve comparable or superior attack performance across these scenarios.

In the white-box scenario, our method can achieve a 100% average ASR on models trained on the CIFAR-10 dataset. Similar performance is also observed with the PGD, AutoPGD, and CGA methods on CIFAR-10, though the C&W method exhibits a slightly inferior attack performance. For models trained on the GTSRB-43 dataset, our method maintains a 100% average ASR using the OOD dataset FFHQ and achieves 99.07% average ASR using the OOD dataset

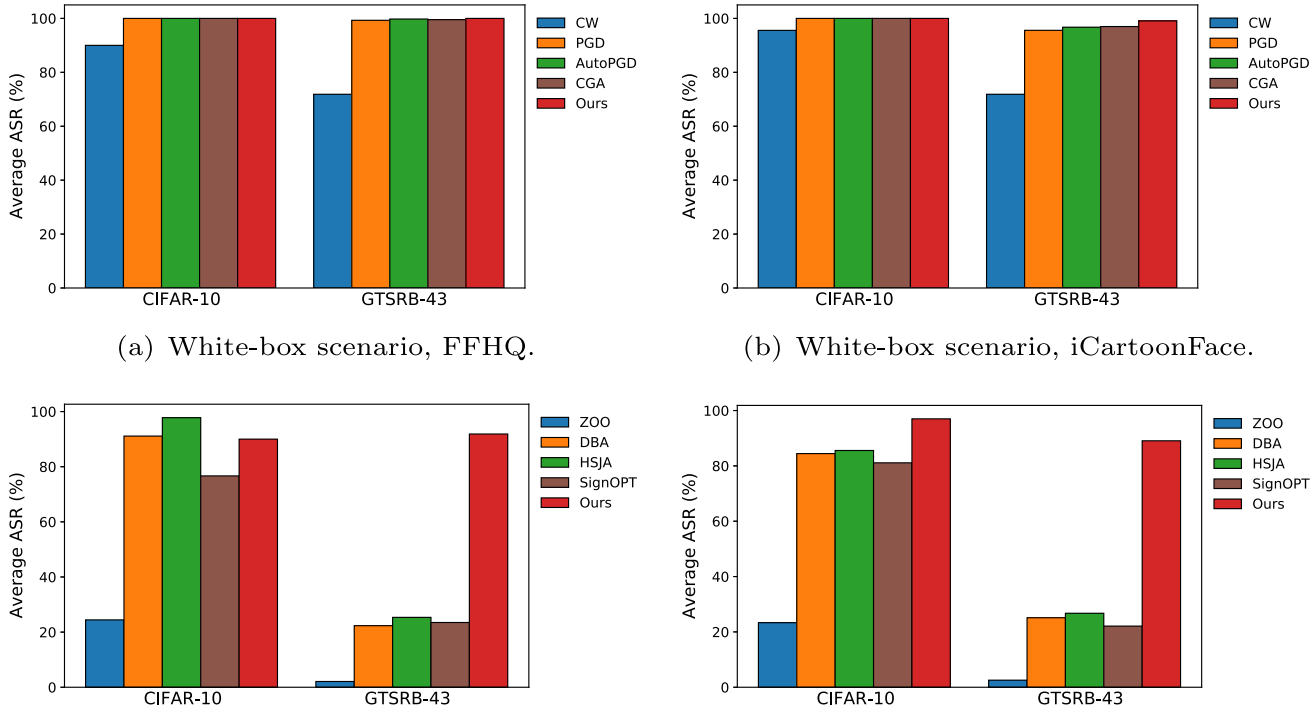
Table 3 Comparison with different attack methods on the white-box and black-box scenarios

Scenarios	Method	OOD Dataset	CIFAR-10			GTSRB-43		
			Average ASR (SD) %	Best ASR %	Worst ASR %	Average ASR (SD) %	Best ASR %	Worst ASR %
White-box	CW [23]	FFHQ	90.00 (10.48)	100.00	66.67	71.86 (21.05)	100.00	20.00
		iCartoonFace	95.56 (7.37)	100.00	77.78	71.86 (16.32)	100.00	30.00
	PGD [15]	FFHQ	100.00 (0.00)	100.00	100.00	99.30 (3.34)	100.00	80.00
		iCartoonFace	100.00 (0.00)	100.00	100.00	95.58 (6.58)	100.00	70.00
	AutoPGD [24]	*FFHQ*	*100.00 (0.00)*	*100.00*	*100.00*	*99.77 (1.50)*	*100.00*	*90.00*
		iCartoonFace	100.00 (0.00)	100.00	100.00	96.74 (5.59)	100.00	80.00

Table 3 continued

Scenarios	Method	OOD Dataset	CIFAR-10 Average ASR (SD) %	CIFAR-10 Best ASR %	CIFAR-10 Worst ASR %	GTSRB-43 Average ASR (SD) %	GTSRB-43 Best ASR %	GTSRB-43 Worst ASR %
	CGA [25]	FFHQ	100.00 (0.00)	100.00	100.00	99.53 (2.11)	100.00	90.00
		iCartoonFace	*100.00 (0.00)*	*100.00*	*100.00*	*96.98 (4.59)*	*100.00*	*90.00*
	Ours	**FFHQ**	**100.00 (0.00)**	**100.00**	**100.00**	**100.00 (0.00)**	**100.00**	**100.00**
		iCartoonFace	**100.00 (0.00)**	**100.00**	**100.00**	**99.07 (2.90)**	**100.00**	**90.00**
Black-box	ZOO [26]	FFHQ	24.44 (28.89)	88.89	0.00	2.09(4.60)	20.00	0.00
		iCartoonFace	23.33 (21.34)	55.56	0.00	2.56(20.00)	20.00	0.00
	DBA [16]	FFHQ	91.11 (10.89)	100.00	66.67	22.33 (20.78)	80.00	0.00
		iCartoonFace	84.44 (14.23)	100.00	55.56	25.12(18.60)	70.00	0.00
	HSJA [17]	*FFHQ*	**97.78 (4.44)**	**100.00**	**88.89**	*25.35(19.09)*	*80.00*	*0.00*
		iCartoonFace	*85.56 (12.22)*	*100.00*	*55.56*	*26.74(16.24)*	*80.00*	*0.00*
	SignOPT [27]	FFHQ	76.67 (9.23)	88.89	55.56	23.49 (22.61)	90.00	0.00
		iCartoonFace	81.11 (12.22)	100.00	55.56	22.10 (17.46)	70.00	0.00
	Ours	**FFHQ**	*90.00 (10.00)*	*100.00*	*70.00*	**91.86 (13.16)**	**100.00**	**50.00**
		iCartoonFace	**97.00 (4.58)**	**100.00**	**90.00**	**89.07 (13.61)**	**100.00**	**40.00**

The highest results are highlighted in bold and the second highest in italics



(a) White-box scenario, FFHQ. (b) White-box scenario, iCartoonFace.

(c) Black-box scenario, FFHQ. (d) Black-box scenario, iCartoonFace.

Fig. 2 Comparison with different attack methods on the white-box and black-box scenarios

iCartoonFace. In contrast, the compared methods do not exhibit superior performance, where the average ASR ranges from 71.86% to 99.77% using the OOD dataset FFHQ and from 71.86% to 96.98% the OOD dataset iCartoonFace.

The possible reason why our method shows better performance is that due to the significant difference between in-distribution samples and out-of-distribution samples, ℓ-norm-based adversarial example attacks indeed require much larger magnitudes of perturbations. Moreover, while all attack methods show perfect performance with respect to best ASR, the attack performance presents significant differences in the aspect of worst ASR. For instance, the worst ASR on GTSRB-43 is 90.00% for our attack methods, whereas it is only 70.00% and 20.00% for the PGD method and the C&W method, respectively.

In the black-box scenario, our method consistently shows superior attack performance on both datasets. For example, ours can achieve over 90.00% average ASR on CIFAR-10 and 89.07% average ASR on GTSRB-43, while the method ZOO only gains no more than 24.44% and 2.56% average ASR on CIFAR-10 and GTSRB-43, respectively. Although DBA and HSJA can obtain similar attack performance in comparison with our method on CIFAR-10, the significant advantages of our method can be seen on GTSRB-43. For instance, the average ASR of our method on GTSRB-43 is at least higher 62% than that of DBA and HSJA where the best performance is at most 26.74% average ASR. This also reminds model owners that relying solely on current adversarial example attack methods may not provide a comprehensive assessment of a model's resilience in the open world.

5.3 Attack performance on models with defenses

In this section, we examine the effectiveness of our proposed attack methods in evading state-of-the-art defense mechanisms designed to protect machine learning models.

5.3.1 Overview of defenses

Existing works mainly focus on the detection of OOD samples to improve the robustness of models in the open world. To do this, these methods first compute a score for each sample based on the outputs of the model. The score can be regarded as the probability that this sample is an in-distribution example. Then, a sample is an in-distribution example if its score is larger than a pre-defined threshold τ. Specifically, given a victim model F, an OOD detector can be represented as:

$$Detector(x) = \begin{cases} 1 & F(x) > \tau \\ 0 & F(x) \leq \tau \end{cases} \tag{3}$$

Depending on whether or not the victim model F needs to be changed, OOD methods can be divided into two categories. The first category does not require any changes of the victim model F. In other words, this type of defense can be directly applied to a model without retraining and it detects OOD examples based on the confidence scores of models. We choose one classic and widely-used defense method ODIN [28], considering its excellent performance in this category. The second category requires changes of models, such as retraining models with novel loss functions, adding auxiliary datasets into the training process, and even introducing adversarial training. One classic defense method OE [29] and the latest state-of-the-art defense method ATOM [30] are considered in this work. A brief introduction of each defense is given as follows.

ODIN This defense [28] utilizes the model's outputs, i.e. confidence scores, to detect OOD examples. The main assumption is that neural networks tend to output higher confidence scores to in-distribution examples than OOD examples. The defense utilizes temperature scaling [54] and adversarial perturbations [31] to further enlarge the differences between in-distribution and OOD examples. In detail, for a given example x, a perturbed example $\tilde{x}$ is obtained by adding adversarial perturbations and the calibrated confidence score of $\tilde{x}$ can be calculated by temperature scaling. An example belongs to in-distribution if the score is greater than a pre-defined threshold τ. This defense can be considered as a post-processing technique and does not require model owners to retrain victim models.

In our experiments, we utilize the ImageNet dataset to choose hyperparameters. For WideResNet, we set the perturbation magnitude as 0 for both CIFAR-10 and GTSRB-43. For DenseNet, the perturbation magnitude is set as 0.0004 for CIFAR-10 and 0.001 for GTSRB-43, respectively. The temperature scaling is set as 1000 in all settings.

OE This defense [29] makes use of auxiliary OOD datasets to train anomaly detectors that can generalize and detect unseen OOD examples. It introduces a new loss function that can not only learn the original classification objective but also learn heuristics to detect whether a sample is an OOD example by the auxiliary OOD dataset. This approach requires retraining a model.

ATOM This defense [30] combines adversarial training and an auxiliary OOD dataset to collaboratively enhance the performance of OOD detection. More specifically, this approach first selects informative outliers from an auxiliary OOD dataset, and then these selected OOD samples and in-distribution samples are utilized to adversarially train the model. The projected gradient descent method [15] is adopted for adversarial training. This approach also requires retraining a model.

For all these defense methods, we follow the convention of the research community in OOD detection and the threshold τ is chosen when the true-positive rate (TPR) is 95% where the TPR refers to the ratio of in-distribution examples correctly classified as in-distribution examples. For OE and ATOM, the ImageNet dataset is used as the auxiliary OOD dataset for training. We train the models with the SGD optimizer and the suggested hyperparameters of their papers are used.

All hyperparameters of our attack methods on defense models are the same as those on raw models, except for loss functions. Specifically, we use loss function $f4$ for OE and cross-entropy loss function $f3$ for the victim model DenseNet trained on GTSRB-43 in the black-box scenario and ℓ_2 loss function $f2$ is used for the remaining victim models. The reason is that these loss functions show better attack performance and it is hard to achieve the best performance only by one loss function for various defense measures. We illustrate different types of loss functions in Section 5.4.1. Again, we only consider the targeted attack for models with defenses, because a successful targeted attack also indicates a successful untargeted attack. Note that the ASR in defenses has a higher requirement: one successful attack means that one sample needs to first evade the OOD detection, and is recognized as a pre-specified class that the adversary wishes.

5.3.2 Defense performance of models

Table 4 shows the performance of various defense measures. Here, the false-positive rate (FPR) refers to the ratio of OOD samples that are predicted as in-distribution samples. A lower FPR value indicates better detection performance. Accuracy refers to the ratio of samples that is predicted as an in-distribution sample and is recognized as a correct class. We can see that almost all models show outstanding detection performance and prediction performance, although the method ODIN on CIFAR-10 shows poor OOD detection performance.

5.3.3 Attack results

Table 5 summarizes the attack performance of our proposed methods under different defense measures. Overall, all defenses cannot prevent our attacks on the white-box and black-box scenarios, although these defenses can lower our attack success rate to some degree. The defense measures mainly concentrate on decreasing the worst ASR of our methods, while the best ASR is hardly reduced. The attack performance in the white-box scenario generally is better than that in the black-box scenario.

For the defense method ODIN, the average ASR in the white-box scenario exceeds 86.00% on CIFAR-10 and 69.38% on GTSRB-43. In contrast, the attack performance in the black-box scenario shows a decrease where the average ASR is more than 40.00% on CIFAR-10 and 9.53% on GTSRB-43. With regard to the best ASR, our attack performance remains 100.00% in the white-box scenario and more than 60.00% in the black-box scenario. We observe that the worst ASR value of 0.00% can be seen on several victim models in the black-box scenario, indicating that the ODIN defense indeed improves the robustness of certain victim models' classes to some extent.

For the defense method OE, the average ASR in the white-box scenario varies from 13.00% to 56.00% on CIFAR-10 and from 23.26% to 73.26% on GTSRB-43. In contrast, in the black-box scenario, the average ASR is above 35.00% on CIFAR-10 and 1.16% on GTSRB-43. We can also see that

Table 4 Performance of victim models with defense measures

Method	Dataset	Model	FPR % ↓	Accuracy % ↑
ODIN	CIFAR-10	WideResNet	54.978	92.10
		DenseNet	46.024	91.12
	GTSRB-43	WideResNet	8.502	93.11
		DenseNet	11.400	92.86
OE	CIFAR-10	WideResNet	0.110	91.99
		DenseNet	0.226	91.38
	GTSRB-43	WideResNet	0.002	93.71
		DenseNet	0.014	93.23
ATOM	CIFAR-10	WideResNet	0.014	90.16
		DenseNet	0.038	89.05
	GTSRB-43	WideResNet	0.016	92.34
		DenseNet	0.096	91.43

↓ means smaller is better while ↑ means larger is better

Table 5 Attack performance on various defense methods

Defense Method	In-distribution Dataset	Victim Model	OOD Dataset	White-box			Black-box		
				Average ASR (SD) %	Best ASR %	Worst ASR %	Average ASR (SD) %	Best ASR %	Worst ASR %
ODIN	CIFAR-10	WideResNet	FFHQ	96.00 (4.90)	100.00	90.00	45.00 (38.28)	100.00	0.00
			iCartoonFace	87.00 (14.87)	100.00	60.00	59.00 (24.27)	90.00	20.00
		DenseNet	FFHQ	93.00 (12.69)	100.00	60.00	40.00 (38.73)	100.00	0.00
			iCartoonFace	86.00 (12.81)	100.00	60.00	50.00 (20.00)	90.00	20.00
	GTSRB-43	WideResNet	FFHQ	93.72 (8.08)	100.00	70.00	37.21 (31.28)	100.00	0.00
			iCartoonFace	75.12 (18.97)	100.00	20.00	31.40 (25.30)	90.00	0.00
		DenseNet	FFHQ	87.91 (17.33)	100.00	30.00	13.72 (23.13)	80.00	0.00
			iCartoonFace	69.38 (22.11)	100.00	10.00	9.53 (13.63)	60.00	0.00
OE	CIFAR-10	WideResNet	FFHQ	44.00 (15.62)	60.00	10.00	35.00 (29.41)	100.00	0.00
			iCartoonFace	13.00 (11.87)	30.00	0.00	63.00 (19.52)	90.00	40.00
		DenseNet	FFHQ	56.00 (20.10)	90.00	20.00	44.00 (28.00)	100.00	10.00
			iCartoonFace	41.00 (15.78)	70.00	20.00	75.00 (17.46)	100.00	40.00
	GTSRB-43	WideResNet	FFHQ	73.26 (17.75)	100.00	30.00	7.44 (14.16)	70.00	0.00
			iCartoonFace	42.09 (13.90)	70.00	10.00	5.81 (11.05)	40.00	0.00
		DenseNet	FFHQ	53.64 (21.52)	100.00	10.00	4.19 (12.24)	70.00	0.00
			iCartoonFace	23.26 (18.26)	60.00	0.00	1.16 (3.21)	10.00	0.00
ATOM	CIFAR-10	WideResNet	FFHQ	86.00 (8.00)	100.00	70.00	23.00 (24.52)	80.00	0.00
			iCartoonFace	56.00 (14.97)	80.00	40.00	34.00 (22.00)	80.00	0.00
		DenseNet	FFHQ	78.00 (15.36)	100.00	50.00	11.00 (14.46)	50.00	0.00
			iCartoonFace	58.00 (15.36)	90.00	40.00	18.00 (13.27)	50.00	0.00
	GTSRB-43	WideResNet	FFHQ	59.30 (26.80)	100.00	10.00	0.93 (3.61)	20.00	0.00
			iCartoonFace	54.65 (21.50)	90.00	10.00	1.40 (5.10)	30.00	0.00
		DenseNet	FFHQ	62.79 (23.06)	100.00	0.00	0.23 (1.51)	10.00	0.00
			iCartoonFace	56.98 (22.26)	100.00	0.00	2.56 (4.87)	20.00	0.00

SD refers to standard deviation

the best ASR is still high in both scenarios. Compared with the white-box scenario, the worst ASR has a lower value in the black-box scenario. We also observe that different GAN models have effects on attack performance. For instance, the GAN trained on FFHQ can achieve better performance than that on iCartoonFace in the white-box scenario.

For the defense method ATOM, our attack methods can achieve at least an average ASR of 54% for all victim models in the white-box scenario. As a comparison, the average ASR fluctuates from 11.00% to 34.00% on CIFAR-10 and from 0.23% to 2.56% on GTSRB-43. Although the worst ASR value is low, especially in the black-box scenario, our attack performance shows promising performance with respect of the best ASR. That is, some vulnerable classes of the victim model are still easily fooled. We should emphasize that usually the robustness level of a model largely depends on the most vulnerable points.

Table 6 presents the attack performance on defense methods using the LSUN church OOD dataset [55]. This dataset consists of images of church buildings and is utilized by the original state-of-the-art defense methods for evaluating OOD detection performance. Here, we employ the same dataset to conduct attacks, thereby neutralizing any potential advantage of our method. The victim model is the WideResNet model trained on CIFAR-10. Despite the challenges, our method demonstrates robust attack capabilities against these defense methods. For instance, in the case of the ODIN defense method, our method achieves average ASRs of 81.00% in the white-box scenario and 44.00% in the black-box scenario.

In summary, our attack methods successfully evade these defenses and achieve targeted attacks in both white-box and black-box scenarios. Although a majority of worst ASR values is low and even 0, the fact that these defense measures can not decrease the best ASR indicates that the real threats

Table 6 Attack performance on defense methods using the OOD dataset LSUN church

Defense Method	White-box			Black-box		
	Average ASR(SD) %	Best ASR%	Worst ASR%	Average ASR(SD) %	Best ASR%	Worst ASR%
ODIN	81.00 (15.78)	100.00	60.00	44.00 (28.00)	90.00	0.00
OE	27.00 (25.32)	90.00	0.00	62.00 (28.57)	100.00	0.00
ATOM	50.00 (16.12)	90.00	30.00	35.00 (27.66)	80.00	0.00

The victim model is the WideResNet model trained on CIFAR-10. SD: standard deviation

of a model can not be alleviated. Our work also underscores the importance of focusing on the most vulnerable classes within a model when evaluating the effectiveness of defense measures.

5.4 Analysis of attack performance

In this section, we analyze attack performance in terms of different types of loss functions (see Section 5.4.1) and optimization methods (see Section 5.4.2). We also depict the attack process of our attack methods in both scenarios in Section 5.4.3. For our analysis, we fix the victim models as the WideResNet model trained on CIFAR-10 and choose StyleGAN trained on FFHQ.

5.4.1 Effects of loss functions

Loss functions play an important role in generating adversarial OOD examples. As shown in (1), our attack framework constructs an adversarial OOD example by minimizing a loss function. There are many different types of loss functions and in this work, we study the following loss function f:

$$\text{f1} = |F(G(z)) - F(x_{ref})| \tag{4}$$

$$\text{f2} = (F(G(z)) - F(x_{ref}))^2 \tag{5}$$

$$\text{f3} = CrossEntropy(F(G(z)), (F(x_{ref}))_{max}) \tag{6}$$

$$\text{f4} = \text{f2} + \lambda \cdot (-\max(softmax(F(G(z))))) \tag{7}$$

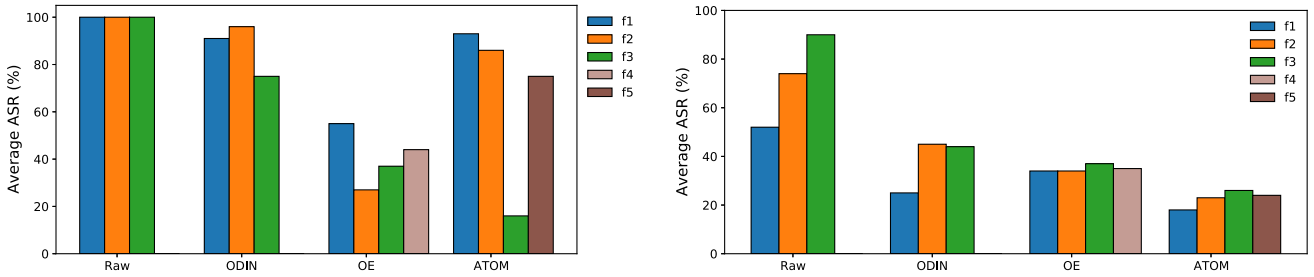
$$\text{f5} = \text{f2} + \lambda \cdot (softmax(F(G(z)))_{m+1}) \tag{8}$$

λ is a hyperparameter. f1, f2, f3 are ℓ_1 loss, ℓ_2 loss, and cross-entropy loss function, respectively. They are common and basic loss functions. Loss function f4 is designed for OE defense measures but can be widely applied to attack these defense techniques that detect OOD samples based on higher output scores. It adds a new item that aims to maximize the output scores of a sample, which makes the attack evade detection more efficiently. Loss function f5 is designed for ATOM defense measure and it adds a new item that aims to minimize the output scores of the OOD class of a sample. Similarly, this loss function can be applied to attack these defense techniques that add a new OOD class $m + 1$ in ML

Table 7 Attack performance in terms of different types of loss functions

Method	Loss function	White-box			Black-box		
		Average ASR(SD) %	Best ASR%	Worst ASR	Average ASR(SD) %	Best ASR%	Worst ASR%
Raw	f1	100.00 (0.00)	100.00	100.00	52.00 (34.58)	100.00	0.00
	f2	100.00 (0.00)	100.00	100.00	74.00 (26.15)	100.00	20.00
	f3	100.00 (0.00)	100.00	100.00	90.00 (10.00)	100.00	70.00
ODIN	f1	91.00 (11.36)	100.00	70.00	25.00 (35.28)	100.00	0.00
	f2	96.00 (4.90)	100.00	90.00	45.00 (38.28)	100.00	0.00
	f3	75.00 (18.03)	100.00	50.00	44.00 (33.23)	100.00	0.00
OE	f1	55.00 (23.35)	90.00	20.00	34.00 (31.69)	100.00	0.00
	f2	27.00 (15.52)	60.00	10.00	34.00 (30.40)	100.00	0.00
	f3	37.00 (30.02)	100.00	0.00	37.00 (27.95)	100.00	10.00
	f4	44.00 (15.62)	60.00	10.00	35.00 (29.41)	100.00	0.00
ATOM	f1	93.00 (11.87)	100.00	60.00	18.00 (19.90)	60.00	0.00
	f2	86.00 (8.00)	100.00	70.00	23.00 (24.52)	80.00	0.00
	f3	16.00 (14.29)	40.00	0.00	26.00 (26.53)	80.00	0.00
	f5	75.00 (12.04)	90.00	60.00	24.00 (25.77)	90.00	0.00

SD: standard deviation



(a) White-box scenario (b) Black-box scenario

Fig. 3 Attack performance on various models in terms of different types of loss functions

models besides the normal number of classes m. The new class $m+1$ of this type of defense is specially used for OOD detection and a higher output score of this class means a higher OOD probability.

Results Table 7 and Fig. 3 present attack performance across different loss functions in both scenarios. Overall, we can see that different loss functions indeed have different attack performances. For example, loss functions are hard to show the difference when attacking raw models in the white-box scenario. In contrast, the loss function f3 excels in raw models in the black-box scenario. When mounting attacks against victim models with defenses, it becomes somewhat more difficult to choose a loss function that is applied for all defenses, because the attack performance of these loss functions fluctuates. However, loss functions f1, f2, and f3 can be regarded as good starting points for an attack. We also observe that f4 and f5 loss functions also show a good performance, comprehensively considering both attack scenarios.

5.4.2 Effects of optimization methods

We study two types of optimization methods: SGD [44] and Adam [56]. Both optimization methods are used in the white-box scenario because they require computing gradients. We do not consider other optimization methods in the black-box scenario in addition to the Powell method. This is because existing black-box optimization methods are hard to be applied to the current attack scenario and a new black-box optimization needs to be proposed. We leave it for further work.

Results Table 8 and Fig. 4 show attack performance on different types of optimization methods. We can observe that the Adam optimizer shows better attack performance on victim models with defense measures, compared with the SGD optimizer. For example, the Adam optimizer can increase the average ASR by 22.00% on OE and the attack performance on ATOM can be improved to 99.00%. For raw models, both optimizers achieve amazing performance and do not show a difference.

Table 8 Attack performance on various models in terms of different types of optimizers

Method	Optimizer	Average ASR (SD) %	Best ASR %	Worst ASR %
Raw	SGD	100.00 (0.00)	100.00	100.00
	Adam	100.00 (0.00)	100.00	100.00
ODIN	SGD	96.00 (4.90)	100.00	90.00
	Adam	99.00 (3.00)	100.00	90.00
OE	SGD	27.00 (15.52)	60.00	10.00
	Adam	49.00 (25.87)	90.00	10.00
ATOM	SGD	86.00 (8.00)	100.00	70.00
	Adam	99.00 (3.00)	100.00	90.00

SD refers to standard deviation

5.4.3 Optimization processes

To further understand the dynamics of the attack process, we analyze the optimization trajectories during the attack on a raw model trained on CIFAR-10. In this experiment, cross-entropy loss is used in both scenarios. SGD optimizer and Powell optimizer are used in the white-box and black-box scenarios, respectively. Here, the target label that the adversary wishes is labeled 0.

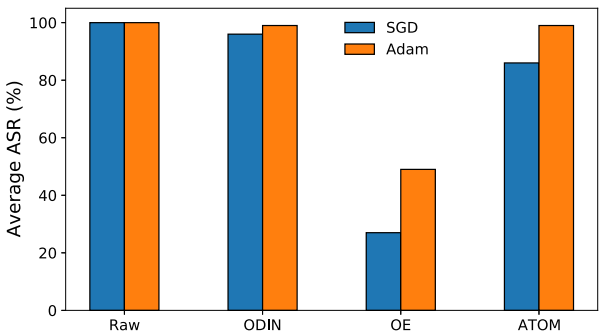
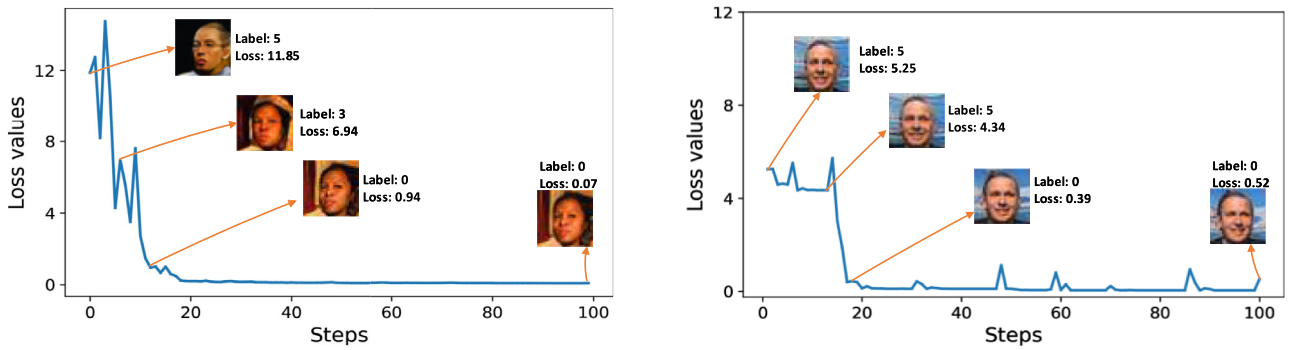


Fig. 4 Attack performance on various models in terms of different types of optimizers



(a) SGD optimization in the white-box scenario. (b) Powell optimization in the black-box scenario.

Fig. 5 The optimization process of our attacks on both scenarios. The target label is 0

Figure 5 visualizes the optimization processes for both white-box and black-box scenarios, demonstrating how they gradually minimize the loss function and eventually converge to adversarial examples that fulfill the attack goals. Specifically, as the loss values decrease, We can observe that attack methods in the white-box and black-box scenarios can succeed in finding target samples.

6 Discussion

6.1 Implications

Our proposed attack methods leverage off-the-shelf pre-trained GAN models to craft adversarial OOD examples that can fool victim classification models. Essentially, our proposed framework establishes a mapping to identify the vulnerable points of a victim classification model where adversarial OOD examples can be found in a GAN model. This fundamentally differs from perturbation-based adversarial example attacks. Although traditional adversarial example attacks can find adversarial OOD examples, these examples are discrete and isolated. In contrast, adversarial OOD examples constructed by our methods are generated from a GAN model.

In the era of generative ML, obtaining well-trained GANs from open-source platforms has become increasingly accessible. Our methods can seamlessly leverage these GAN models. As a result, it is feasible for technology companies to integrate our techniques into their testing processes, enhancing secure software development. By exposing significant vulnerabilities in ML models from an attacker's perspective, we aim to foster the development of more resilient defense strategies and promote responsible usage. Furthermore, we believe that powerful attacks can drive the advancement of defensive methods, ultimately enhancing system robustness.

6.2 Limitations

While our novel attack methods can fool ML models and circumvent state-of-the-art OOD detection, it is still possible to mitigate our attacks. For instance, as illustrated in Section 5.3, one possible efficient method is to design OOD detection that focuses on the most vulnerable classes of a victim model. This approach might include training detection models specifically on GAN-generated examples to bolster robustness.

7 Conclusion

Real-world ML models face various input examples, including OOD examples that have a different distribution from the training set. In this paper, we have proposed a novel attack that causes ML models to misclassify an OOD example as a pre-specified class that the adversary wishes. By leveraging an off-the-shelf pre-trained GAN model, our framework attempts to craft the OOD example by minimizing the distance between the victim model's output and the pre-specified class. Based on different attack scenarios: white-box and black-box, different attack methods are proposed. We conduct extensive experiments on different victim models on different datasets. Our experimental results show that our attack methods achieve comparable performance on both scenarios, compared with eight adversarial example attacks. Moreover, our evaluation also demonstrates that even for victim models deploying defense mechanisms, our attack methods can still achieve competitive performance on the white-box and black-box scenarios.

Our proposed attack methods can utilize off-the-shelf unconditional GANs. Besides, our methods can also craft OOD examples in a unified framework, which can be applied to both the white-box and black-box scenarios. More

importantly, our methods have a different attack mechanism from existing attack methods. Therefore, it is possible to consider our methods as a supplementary test tool to evaluate the robustness of real-world ML models. In future, we plan to develop defense measures to mitigate our attack. In addition, it is a promising direction to design a more powerful attack aiming to reduce the number of queries in the black-box scenario.

Acknowledgements This research was funded in whole by the Luxembourg National Research Fund (FNR), grant reference 13550291

Author Contributions Hailong Hu: Conceptualization, Methodology, Visualization, Writing - original draft. Jun Pang: Conceptualization, Methodology, Writing - review and editing, Resources.

Data availability and access The authors declare that the datasets utilized in this study were derived from public domain resources. They are available within the article.

Declarations

Ethical and informed consent for data used This study did not involve any human or animal experimentation. Therefore, there are no ethical or informed consent concerns regarding the use of the data.

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

References

1. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Proceedings of annual conference on Neural Information Processing Systems (NeurIPS). Curran Associates, Inc., pp 2672–2680
2. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 770–778
3. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 4700–4708
4. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), pp 4171–4186
5. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Proceedings of annual conference on Neural Information Processing Systems (NeurIPS). Curran Associates, Inc., pp 1877–1901
6. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36(4):1234–1240
7. Demontis A, Melis M, Pintor M, Jagielski M, Biggio B, Oprea A, Nita-Rotaru C, Roli F (2019) Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In: Proceedings of USENIX security symposium (USENIX Security). USENIX Association, pp 321–338
8. Cao Y, Xiao C, Cyr B, Zhou Y, Park W, Rampazzi S, Chen QA, Fu K, Mao ZM (2019) Adversarial sensor attack on lidar-based perception in autonomous driving. In: Proceedings of ACM SIGSAC conference on Computer and Communications Security (CCS). ACM, pp 2267–2281
9. Ji Y, Zhang X, Ji S, Luo X, Wang T (2018) Model-reuse attacks on deep learning systems. In: Proceedings of ACM SIGSAC conference on Computer and Communications Security (CCS). ACM, pp 349–363
10. Pang R, Shen H, Zhang X, Ji S, Vorobeychik Y, Luo X, Liu A, Wang T (2020) A tale of evil twins: adversarial inputs versus poisoned models. In: Proceedings of ACM SIGSAC conference on Computer and Communications Security (CCS). ACM, pp 85–99
11. Pei K, Cao Y, Yang J, Jana S (2017) Deepxplore: automated whitebox testing of deep learning systems. In: Proceedings of Symposium on Operating Systems Principles (SOSP). ACM, pp 1–18
12. Fang Z, Li Y, Lu J, Dong J, Han B, Liu F (2022) Is out-of-distribution detection learnable? In: Proceedings of annual conference on Neural Information Processing Systems (NeurIPS). Curran Associates, Inc
13. Biggio B, Corona I, Maiorca D, Nelson B, Šrndić N, Laskov P, Giacinto G, Roli F (2013) Evasion attacks against machine learning at test time. In: Proceedings of joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD). Springer, pp 387–402
14. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R (2014) Intriguing properties of neural networks. In: Proceedings of International Conference on Learning Representations (ICLR)
15. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2018) Towards deep learning models resistant to adversarial attacks. In: Proceedings of International Conference on Learning Representations (ICLR)
16. Brendel W, Rauber J, Bethge M (2018) Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: Proceedings of International Conference on Learning Representations (ICLR)
17. Chen J, Jordan MI, Wainwright MJ (2020) Hopskipjumpattack: a query-efficient decision-based attack. In: Proceedings of IEEE symposium on Security and Privacy (SP). IEEE, pp 1277–1294
18. Nguyen A, Yosinski J, Clune J (2015) Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 427–436
19. Hein M, Andriushchenko M, Bitterwolf J (2019) Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In: Proceedings of IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 41–50
20. Meinke A, Hein M (2020) Towards neural networks that provably know when they don't know. In: Proceedings of International Conference on Learning Representations (ICLR)
21. Hendrycks D, Gimpel K (2017) A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: Proceedings of International Conference on Learning Representations (ICLR)

22. Sehwag V, Bhagoji AN, Song L, Sitawarin C, Cullina D, Chiang M, Mittal P (2019) Analyzing the robustness of open-world machine learning. In: Proceedings of ACM workshop on artificial intelligence and security. ACM, pp 105–116
23. Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: Proceedings of IEEE symposium on Security and Privacy (SP). IEEE, pp 39–57
24. Croce F, Hein M (2020) Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proceedings of International Conference on Machine Learning (ICML). PMLR, pp 2206–2216
25. Yamamura K, Sato H, Tateiwa N, Hata N, Mitsutake T, Oe I, Ishikura H, Fujisawa K (2022) Diversified adversarial attacks based on conjugate gradient method. In: Proceedings of International Conference on Machine Learning (ICML). PMLR, pp 24872–24894
26. Chen P-Y, Zhang H, Sharma Y, Yi J, Hsieh C-J (2017) Zoo: zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: Proceedings of ACM workshop on artificial intelligence and security. ACM, pp 15–26
27. Cheng M, Singh S, Chen P, Chen P-Y, Liu S, Hsieh C-J (2020) Sign-opt: a query-efficient hard-label adversarial attack. In: Proceedings of International Conference on Learning Representations (ICLR)
28. Liang S, Li Y, Srikant R (2018) Enhancing the reliability of out-of-distribution image detection in neural networks. In: Proceedings of International Conference on Learning Representations (ICLR)
29. Hendrycks D, Mazeika M, Dietterich T (2019) Deep anomaly detection with outlier exposure. In: Proceedings of International Conference on Learning Representations (ICLR)
30. Chen J, Li Y, Wu X, Liang Y, Jha S (2021) Atom: robustifying out-of-distribution detection using outlier mining. In: Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)
31. Goodfellow IJ, Shlens J, Szegedy C (2015) Explaining and harnessing adversarial examples. In: Proceedings of International Conference on Learning Representations (ICLR)
32. Baluja S, Fischer I (2018) Learning to attack: adversarial transformation networks. In: Proceedings of AAAI conference on artificial intelligence (AAAI). AAAI
33. Xiao C, Li B, Zhu J-Y, He W, Liu M, Song D (2018) Generating adversarial examples with adversarial networks. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), pp 3905–3911
34. Song Y, Shu R, Kushman N, Ermon S (2018) Constructing unrestricted adversarial examples with generative models. In: Proceedings of annual conference on Neural Information Processing Systems (NeurIPS). Curran Associates, Inc., pp 8322–8333
35. Yang J, Zhou K, Li Y, Liu Z (2021) Generalized out-of-distribution detection: a survey. arXiv preprint arXiv:2110.11334
36. Shen Z, Liu J, He Y, Zhang X, Xu R, Yu H, Cui P (2021) Towards out-of-distribution generalization: a survey. arXiv preprint arXiv:2108.13624
37. Radford A, Metz L, Chintala S (2016) Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proceedings of International Conference on Learning Representations (ICLR)
38. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: Proceedings of International Conference on Machine Learning (ICML). PMLR, pp 214–223
39. Brock A, Donahue J, Simonyan K (2019) Large scale GAN training for high fidelity natural image synthesis. In: Proceedings of International Conference on Learning Representations (ICLR)
40. Miyato T, Kataoka T, Koyama M, Yoshida Y (2018) Spectral normalization for generative adversarial networks. In: Proceedings of International Conference on Learning Representations (ICLR)
41. Karras T, Aila T, Laine S, Lehtinen J (2018) Progressive growing of GANs for improved quality, stability, and variation. In: Proceedings of International Conference on Learning Representations (ICLR)
42. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T (2020) Analyzing and improving the image quality of stylegan. In: Proceedings of IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 8110–8119
43. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of IEEE/CVF conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 4401–4410
44. Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat, 400–407
45. Powell MJ (1964) An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput J 7(2):155–162
46. Krizhevsky A (2009) Learning multiple layers of features from tiny images. Master's thesis, University of Toronto
47. Stallkamp J, Schlipsing M, Salmen J, Igel C (2012) Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw 32:323–332
48. Zheng Y, Zhao Y, Ren M, Yan H, Lu X, Liu J, Li J (2020) Cartoon face recognition: a benchmark dataset. In: Proceedings of ACM international conference on multimedia (MM). ACM, pp 2264–2272
49. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252
50. Zagoruyko S, Komodakis N (2016) Wide residual networks. In: Proceedings of the British Machine Vision Conference (BMVC)
51. Carlini N, Wagner D (2017) Adversarial examples are not easily detected: bypassing ten detection methods. In: Proceedings of the ACM workshop on Artificial Intelligence and Security (AISec). ACM, pp 3–14
52. Tramer F, Carlini N, Brendel W, Madry A (2020) On adaptive attacks to adversarial example defenses. In: Proceedings of annual conference on Neural Information Processing Systems (NeurIPS). Curran Associates, Inc., pp 1633–1645
53. ART (2018) Adversarial Robustness Toolbox (ART). https://github.com/Trusted-AI/adversarial-robustness-toolbox
54. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531
55. Yu F, Seff A, Zhang Y, Song S, Funkhouser T, Xiao J (2015) LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365
56. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: Proceedings of International Conference on Learning Representations (ICLR)

Hailong Hu is currently an assistant researcher at Chongqing Technology and Business University. He received his PhD degree in Computer Science from the University of Luxembourg. His work has received the Best Paper Award Honorable Mention at the 2021 Annual Computer Security Applications Conference (ACSAC) and the Best Paper Finalist at the 2018 IEEE International Conference on Networking, Architecture, and Storage (NAS). His research interests include trustworthy generative models and biometric authentication.

Jun Pang is currently an assistant professor at University of Luxembourg. He received his PhD degree from Vrije Universiteit Amsterdam. His research interests include formal methods, graph machine learning, security and privacy.